ADDING AGILITY TO SOFTWARE READINESS ASSESSMENT PROCEDURES – A CASE ON DIGITAL TRANSFORMATION FROM THE AUTOMOTIVE INDUSTRY

Marcus Grieger  
*Kühne Logistics University, marcus.grieger@the-klu.org*

André Ludwig  
*Kuehne Logistics University, andre.ludwig@the-klu.org*

Jun Shen  
*University of Wollongong, jshen@uow.edu.au*

Follow this and additional works at: [https://aisel.aisnet.org/ecis2018_rp](https://aisel.aisnet.org/ecis2018_rp)

Recommended Citation  
[https://aisel.aisnet.org/ecis2018_rp/102](https://aisel.aisnet.org/ecis2018_rp/102)

This material is brought to you by the ECIS 2018 Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in Research Papers by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
ADDING AGILITY TO SOFTWARE READINESS ASSESSMENT PROCEDURES – A CASE ON DIGITAL TRANSFORMATION FROM THE AUTOMOTIVE INDUSTRY

Research paper

Grieger, Marcus, Kühne Logistics University, Hamburg, Germany, marcus.grieger@the-klu.org
Ludwig, André, Kühne Logistics University, Hamburg, Germany, andre.ludwig@the-klu.org
Shen, Jun, University of Wollongong, Wollongong, Australia, jshen@uow.edu.au

Abstract

In order to enable continuous and fast software development, manufacturers adopt agile development methodologies. In respect thereof, the organisation’s operational processes need to be adapted to fit changing software project’s needs. By means of a case study with an automotive OEM that introduces microservices (MS) architectures, the transformation of its software readiness assessment (SRA) procedure is investigated. The article introduces an artefact that builds upon standardized technical and organisational constructs inherent to MS projects. Further, by conceptually modelling a methodology is presented that guides an organisation in the transformation to implement agile SRA within its current operational infrastructure. The artefact is validated by means of three MS projects and respectively adapted. The findings suggest the artefact to be a useful intermediary step, but its successful implementation requires the integration of all contributing departments. The study deepens the knowledge about the transformation of an organisation’s operational procedures by an empirical case and possible methodological paths.

Keywords: Digital Transformation, Software Readiness Assessment, Automotive Industry, Agile Software Development.

1 Introduction

Emerging technologies provide manifold possibilities to build novel digital services and respective business models in manufacturing industries (Matt et al., 2015). As a result, the competitive dynamics are increasing, propelled by both incumbents and market entrants that largely come from the internet industry or consumer electronics (Charias and Hess, 2016). Consequently, original equipment manufacturers (OEMs) have servitized their portfolios, which manifests itself by an increased focus on the design and provision of digital services, and a shift from selling products to product-service solutions, driven by financial, strategic and marketing aspects (Baines et al., 2009). However, creating services alone is not a guarantee for success as this shift involves a “transformational journey” as well (Gaiardelli et al., 2015, p. 1165). OEMs need to accelerate their innovation and service development processes in order to be able to compete with IT companies and startups (Tian et al., 2016). In this context, many organisations use and adapt digital technologies to enable new value creating activities, generally referred to as digital transformation (Gimpel and Röglinger, 2015, Bounfour, 2016).

Although a universally accepted definition of digital transformation does not exist (Ferreira, 2017), the domain has been investigated alongside other topics with regard to different dimensions (Bounfour, 2016), business strategic aspects (Bharadwaj et al., 2013, Hess et al., 2016), driving factors (Gimpel and Röglinger, 2015, Charias and Hess, 2016) as well as implementation processes (Dremel et al., 2017). A variety of companies among different sectors were surveyed and specific fields of action for executing an organisation’s digital transformation could be identified, namely being the customer, the accrued data,
the transformation of the value proposition, the organization, a firm’s operation activities, as well as its transformation management (Gimpel and Röglinger, 2015). A large number of these studies demonstrate how digital transformation is a complex venture that “affects many or all segments within a company” (Hess et al., 2016, p. 2). Generally, manufacturing firms recognize the need for digital transformation and, in turn, many have formed units in respect thereof (Charias and Hess, 2016). Current research on digital transformation has mainly focused on detecting obstacles and necessary fields of action for a successful implementation and management of the according processes. But, executing this shift challenges traditional manufacturers (Baines et al., 2009, Beuren et al., 2013, Reim et al., 2014, Charias and Hess, 2016) as the tools and methodologies that guide the realisation of these transformational areas are scarce (Wallin, 2013, Bounfour, 2016), and empirical studies, such as Hess et al. (2016), focus on the strategic transformation implications, but do not give guidance on how to practically conduct specific tasks.

