Next Generation Hardware Development: Framework for a Tailorable Development Method

Günther Schuh[1], Tim Wetterney[2], Felix Lau[2], Stephan Schröder[2]
[1]Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany
[2]Fraunhofer-Institute for Production Technology IPT, Aachen, Germany

Abstract—**Companies are adapting their traditional development processes, aiming for project-specific designs that are referred to as "Agile Product Development" – flexible, adaptive and accelerated processes. Implementing these principles supports developers to react to challenges such as shortened innovation cycles and increased market dynamics. The project-specific tailoring of a development process has to be carried out right from the start. Early on, companies have to create a unique path for each development project by choosing the right development approach – plan-driven, agile or hybrid –, assembling properly skilled development teams and creating an enabling environment and applicable process design. But, due to the lack of appropriate methodologies for a project-specific tailoring, companies find it difficult to adjust development methods according to the project requirements at hand. Therefore, development projects often do not meet expected budgets, deadlines or product goals. As a first step, the authors present a selection of essential development method elements based on a literature review. These elements, classified in content- and process-related elements, form the basis for a comprehensive method framework. The method framework supports companies in identifying and configuring crucial method elements based on which project-specific development strategies can be derived.**

I. INTRODUCTION

Challenges such as growing complexity and dynamics in development and production [1], an increased innovation and time pressure due to shortened product and process life cycles [2] as well as continuously rising quality requirements [3] demand nowadays companies to adapt their traditional processes and understanding about how new products are to be developed [4, 5]. The more and more frequently occurrence of new and disruptive technologies, the market impact of which is very difficult to predict, intensifies these challenges additionally [79]. In consequence, new and optimized development methods, which are highly efficient both in regard to the consumed time and resources, are required so that companies are able to develop innovative products in short development cycles at competitive prices [6, 7].

As a reaction to this highly challenging environment, companies are adapting their traditional, phase-oriented development methods – in hardware development mostly represented by COOPER'S *Stage-Gate* framework – by implementing *agile principles* taken from highly iterative ones such as *Scrum* the origins of which lie within the software development [8, 9]. This methodical shift is based on the assumption that the so-called *agile development methods* are better suited to deal with changing requirements, the need for short development cycles and a continuous integration of and feedback from the customer [10]. This is due to the fact that these "light weight" methodologies are less formal, structured and plan-driven than traditional "heavy weight" approaches such as *Waterfall* – the lack of formality and the usage of suitable practices correlating with an increase in team autonomy, decision speed and overall efficiency [11–13].

While more and more companies – with a growing number of non-IT-related industries – are implementing agile principles into their development processes, the success of these activities is highly varying [14]. There is high insecurity about the optimal balancing of formal development structures vs. agile, highly flexible ones, the way hardware-based products have to be incrementalized in order to generate functional prototypes after each development sprint or the question how highly autonomous development teams have to be managed from a strategic perspective by senior management. This insecurity results in a variety of different *hybrid development methods*, which are adapted company- and project-specific in a more or less experimental approach [15–17]. A lack of understanding about the interrelations between project-specific requirements on the one side and the mechanisms of different practices and structures in development methods on the other can be observed. As a consequence, development projects often do not meet the estimated timeline, cause a higher than expected financial investment or lead to products with insufficient performance, quality or innovativeness [18, 19].

In spite of the existence of a significant number of different plan-driven and agile methodologies, there is not one universal development method that fits all types of development projects [20]. Instead, case studies show and research agrees on the fact that methodologies have to be chosen or rather customized in regard to the specific requirements of different development projects [21, 22, 15]. Yet, this adaption process is all but trivial, since it requires a deep understanding about the effects of different development methods' elements such as architectures, roles and practices as well as their interrelations to specific development projects' characteristics. For the software development environment FIRESMITH made a first attempt to provide a requirement-oriented, tailorable development method called OPEN [23]. However, FIRESMITH'S studies not only focus on software development processes in general, but *agile methods* in specific, hereby providing only limited support for the hardware development environment with its domain-specific requirements [24, 25] and without coverage of more plan-driven approaches.

As a first step towards the development of a tailorable development method for the hardware development, this paper aims for providing a framework which contains and structures general development methods' elements. In contrast to the existing research in this field [21, 26], the authors are not trying to classify established development methods, i.e. Scrum or Waterfall, in abstract dimensions such as agile or plan-driven, but on a more detailed level. Hereby contributing to the growing research discipline of *method engineering* (ME) [27], the authors present only the first of four planned papers targeting on the conception of an adaptive, tailorable development method for hardware projects. The aim is to provide a comprehensive tool for developers that enables them to easily derive a tailored development method based on the specific requirements of the development project.

In chapter II the applied methodology within this paper is explained. Subsequently, chapter III explains the theoretical background of development methods, characteristics of development projects as well as fundamentals on the research discipline method engineering. Chapter IV contains a literature analysis focused on existing methodology frameworks and decision models from the software and hardware environment. Based on the work provided in the previous chapter, in chapter V general elements for the description of development methods are first derived and then clustered within suitable categories – hereby forming the building blocks of the method framework. The conclusion and explanation of future research demand in chapter VI complete the paper.

II. METHODOLOGY

In order to focus on problems with practical relevance, this paper follows a structured approach as shown in Fig. 1. The methodological approach adopts the research process of applied science the target of which is to develop models that shape the future by describing, explaining and configuring parts of the reality [28].

ULRICH's methodology is based on seven sequential process steps. This paper covers the steps A to E; the practical testing (step F) as well as the industrial verification (step G) stay beyond the scope of this paper. First, problems with practical relevance need to be identified and summarized. In this regard, chapter I focuses on the underlying practical problem, which has been derived based on past and ongoing projects with industry partners as well as discussions with other researchers in this field. Chapter III, Theoretical Background, and chapter IV, Literature Review, cover the methodological process steps B and C. Theories, hypotheses and methods from existing research were identified, analyzed and interpreted. Especially in the Literature Review chapter the leading researchers' approaches were evaluated concerning their application potential in the hardware environment. The Results chapter (chapter V) of this paper addresses steps D and E of the methodology. The relevant method elements were derived from existing development methods and serve as basis for future research regarding tailorable development methodologies.