Since digital products and services become more individualized and fragmented into smaller features (Olsson et al., 2012), the demands software development projects need to meet change too. Flexibility, incremental releases and development speed, i.e. the reduction of a product’s time to market, are primary project requirements and are a necessary precondition to stay competitive (Al Alam et al., 2016). Therefore, OEMs increasingly introduce agile development methods (Piccinini et al., 2015) that promise greater productivity, product and service quality, and shorter development cycles in return (Dingsøyr and Lassenius, 2016). However, organisations are struggling to adapt their internal operations to these requirements, especially to provide an integrated IT infrastructure that enables collaborative, fast and flexible product development (Gimpel and Röglinger, 2015).

In product-centric organisations, particularly those in the automotive industry, security and quality assurance have been central pillars around which value-creating processes are structured, both culturally as well as operationally. In order to assure these quality demands are met, organizations set-up mechanisms and processes to assess a product’s market readiness (Goicoechea and Fenollera, 2012), so-called software readiness assessment (SRA) procedures. When these organisations adopt agile development methodologies, their operational processes have to be transformed accordingly, as digital products need to be assessed quickly and continuously in order to support effective decision making (Nierstrasz and Lungu, 2012). Empirical studies on how to transform SRA procedures within operational units do not exist and the need for applicable processes that fit into existing IT environments is evident.

The aim of this paper is to complement the current theoretical understanding of the digital transformation of operational processes. More specifically, we want to clarify the transition of an organisation’s SRA procedures to suit agile development methodologies. Managerially, we want to support decision makers with practical guidance on how to facilitate this transition in an approach that is integrable within an existing operational infrastructure. Based on the preliminary considerations, we derive the following research question for our investigation:

**How can an organisation transform its software readiness assessment procedure to enable agile product development within its current operational infrastructure?**

We answer this question by conducting an empirical study with a major automotive OEM, which introduces microservice architectures to address the need for more flexible and faster software development. The automotive industry is well-suited to conduct this study, as OEMs have already established IT departments and provide digital services, which are qualitatively assessed before going live. Furthermore, they are in the process of digital transformation, which is a topic of great concern to them. Despite that, the industry is still looking for applicable solutions to guide this process. As Tian et al. (2016, p. 6) point out, a major “challenge for automotive companies is to operate and innovate like IT companies and make the car a central part of an ecosystem that merges cyber-physical content and social networking, as well as agile processes for development”.

The remainder of this article is structured as follows: In section 2 we present the theoretical background on digital transformation, SRA procedures and agile product development. Section 3 outlines the applied methodology for the investigation. Section 4 comprehensively describes the procedure within the case
company as well as an applied modelling approach. Subsequently, section 5 discusses the validation of our approach by means of three software projects. Section 6 discusses the findings in terms of its managerial and scientific implications. Finally, section 7 gives a brief summary, outlines the research limitations and points out room for future scientific activities.

2 Theoretical Background

2.1 Microservice Architecture

As digital products and services become more individualized, fragmented into smaller features and time-to-market cycles shortened (Olsson et al., 2012), microservice architectures have gained in popularity. Microservices (MS) are not firmly defined but can be approximated by a number of characteristics (Wolff, 2017). MS are inspired by service-oriented architectures and are small, self-contained services that are independently processed and communicate event-based, e.g. via APIs (Dragoni et al., 2017). They are introduced to split large software systems into smaller components (Wolff, 2017). MS can be independently deployed, function autonomously and possess their own data storage (Hasselbring and Steinacker, 2017). They are built around business capabilities and overcome the limited scalability of monolithic systems. Typically, they are created and owned by one team, following a DevOps approach (Dragoni et al., 2017), characterized by limited scope and limited functional requirements (Nadareishvili et al., 2016).