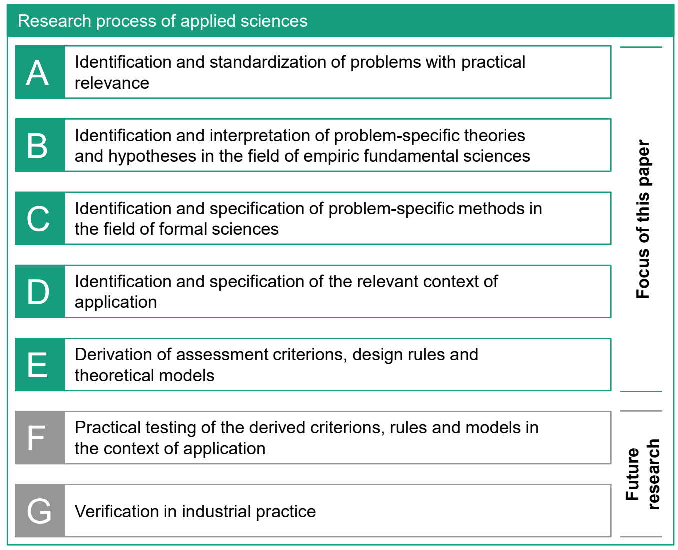

Figure 1 - Research methodology of applied science according to ULRICH [28]

III. THEORETICAL BACKGROUND

In this chapter, definitions regarding the term development method are provided, specific characteristics of development projects are described and the research field method engineering is explained. These information are the theoretical basement for the scientific derivation and conceptualization of a development method framework as presented in chapter V.

A. Development method

There is a variety of different development methods for different disciplines, most widely applied in the fields of software and product development as well as in mechatronics [29, 30]. In the following, an overview of different definitions from the literature is provided based on which a suitable definition in regard to the objective of this paper is derived.

BRUNNER gives a very abstract definition by describing a development method as a tool to systematically describe those of a company's activities that transform the input of a customer using an adequate amount of resources into the output that corresponds with the customers' requests [31].

HAWLITZKY gives a more detailed definition by explaining development methods as a means to generally describe, plan and execute complex tasks from a developers' perspective. For HAWLITZKY this includes the definition of general processes, tools and organizational architectures which are embedded in a phase model that contains the typical stages of the product development cycle [29].

Next to the guidelines regarding the organizational frame, general procedures and sequences of tasks which the previous author defines, BALZERT mentions roles as well as process deliverables as further aspects development methods have to contain according to his understanding [32].

BRINKKEMPER provides the most comprehensive definition by not only integrating all of the aspects mentioned above, but also highlighting that development methods always contain a certain way of thinking or philosophy [33]. In consideration of the sometimes highly conflictive approaches as used by plan-driven and agile development methods, the authors consider this aspect as crucial.

Hence, in this paper development methods are defined as approaches to perform development projects based on a specific way of thinking that consists of guidelines and principles which are structured systematically in terms of development activities with corresponding developer roles and work products [34].

B. Characteristics of development projects

In contrast to other project types, development projects are characterized by their uniqueness, which in consequence requires a unique development process as well as a unique project objective. Similar to other project types, development projects are limited in their *duration* [35] and can be differentiated by their *project size [36]*. Often, development projects require interdisciplinary and inter-divisional cooperation between different groups of individuals [37]. According to PAULUKUHN and SCHMELZER, the main characteristics for development projects are:

- *high risk*, due to the uncertainty regarding project success, timely completion and necessary resources,
- *necessity for creative problem solving*, due to the fact that teams work on a previously unknown problem,
- *high complexity*, resulting from a variety of involved individuals, functions, interests and departments,
- *high degree of novelty*, because of the creation of something new that did not exist before,
- *high variability and dynamics*, resulting from constantly changing requirements [37, 38].

The listing of these characteristics is only an excerpt the purpose of which is to provide a basic understanding about how the development environment differs from other project types. A profound analysis concerning all relevant project characteristics with a possible impact on the configuration of a tailorable development method is part of future work.

C. Situational method engineering

The term method engineering describes an engineering research field that focuses on designing and adapting methods, tools and techniques for the development of systems in general [39]. Although there is a variety of very capable development methods, it is widely recognized that these approaches mostly are too generic to be applied in a specific development context [40]. Hence, tailoring a development method to the problem at hand becomes necessary [41]. *Situational method engineering* as sub-discipline of ME provides assistance for the customized creation of a project-specific development method by selecting method elements – referred to as *method fragments* or *method chunks* – that are stored in a *repository* or *method base* [34]. Yet, despite the advantages of a customized method, constructing models from scratch is difficult and cost-intensive, since existing methods provide little assistance regarding the tailoring process and there is also little knowledge about the interdependencies between the project characteristics on one side and the appropriate method elements on the other [41, 42]. As a first step to improve the tailoring process, the authors focus on ways to create method element clusters stored within a method base, later referred to as *method element framework,* in regard to the specific requirements of the development projects at hand. The following figure shows the typical tailoring process with the method base being the origin for the fragment respectively method element selection, Fig. 2.

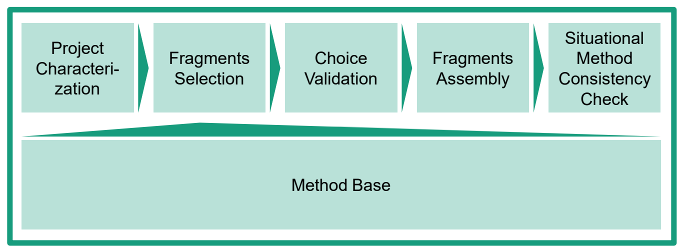

Figure 2 - Process of project-specific method tailoring [43]

IV. LITERATURE REVIEW

The following chapter outlines the current state of research regarding decision models for the project-specific selection of development methodologies. The aim of this literature review is to provide an overview of the most recognized contributions in this field as well as analyzing the existing models' deficits in order to illustrate the current gap in research.

For the definition of the character of a specific development project, ALEXANDER and DAVIS present twenty qualitative criteria which are clustered within the five categories *personnel, problem, product, resource* and *organization*. These criteria are arranged within a selection table and evaluated with a specific rating that describes the appropriateness of using one of six presented development methodologies. This evaluation is based on three method features which are *opportunity to modify software, degree and time of delivered functionality* and *level of activity in respect of time.* Unfortunately, the authors neither provide information about how these method's features were defined nor how the correlation between these features and the project's criteria was estimated [44].

BOEHM and TURNER present a five axis diagram that contains five project characteristics which, according to the authors, function as key discriminators when choosing between agile and plan-driven methodologies, namely: *criticality, personnel, dynamism, culture* and *size*. With three out of five criteria being people-related, the authors put a strong emphasis on the human factor concerning the successful execution of agile methodologies. Yet, it is again not explained which of the agile or plan-driven methodologies' elements increase or weaken the rating of the different dimensions [45, 46].

BUNSE and VON KNETHEN recommend the usage of different agile and plan-driven methodologies based on a system of eleven criteria, which characterize the development project. Regarding the development methodologies, the authors differentiate between *sequential*, *prototyping*, *repetitive* and *agile* approaches. However, it is again unclear, which building blocks respectively instruments, structures or processes of the different methodologies support or prevent their usage, since the selection is done on the method level, not on the level of their underlying elements [26].

ENGELS identifies six key characteristics as main criteria for the selection of an appropriate development method: *Priority of time vs. scope, requirements dynamism, client culture, expected customer participation, dependency between features / components, project size.* In contrast to the previously described models, ENGELS suggests the usage of hybrid methodologies, which are derived by picking suitable elements or building blocks from plan-driven or agile approaches. Although this approach seems reasonable concerning the earlier derived necessity for project-specifically tailored methodologies, the authors provide no guidance on how to create these hybrid models [47, 48].

Based on the agility measurement index of DATTA [49], IACOVELLI derives attributes that characterize an ideal project environment for the application of agile methods. The author evaluates eight agile methods in regard to this set of criteria. As a result, IACOVELLI is able to define three new agile development classes: *Software Development Practices Oriented Methods*, *Project Management Oriented Methods* and *Hybrid Methods.* Furthermore, he derives eight *agile components* which capsule certain agile practices. While the classification of agile methodologies certainly provides a better understanding of their character, the derivation of agile components or building blocks has a higher potential for the project-specific method tailoring this paper is aiming for. Yet, IACOVELLI'S approach is not applicable for this purpose, since his component framework is neither complete nor is it meant to be applied as tailoring approach. The latter is due to the fact that project characteristics are not presented as single items, but in clusters as for the identified agile component *quality* which works for complex and risky projects as well, hereby preventing the usage for i.e. projects that are complex, but not risky [50].

In comparison to the previously presented approaches FERNANDES analyses and classifies two agile methodologies on the level of their underlying practices. For this purpose, the author identifies a set of characteristic practices for each of the agile methodologies, e.g. *Daily Meetings* in Scrum or *Pair Programming* used in *XP*, and compares these characteristics with attributes and sub-attributes such as *response to change* or *minimizing complexity,* which the authors inductively derived from a set of agile methodologies using the *quasiformal comparison* [51]. FERNANDES results describe what the relationship between Scrums' and XP's defining practices and a set of attributes of the agile methodologies are. While the identification of these relationships is one of the ultimate goals to be achieved when aiming for a project-specific method tailoring as this paper is doing, the benefit of his work is limited. Firstly, the evaluation is conducted based on qualitative criteria without explanation; secondly, the set of comparing agile attributes is not comprehensive, but only an exemplary selection, therefore the method is not applicable for all kinds of development projects [52].

In one of his latest publications COOPER presents a hybrid method for the development of physical products. By reviewing industrial best-practice approaches, the author identifies three mayor principles as the basis for his *Next Generation Idea-to-Launch System*: *Adaptiveness and Flexibility*, *Agility* and *Accelerating*. His approach is based on the usage of iterative processes in order to quickly *adapt* to changing information, *agile* development elements focusing on lean, non-bureaucratic processes as well as overlapping stages, properly resourced teams and process automation in order to create an *accelerated* product development method. COOPER differentiates between two project types: traditional product development projects and more innovative, bolder projects with a higher amount of technological risk. He does state the differentiating emphasis of the new system's elements, but yet fails to elaborate on the project-specific decision-making, which stays vague especially in terms of element selection and assembly criteria. Although he mentions a customized process via risk-based contingency approaches, detailed tailoring advices are missing [15].

Overall, seven academic works focusing on decision models and method frameworks from the hardware and software development domain have been analyzed. As a result, the analysis showed that none of the existing works provides a systematic and comprehensive approach to tailor or create a development method in regard to the specific characteristics of the project at hand. However, several reasonable approaches could be identified that will be taken into consideration for the derivation of basic method elements for the *method element framework* in chapter V.

V. RESULTS

In chapter V the authors present the essential building units of a project-specifically tailorable development method, which define its character and its specific mode of action. This is done based on the definitions for development methodologies and project types provided in chapter III as well as on the methodological knowledge gained in the literature review presented in chapter IV. After the selection of a suitable model framework, we will continue with the explication of the model elements the authors identified. These elements will be classified either as *process elements* or *content elements,* hereby building on the works of SHUJA and

KREBS according to whom these two perspectives are applicable for a variety of typical method elements [53].

A. Model classification

This paper will focus on the development of a *descriptive model* as presented in KÜLL and STÄHLY [54]. According to the authors, a descriptive model is used to depict and characterize the logic of a system. For the purpose of this paper, the system to be described contains the elemental building blocks of a development method. In a future paper, a similar descriptive model will be provided for the dimensions and characteristics of a development project. Building on these works, an *explanatory model* will be developed the purpose of which will be to explicate causal connections and correlations between different types of practices, procedures or tools – extracted from the method element framework – on one side and different development project characteristics such as project size or complexity on the other. Ultimately, these scientific papers will allow the conceptualization of a tailorable development method, a *decision model*, which transforms the project-specific characteristics, the system *input,* into the optimal selection of method elements from the method element framework, the system *output*, as shown in Fig. 3**.**