2.2 Agile Software Development

In order to meet flexible and fast product delivery, more and more large companies adopt agile development methodologies (Henriques and Tanner, 2017) replacing the predominantly traditional ones, such as the waterfall procedure, as they are not suited for immediate requirement changes (Kotaiah and Khalil, 2017). To specify agility, most researchers rely on the definition of the Agile Manifesto that is based on practitioner's experience (Hummel, 2014) and comprises four values and twelve principles (Beck et al., 2001). In adopting agile methods, organizations expect to raise productivity, product and service quality, and accelerate their software development cycles (Dingsøyr and Lassenius, 2016). Many researchers have investigated a number of agile software development methodologies, such as eXtreme programming, Crystal, SCRUM and Feature Driven Development. Generally, the agile development process iteratively goes through the multiple phases, i.e. requirements definition, design, implementation, testing, review and completion (Kotaiah and Khalil, 2017), by promoting small release cycles, which continuously integrate customers into the development process, that is iteratively in defined periods (Hummel, 2014), and new code as soon as it is ready (Abrahamsson, 2002). Software is developed lean and fast, following a minimal viable product (MVP) approach, thus allowing to grow incrementally over time. Agile development methodologies emphasize continuous product delivery, enabled by the regular integration of code during development (Dingsøyr and Lassenius, 2016). Software shall be kept constantly in a releasable state. Therefore, organisations have to automate software building, deployment, testing and release processes (Humble and Farley, 2011). In addition, continuous delivery makes it necessary to adopt a development team composition and assigning responsibility within an organisation. Teams are vertically structured and follow a DevOps approach, taking an end-to-end responsibility for the product where software development and operations are combined (Dingsøyr et al., 2014). In order to instantly know when software is in a releasable state respective assessment processes also have to be set-up.

2.3 Software Readiness Assessment

Before a software product is released to the market or integrated into the goods, assessing its readiness is critical to ensure correct operations. Operational readiness refers to a concept that attempts to quantify the “probability that, at any point in time, the system is ready to be placed into operation on demand when used under stated conditions” (Kececioglu, 2003, p. 24). Subsequently, procedures to predict
operational readiness have been investigated and developed by researchers and practitioners in the form of software maturity (SM) (Mankins, 2009), release readiness (RR) (Al Alam et al., 2017) and software readiness (SR) (Asthana and Olivieri, 2009). All of these research streams examine the assessment of software by certain criteria in order to support the decision between releasing software too early or too late (Quah, 2009). The division among them is blurred as the definitions for RR (Al Alam et al., 2016), SM (Schaefer, 2009), and SR (Olivieri, 2012) are not well-defined and the terms “maturity” and “readiness” are used synonymously (Gove and Uzdzinski, 2013). Nevertheless, we use the term “software readiness” without the intention to exclude relevant publications on “software maturity” or “release readiness”, since it is the designated term within the case company. Typically, SRA are conducted at several points during a software’s life cycle and, depending on the organization, can be either a small or an extensive, highly formal process that involves external peer reviewers. However, assessing technologies poses various organizational and methodological challenges, from choosing the right metrics to ensuring and achieving the right technology level across multiple systems (Mankins, 2009). The purpose of SRA methods is the identification of a discrepancy, which will be resolved by subsequent improvement actions (Pfeffer and Sutton, 1999; Mettler, 2011). Common approaches to assess SR are checklists, industry standards and academically developed methodologies (Asthana and Olivieri, 2009). Since Gibson and Nolan (1974) first introduced a four-stage maturity model, (Gibson and Nolan, 1974) many more assessment models have been developed over the years: the capability maturity model (CMM) (Paulk et al., 1993), the ISO/IEC 15504 norm for software process improvement and capability determination, a defect tracking method using predictive modelling approaches (Quah, 2009) or the IT Capability Model Framework (IT-CMF) (Curley and Kenneally, 2012).