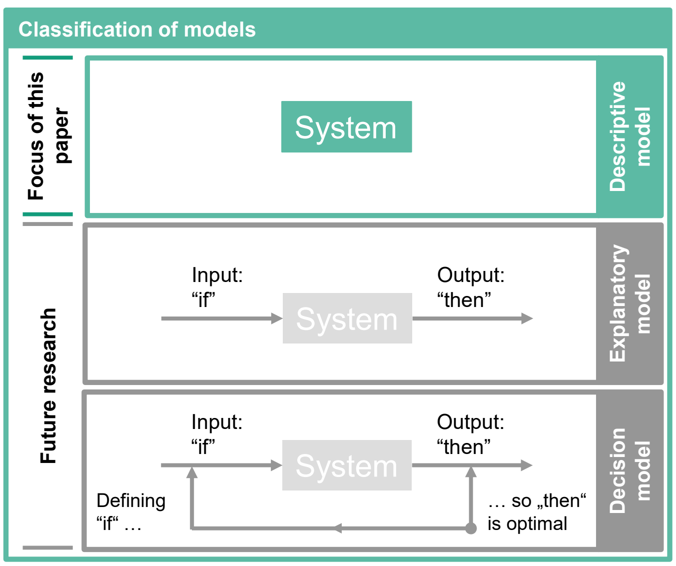

Figure 3 - Model classification according to Küll and Stähly [54]

B. Process-related elements of the method element framework

1. Development Phases

According to LINDEMANN the development process can be described in different levels of granularity, reaching from a micro-level explaining elemental procedures such as how to solve a problem as shown in EHRLENSPIEL, to a macro-level that subdivides the development into general major phases, which contain a certain set of activities, work products and criteria [55, 56]. While procedures on a micro-level are not significantly affected by different characteristics of development projects, phases on a macro-level certainly are [57]. The probably most popular macro-level system in product development is COOPER'S *Stage-Gate*, *Idea-to-Launch Process* [15]. His sequential six stages model is based on the phases idea generation, idea scoping, build business case, development, testing and validation, and launch. Yet, COOPER admits that – depending on the specific project – phases have to be adapted, streamlined or eliminated. Taking into account analyses of several existing plan-driven and agile development methodologies as well as critical reviews of COOPER'S sequential approach [57, 58], the authors recommend the usage of the following three stages: *ideate, develop, validate*.

The ideation phase is meant to cover all activities concerning the overall idea generation, including therefor necessary tasks such as business model generation, technology studies and market studies. Subsequently, the second phase covers the actual development of the product. The character of this phase is very abstract on purpose, since different domains and industries have their own validated and capable development methods the integration of which should be supported [59]. Subsequently, the third phase focuses on the validation of the by then generated work products respectively deliverables. This can either mean the presentation of an *early business concept* to lead users, the *functional testing of a prototype* or the distribution of a *minimal marketable product* to the final customer [60]. With agile methodologies being the origin of this approach, it is intended that these phases are conducted repeatedly in order to improve the product based on the testing results and feedback obtained after each cycle.

2. Proceeding Strategy

In his latest publication COOPER states that in many development projects product requirements are not entirely defined from the beginning [15]. This requires a more experimental development approach that generates results and identifies initially unknown requirements via practical tryout during the development process. This so-called *evolutionary* proceeding strategy is only one of five development classes that BREMER defines [61]. Additionally, he describes *sequential*, *iterative, incremental* and *participatory* proceeding strategies.

Following a *sequential* approach, each development phase, as defined in the previous paragraph, has to be completely finished before the next phase can start. This approach requires a very mature development process with a little degree of uncertainty, meaning, no unexpected events occur during the process. In contrast to that, *iterative* strategies allow and demand the repeated conduction of development phases, each time generating a more mature product. A strictly iterative proceeding strategy assumes that the entire product can be developed in one cycle with a little degree of maturity, but with all necessary functionalities at hand. However, the development of complex products such as planes or cars makes the construction of the overall product in one cycle very difficult if not impossible, which is why an *incremental* proceeding strategy is more appropriate for this type of development. Breaking down the product into several separate modules allows a piece by piece delivery of – ideally – functional and testable increments. Last, BREMER defines *participatory* proceeding strategies that require the involvement of the user as co-developer.

In conclusion, the proceeding strategy defines the principle approach on how to develop a product and has to be chosen in direct dependency of the specific project and product characteristics [46, 12]. In this context, it is important to point out that the proceeding strategies can also be applied in combination, e.g. the iterative conduction of phases is not conflictive with an incremental delivery of the product [17].

3. Milestones

While the importance of phases is decreasing for the overall project planning, *milestones* are becoming more important [16, 15]. Milestones are defined as events to which certain results or deliverables have to be created [62]. They hereby both structure the overall development process and also define the sequence in which different tasks have to be finished. The structuring function is especially important in larger projects with several teams or departments working on the same product but different modules. Here, milestones serve as synchronization points for simultaneously running engineering tasks [63]. The sequencing function is even more essential: For development projects in general but hardware products – because of their increasing changing costs over time – in particular [64], it is of utmost importance to early on reduce the uncertainty regarding technological capabilities, customer requirements or market potentials, to only mention a few [65, 66]. For this purpose, the crucial activities – and thus the respective milestones – have to be identified as well as the sequence in which they have to be conducted. In this regard, CORNING presents a risk-based contingency model that initially identifies key assumptions and unknowns based on which necessary activities are derived, thus preventing developers from pursuing non-value adding activities [15, 67].

In summary, the definition of milestones is highly depending on project- and product-specific requirements, yet a very capable measure to influence key performance indicators such as the necessary time and resources for the development project. For this purpose, the authors recommend the usage of a risk-based contingency model for the project structuring with milestones as presented by CORNING [67].

4. Activities

In each of the earlier defined development phases, different activities are required. In this context, the authors understand activities as processes that have to be conducted in order to generate a certain work product [68]. For the purpose of supporting developers in choosing suitable activities in regard to the specific project requirements and phase, an *activity library* is recommended. For the development phase *ideate*, such a library is defined by the discipline-related clusters *idea generation*, containing creativity techniques such as TRIZ [69], *market research*, explaining different concepts such as lead user integration [70], *business model generation*, showing pros and cons applying e.g. a business canvas approach [71] and *technology studies*, focusing on activities such as technology intelligence in order to identify the best technological alternative [72]. In analogy to this, similar activity libraries have to be developed for the phases *develop* and *validate*.

5. Practices

According to JACOBSON ET AL a *practice* is a proven way of approaching or addressing a problem, which can be communicated to others and applied repeatedly with consistent results [73]. They represent instructions for specific problems occurring in different activities during a development project [17]. JACOBSON ET AL differentiates between three different types of practices two of which apply for the hardware development. *Social Engineering Practices* deal with teamwork, collaboration or communication. For instance, *Stand-Up Meetings* in Scrum are an example for this category. The second category, *Organizational Practices*, focus on more planning-oriented activities such as S*print Planning* in Scrum. With the increasing popularity of agile methodologies, practices became an important topic of research because of their crucial influence for a method's performance and suitability under different development environments [12]. Here, case studies focusing on different practices show positive impacts in dimensions such as team communication, decrease of failure rates or code quality, to mention only a few [74].

Hence, the process-oriented element *practices* comprises a library of different practices which are organized in regard to their specific advantages and disadvantages under varying project characteristics. At this point, a full list of practices cannot be provided due to the sheer number of existing practices some authors expect to reach hundreds [11].

C. *Content-related elements of the method element framework*

1. Deliverables

Deliverables are specific work products which have to be generated in each of the development phases defined in section B [75]. For instance, typical work products in the phase *ideate* are *market analyses*, *competitive analyses* or *technology studies*. These deliverables have to be provided until the end of the development phase, which is represented by a milestone. While the application of deliverables is a common approach in a variety of phase models, a tailorable development method requires flexibility in the means of deliverable definition. Therefore, instead of providing a fixed set of mandatory deliverables for every kind of development project, the authors recommend the definition of project-specific deliverables. For this purpose, the application of CORNING'S risk-based contingency model is a suitable approach to identify the optimal sequence and type of deliverables in order to reduce uncertainty – and thus the risk of failure [67]. A similar approach was suggested by COOPER in 2014 [15].

2. Evaluation Criteria

When products are developed there has to be some kind of assessment in order to evaluate if the generated results meet the initially defined requirements. Hence, the application of evaluation criteria is necessary. Former stage-gate models typically used – often financially-oriented – criteria and KPIs that turned out to be improper due to the high degree of uncertainty especially in early development stages [76]. In consequence, promising development projects were aborted, when the available information concerning future market shares, customer segments or technological capabilities was not sufficient [77, 78]. This problem especially applies for so-called disruptive innovations due to their hard to predict impact on future markets and thus the difficult estimation of potential profits [79].

For this reason, the authors recommend the usage of different evaluation criteria in different development phases. In the initial development phase *ideate*, non-financial criteria such as the *strategic fit* or the *competitive advantage* of the product are more suitable than data-driven ones [80]. In the second phase, *develop*, criteria should focus on product-related aspects such as *functionality* or *design*. These criteria vary with every development project which is why specific criteria cannot be provided here. After a few iteration cycles, activities in the third phase, *validate*, involve testings of mature prototypes such as a *minimum marketable product* under real world conditions [60]. In this context, a variety of measurements for the evaluation of customer satisfaction exists, e.g. the *Kano Model* [81].

3. People

For the design of a tailorable development method, guidelines regarding the optimal composition of *successful teams* in a given development project are necessary. In this context, the criteria for successful teams are *effectiveness* (market success), *efficiency* (meeting budgets and schedules) and *speed-to-market [82].* When setting up a team, the main configuration options are *team size*, *team roles* and *functional diversity* [83]. *Team size* is defined as the number of people within a team. *Team roles* refer to different responsibilities each team member occupies, e.g. *Product Owner* or *Scrum Master* as used in *Scrum [10]*. *Functional diversity* refers to the mixture of different experts in a team.

The expression of these characteristics varies depending on both project-specific requirements on one side as well as dependencies originating from other method elements on the other [45, 46]. For instance, a mechatronics development project might require the integration of several experts from all respective domains, thus influencing the *functional diversity.* At the same time, the number of these experts and the overall team might be limited by scrum-related method elements which recommend an optimal *team size* of three to nine people [84]. Also, research has shown causal connections between the project characteristic *team capability* and the successful application of agile methods, implying that only experienced teams should apply Scrum for example [85].

In conclusion, the three main options for team configuration *team size, team roles* and *functional diversity* have to be chosen in regard to the respective project characteristics at hand as well as in accordance to the other elements of the method element framework.

4. Tools

One of the main drivers for the reported effectiveness of Scrum is the usage of visual project reporting *tools* such as *Burn Down Charts* or *Scrum Boards* [86]. In addition to the two the origin of which lies within the agile methodologies, there is a variety of other planning and reporting tools, i.e. *Gantt Charts* or *Process Mapping*, which base upon more plan-driven approaches [87]. The decision for one of the previously mentioned tools is based on specific project characteristics such as *project size* or *distribution of teams*: While physical Scrum Boards might be applicable for small teams in the same location, large and remotely distributed groups of developers might require cloud-based tools which support the management of interdependencies and complexity [88].

In conclusion, a tailorable development method must provide a set of project management tools the application of which is depending on the given project characteristics. In regard of the plan-driven or agile controversy, it has to be examined how the simultaneous application of tools from different schools is beneficial or counterproductive.

VI. CONCLUSION AND FUTURE RESEARCH

The authors created a method element framework which functions as basement for the project-specific tailoring of a development method. For this purpose, we distinguished between five process-related and four content-related method elements. These elements were detailed and examples were given on how the configuration of these elements is affected by varying project characteristics. Fig. 4 summarizes the findings of this paper.

Researchers and practitioners can use our results in various directions, particularly for the development of a method that enables companies to adapt and choose method elements such as developments phases, proceeding strategies or tools in regard to the project-specific requirements at hand. Yet, for this ultimate objective more research is necessary. As a first step, empirical case studies are required in order to validate the applicability of the method elements we identified.

Furthermore, by creating a descriptive model that comprises suitable elements for a tailorable development method, this paper focused on the methodology side only. In-depth analyses of the defining project characteristics were not undertaken. Hence, future research has to examine which are the crucial project characteristics that impact the development process and if it is possible to derive distinctive development project types. Here, the works from PAULUKUHN and SHENHAR are a valid starting point, although the authors did not examine the interrelations between project characteristics and the development methodology in particular [38, 89]. Subsequently, when research has shown *which* project characteristics affect the development process and vice versa, the next question is *how* the interrelations between certain project characteristics and the method elements – as shown in Figure 4 – work. For this, the following publications from JIANG and FERNANDES should be taken into account [12, 52].

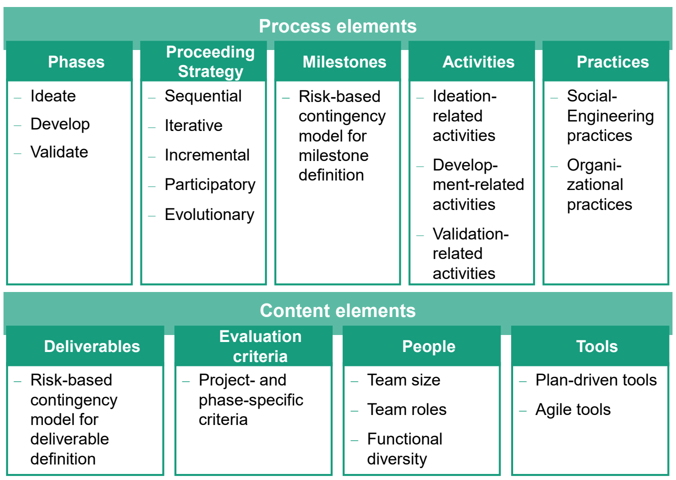

Figure 4 - Method element framework for the project-specific tailoring of a development method

Last, literature on situational method engineering is based on the assumption that all necessary information concerning the relevant project characteristics are available from the start. Yet, this case is found far too rarely in practice [90]. Instead, high degrees of uncertainty are characterizing the early phases of development – the so-called *Fuzzy Front End of Innovation* [76]. For this reason, future works have to examine when the defining project characteristics are available, what level of maturity these information need to have and how changing requirements [91] would affect the tailoring of the development method.

REFERENCES

[1] W. Dohmen, *Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme,* 2002.

[2] W. Eversheim and G. Schuh, *Integrierte Produkt- und Prozessgestaltung.* Berlin. Springer, 2005.

[3] D. Steffen and K. Jilg, "Kundenorientierte Produktentwicklung," *Automobilelektronik,* pp. 56–58, 2009.

[4] R. Reichwald, F. Piller, and C. Ihl, *Interaktive Wertschöpfung. Open Innovation, Individualisierung und neue Formen der Arbeitsteilung.* Wiesbaden. Gabler Verlag / GWV Fachverlage GmbH, Wiesbaden, 2009.

[5] M. Schenk and S. Wirth, *Fabrikplanung und Fabrikbetrieb. Methoden für die wandlungsfähige und vernetzte Fabrik.* Berlin [u.a.]. Springer, 2004.

[6] VDI-Gesellschaft für Produkt- und Prozessgestaltung, *Entwicklungsmethodik für mechatronische Systeme,* Jun, 2004.

[7] R. Berndt, *Management-Konzepte für kleine und mittlere Unternehmen.* Berlin. Springer, 2006.

[8] R. G. Cooper, "Perspective: The Stage-Gate Idea-to-Launch-Process. Update, What's New and NexGen Systems," *Product Innovation Management*, vol. 25, pp. 213–232, 2008.

[9] K. Schwaber, *Agile project management with Scrum.* Redmond, Wash. Microsoft Press, 2004.

[10] K. Schwaber and J. V. Sutherland, *Software in 30 days. How Agile managers beat the odds, delight their customers, and leave competitors in the dust.* Hoboken, N.J. John Wiley & Sons Inc, 2012.

[11] R. Hauber and I. Jacobson, "Practices als Alternative zu vollständigen Prozessen," *Objektspektrum*, vol. 03, pp. 62–69, 2009.

[12] L. Jiang, "Towards a framework for understanding the relationships between classical software engineering and agile methodologies," in *Proceedings of the 2008 International Workshop on Scrutinizing Agile Practices or shoot-out at the agile corral.* New York, NY: ACM, 2008, pp. 9–14.

[13] E. Hanser, *Agile Prozesse. Von XP über Scrum bis MAP.* Heidelberg, New York. Springer, 2010.

[14] A. Komus, *GPM-Studie Status Quo Agile 2014. Agiles und klassisches Projektmanagement - Miteinander statt gegeneinander.* Nürnberg, Oct, 2014.

[15] R. G. Cooper, "What's Next?: After Stage Gate. Progressive companies are developing a new generation of idea to launch processes," *Research Technology Management*, January –February, pp. 20–31, 2014.

[16] A. F. Sommer, C. Hedegaard, I. Dukovska-Popovska, and K. Steger-Jensen, "Improved Product Development Performance through Agile/Stage-Gate Hybrids. The Next-Generation Stage-Gate Process?," *Research-Technology Management*, January—February, 2015.

[17] S. Dyck and T. A. Majchrzak, "Identifying Common Characteristics in Fundamental, Integrated, and Agile Software Development Methodologies," in *2012 45th Hawaii International Conference on System Sciences (HICSS).* IEEE, Ed., pp. 5299–5308.

[18] R. L. Glass, *ComputingFailure.com. War stories from the electronic revolution.* Upper Saddle River, N.J. Prentice Hall PTR, 2001.

[19] J. Gausemeier, "Entwicklung mechatronischer Systeme," in *Handbuch Produktentwicklung.* B. Schäppi, Ed. München, Wien: Hanser, 2005.

[20] R. L. Glass, "Matching methodology to problem domain," *Communications of the ACM*, vol. 47, pp. 19–21, 2004.

[21] J. Charvat, *Project management methodologies. Selecting, implementing, and supporting methodologies, and processes for projects.* New York. Wiley, 2003.

[22] A. Cockburn, *Agile software development. The cooperative game.* Upper Saddle River, NJ. Addison-Wesley, 2007.

[23] D. G. Firesmith and B. Henderson-Sellers, *The OPEN process framework. An introduction.* London, New York. Addison-Wesley, 2002.

[24] G. Herzwurm, *Kundenorientierte Softwareproduktentwicklung.* Wiesbaden. Vieweg+Teubner Verlag, 2000.

[25] K. Thompson, *Agile Processes for Hardware Development,* https://www.cprime.com/wp-content/uploads/woocommerce_uploads/2015/10/cPrime-Agile-Processes-for-Hardware-Development.pdf, 2015.

[26] C. Bunse and A. v. Knethen, *Vorgehensmodelle kompakt.* Heidelberg [u.a.]. Spektrum, Akad. Verl., 2002.

[27] S. Brinkkemper, K. Lyytinen, and R. J. Welke, *Method engineering. Principles of method construction and tool support: Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering 26-28 August 1996, Atlanta, USA.* London, New York. Chapman & Hall, on behalf of the International Federation for Information Processing (IFIP), 1996.

[28] H. Ulrich, T. Dyllick, and G. Probst, *Management.* Bern. P. Haupt, 1984.

[29] N. Hawlitzky, *Integriertes Qualitätscontrolling von Unternehmensprozessen: Methodische Gestaltung eines Quality-Gate-Konzeptes zur Planung, Messung und Steuerung der Prozessqualität.* München. TCW, Transfer-Centrum für Produktions-Logistik und Technologie-Management, 2002.

[30] E. Wallmüller, *Software-Qualitätsmanagement in der Praxis.* München. Hanser, 2001.

[31] F. J. Brunner, Ed., *Taschenbuch Qualitätsmanagement. Leitfaden für Studium und Praxis.* München, Wien. Hanser, 2008.

[32] Balzert, Helmut: *Lehrbuch der Software-Technik*. 2. Aufl. Heidelberg [u.a.]: Spektrum, Akad. Verl. (Lehrbücher der Informatik), 2008.

[33] S. Brinkkemper, M. Saeki, and F. Harmsen, "A Method Engineering Language for the Description of Systems Development Methods," in *Advanced information systems engineering. 13th international conference, Interlaken, Switzerland, June 4 - 8, 2001; proceedings*, vol. 2068. K. R. Dittrich, Ed. Berlin [u.a.]: Springer, 2001, pp. 473–476.

[34] B. Henderson-Sellers and J. Ralyté, "Situational Method Engineering: State-of-the-Art Review," *Journal of Universal Computer Science*, vol. 16, pp. 424–478, 2010.

[35] J.-P. Thommen, *Lexikon der Betriebswirtschaft. Managementkompetenz von A bis Z.* Zürich. Versus Verlag AG, 2008.

[36] B. Verworn, *Modelle des Innovationsprozesses.* Hamburg, https://www.tuhh.de/tim/downloads/arbeitspapiere/Arbeitspapier_6.pdf, 2000.

[37] H. J. Schmelzer, *Organisation und Controlling von Produktentwicklungen. Praxis des wettbewerbsorientierten Entwicklungsmanagements.* Stuttgart. Schäffer-Poeschel, 1992.

[38] L. Paulukuhn, *Typologisierung von Entwicklungsprojekten im Maschinenbau.* Aachen. Shaker, 2005.

[39] Brinkkemper, S., Saeki, M., Harmsen, F., "A Method Engineering Language for the Description of Systems Development Methods," in *Advanced information systems engineering. 13th international conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, 2001: proceedings.* K. R. Dittrich, A. Geppert, M. C. Norrie, Eds. Berlin, New York: Springer, 2001.

[40] B. Fitzgerald, N. L. Russo, and T. O'Kane, "Software development method tailoring at Motorola," *Commun. ACM*, vol. 46, pp. 64–70, 2003.

[41] G. Engels and S. Sauer, "A Meta-Method for Defining Software Engineering Methods," in *Graph Transformations and Model-Driven Engineering*, vol. 5765. D. Hutchison,. Berlin: Springer, 2010, pp. 411–440.

[42] J. Iivari, "Amethodology for IS development as organizational change," *Systems Development for Human Progress,* pp. 197–217, 1989.

[43] A. Verrijn Stuart, T. W. Olle*, Methods and associated tools for the information systems life cycle. Proceedings of the IFIP WG 8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle, Maastricht, the Netherlands, 26-28 September 1994,* Amsterdam, New York: Elsevier, 1994.

[44] L. C. Alexander and A. M. Davis, "Criteria for selecting software process models," in *Fifteenth Annual International Computer Software & Applications Conference,* pp. 521–528.

[45] B. W. Boehm and R. Turner, "People factors in software management:. Lessons from comparing agile and plan-driven methods," *CrossTalk The Journal of Defense Software Engineering,* pp. 4–8, 2003.

[46] B. W. Boehm and R. Turner, *Balancing agility and discipline. A guide for the perplexed.* Boston. Addison-Wesley, 2004.

[47] M. Heinemann and G. Engels, "Auswahl projektspezifischer Vorgehensstrategien," in *Integration von Vorgehensmodellen und Projektmanagement. 17. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e.V. (GI), Stuttgart, 12. - 13.04.2010.* O. Linssen, Ed. Aachen: Shaker, 2010, pp. 132–142.

[48] G. Engels and M. Kremer, "Situational Software Engineering:. Ein Rahmenwerk für eine situationsgerechte Auswahl von Entwicklungsmethoden und Vorgehensmodellen," in *Informatik 2011. Informatik schafft Communities; 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 4.10. bis 7.10.2011, TU Berlin.* H.-U. Heiss, Ed. Bonn: Ges. für Informatik, 2011.

[49] S. Datta, "Agility measurement index," in *the 44th annual southeast regional conference.* R. Menezes, Ed., p. 271.

[50] A. Iacovelli and C. Souveyet, "Framework for Agile Methods Classification," in *Proceedings of the International Workshop on Model Driven Information Systems Engineering. Enterprise, User and System Models.* S. Ebersold, Ed., 2008.

[51] T. W. Olle, H. G. Sol, and C. J. Tully, *Information systems design methodologies. A feature analysis: Proceedings of the IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies, York, U.K., 5-7 July, 1983.* Amsterdam, New York, New York, N.Y. North-Holland; Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co, 1983.

[52] J. M. Fernandes and M. Almeida, "Classification and Comparison of Agile Methods", in *2010 Seventh International Conference on the Quality of Information and Communications Technology (QUATIC),* pp. 391–396.

[53] A. K. Shuja and J. Krebs, *IBM rational unified process reference and certification guide. Solution designer.* Upper Saddle River, N.J. IBM Press, 2008.

[54] S. Zelewski, "Grundlagen," in *Betriebswirtschaftslehre.* 4th ed. H. Corsten, Ed. München [u.a.]: Oldenbourg, 2008, pp. 1–98.

[55] K. Ehrlenspiel, *Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit.* München [u.a.]. Hanser, 2007.

[56] U. Lindemann, *Methodische Entwicklung technischer Produkte.* Berlin, Heidelberg. Springer Berlin Heidelberg, 2009.

[57] D. Kelley, "Adaptation and Organizational Connectedness in Corporate Radical Innovation Programs," *Journal of Product Innovation Management*, vol. 26, pp. 487–501, 2009.

[58] R. Sethi and Z. Iqbal, "Stage-Gate Controls, Learning Failure, and Adverse Effect on Novel New Products," *Journal of Marketing*, vol. 72, pp. 118–134, 2008.

[59] M. Boucher and D. Houlihan, *System Design. New Product Development for Mechatronics,* http://www.plm.automation.siemens.com/zh_cn/Images/Aberdeen_Mechatronics%20System%20Design%20Benchmark_nonlogo_Jan08_tcm78-58217.pdf, 2008.

[60] R. Pichler, *Agile product management with Scrum. Creating products that customers love.* Upper Saddle River, NJ. Addison-Wesley, 2010.

[61] G. Bremer, *Vorgehensmodelle für die betriebliche Anwendungsentwicklung.* Stuttgart. Teubner, 1998.

[62] M. Gessler and R. Kaestner, *Kompetenzbasiertes Projektmanagement (PM3).* Nürnberg. GPM Dt. Ges. für Projektmanagement, 2009.

[63] Müller, G., Reindl, P., "Der BMW DMU-Prozess mit Entwicklungspartnern," in *Virtuelle Produktentstehung in der Fahrzeugtechnik. Tagung Berlin, 9. und 10. September 1999.* VDI, Ed. Düsseldorf: VDI Verlag, 1999, pp. 135–147.

[64] B. Gloger, *Scrum in der Hardwareentwicklung,* https://borisgloger.com/wp-content/uploads/2014/07/Whitepaper-Hardware.pdf?882268, 2016.

[65] A. Khurana and S. R. Rosenthal, "Towards holistic 'front-ends' in new product developments," *Journal of Product Innovation Management*, vol. 15, pp. 57–74, 1998.

[66] S. K. Markham, "The Impact of Front-End Innovation Activities on Product Performance," *Journal of Product Innovation Management*, vol. 30, pp. 77–92, 2013.

[67] B. Kirk, "Accelerating time to market: Using a next generation innovation framework," in *Stage-Gate Innovation SUMMIT,* 2013, pp. 26–28.

[68] R. Höhn, Ed., *Das V-Modell XT. Anwendungen, Werkzeuge, Standards.* Berlin, Heidelberg. Springer, 2008.

[69] C. Gundlach, Ed., *Innovation mit TRIZ. Konzepte, Werkzeuge, Praxisanwendungen ; jetzt mit eBook, Fachinformation per Mausklick.* Düsseldorf. Symposion, 2006.

[70] E. v. Hippel, *Democratizing innovation.* Cambridge, Mass. MIT Press, 2005.

[71] A. Osterwalder, Y. Pigneur, and T. Clark, *Business model generation. A handbook for visionaries, game changers, and challengers.* [Amsterdam]. Alexander Osterwalder & Yves Pigneur, 2010.

[72] G. Schuh and S. Klappert, *Technologiemanagement. Handbuch Produktion und Management 2.* Berlin, Heidelberg. Springer-Verlag Berlin Heidelberg, 2011.

[73] I. Jacobson, P. W. Ng, and I. Spence, "Enough Process - Let's Do Practices," *Journal of Object Technology*, vol. 6, pp. 41–66, 2007.

[74] D. E. Strode, *The Agile Methods: An Analytical Comparison of Five Agile Methods and Investigation of Their Target Environment.* New Zealand, 2005.

[75] R. G. Cooper, "Stage-Gate Systems. A New Tool for Managing New Products," *Business Horizons,* pp. 44–54, 1990.

[76] E. Stevens, "Fuzzy front-end learning strategies. Exploration of a high-tech company," *Technovation*, vol. 34, pp. 431–440, 2014.

[77] Q. Zhang and W. J. Doll, "The fuzzy front end and success of new product development. A causal model," *Euro Jrnl of Inn Mnagmnt*, vol. 4, pp. 95–112, 2001.

[78] R. G. Cooper, "Managing Technology Development Projects," *IEEE Engineering Management Review*, vol. 35, pp. 67–76, 2007.

[79] C. M. Christensen, Eichen, Stephan F. von der, and K. Matzler, *Innovators Dilemma. Warum etablierte Unternehmen den Wettbewerb um bahnbrechende Innovationen verlieren.* München. Vahlen, Franz, 2010.

[80] R. G. Cooper, "Invited Article. Where Are All the Breakthrough New Products?: Using Portfolio Management to Boost Innovation," *Research-Technology Management*, vol. 56, pp. 25–33, 2013.

[81] C.-C. Chen and M.-C. Chuang, "Integrating the Kano model into a robust design approach to enhance customer satisfaction with product design," *International Journal of Production Economics*, vol. 114, pp. 667–681, 2008.

[82] N. Sivasubramaniam, S. J. Liebowitz, and C. L. Lackman, "Determinants of New Product Development Team Performance. A Meta-analytic Review," *J Prod Innov Manag*, vol. 29, pp. 803–820, 2012.

[83] J. R. Hackman, *Leading teams. Setting the stage for great performances.* Boston, Massachusetts. Harvard Business Review Press, 2002.

[84] A. Mundra, S. Misra, and C. A. Dhawale, "Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software," in *2013 13th International Conference on Computational Science and Its Applications (ICCSA),* pp. 119–123.

[85] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, pp. 961–971, 2008.

[86] K. Schwaber, *Agiles Projekmanagement mit Scrum.* Unterschleissheim. Microsoft Press, 2007.

[87] J. M. Wilson, "Gantt charts. A centenary appreciation," *European Journal of Operational Research*, vol. 149, pp. 430–437, 2003.

[88] M. A. Razzak and D. mite, "Knowledge Management in Globally Distributed Agile Projects - Lesson Learned," in *2015 IEEE 10th International Conference on Global Software Engineering (ICGSE),* pp. 81–89.

[89] A. J. Shenhar, "From theory to practice. Toward a typology of project-management styles," *IEEE Trans. Eng. Manage.*, vol. 45, pp. 33–48, 1998.

[90] T. Williams and K. Samset, "Issues in front-end decision making on projects," *Proj Mgmt Jrnl*, vol. 41, pp. 38–49, 2010.

[91] M. Rossi and B. Ramesh, "Managing Evolutionary Method Engineering by Method Rationale," *Journal of the Association for Information Systems*, vol. 5, pp. 356–391, 2004.