Deciding the readiness of software “requires continuous and customized measurement” (Al Alam et al., 2017, p. 382), and often these decisions are based on good practices subject to an informant’s bias (Mettler, 2011). Further investigation on how methods can be used and applied in existing maturity assessment methods, that is, the existing operational infrastructure of an organisation, is needed. The models are highly complex and specialized, which is why the majority of the information gathering tasks are still carried out manually (Proença and Borbinha, 2016). Research on assessing SR is ongoing and as Mettler (2011, p.82) points out “lots of these models do not describe how to effectively perform these actions. As software development methodologies shift towards agile, SRA procedures have to transition as well, and researchers are even calling for an agile software assessment discipline, which is the “study of the tools and techniques that will allow developers to integrate analysis tools into the daily workflow of software development” (Nierstrasz and Lungu, 2012, p. 3).

3 Research Methodology

3.1 Case Research

To investigate the transformation of software readiness assessment processes, we conducted an interpretative case study with an automotive OEM using a phenomenon-based research approach. We chose to conduct a case study as it allows the investigation of organizational problems in their natural environment. In essence, case studies aim to clarify the reasons for decisions, how they are implemented, and what are their outcomes (Schramm, 1971). Thus, we are able to gain valuable insights into understanding the complexity of operational transformations of SRA procedures that are considered in the problem-solving process and artefact design (Recker, 2013). Phenomenon-based research enables us to study an organization in its real-life environment and allows us to consider the restrictions and impediments organizations face (Krogh et al., 2012). Corresponding with the four key characteristics outlined by Paré (2004), case research in this context is useful as the subject of investigation is complex, it cannot be studied outside its environmental context, there is insufficient theoretical knowledge and a holistic approach is needed.

Following the activities for phenomenon-based research outlined by Krogh et al. (2012) (see Figure 1) we began by identifying the research problem (Yin, 2014). Further, we intensified the exploration of the subject by collecting data on the OEM’s SRA processes, screening documents and conducting semi-
structured interviews with employees (Mayring, 2014). By applying conceptual modelling (Hoppenbrouwers et al., 2005) we analysed the OEM SRA procedure and developed a software readiness template (SRT) based on the current procedure. Subsequently, we developed the continuous delivery checks (CDC), thus aligning the OEM’s assessment procedure to it prevalent agile infrastructure. Following a Human Risk and Effectiveness Strategy for evaluating design science research strategies, the SRT was validated by means of three MS development projects (Venable et al., 2016).

3.2 Case Description

The study took place at the central headquarters of an automotive OEM where the corporate IT department is situated. The time period of observation and collaboration stretched from 01.02.2017 to 15.08.2017. The case company has an annual turnover exceeding 10 billion Euros and over 100,000 employees of which the large majority works in production. The OEM produces vehicles at multiple facilities throughout the world and is a public limited company. The OEM’s product portfolio ranges from compact to premium segment vehicles. In the course of its digitization strategy, the OEM wants to increase the number of digital products, that is digital services. The company has an IT department that not only provides infrastructural IT services, but also takes responsibility for software projects on both product and corporate levels. Software development projects primarily act within waterfall environments, but occasionally proceed in an agile way. The OEM’s SRA approach is owned and supported by a separate unit within the IT testing department. The OEM intends to transition its software project development methodology from predominantly waterfall procedures to an agile delivery management. Therefore, the IT architecture has to be aligned with a more flexible and customer-centred focus that enables fast digital service developments within software projects. To meet these demands, the IT architecture department introduced the concept of microservice (MS) architecture and has supported several software development projects to implement it.

3.3 Data Collection and Analysis

During the time of the case study, we aimed to get a holistic understanding of the transformation process of SRA procedures. Accordingly, we used multiple means of data collection (Recker, 2013). The SRA information was provided by the department for the integration of customer and commercial processes in the forms of documents and access to the company’s proprietary online tool. Further, we conducted semi-structured interviews with personnel from multiple departments, most notably, the head of the SRA advancement and management, SRA commissioning managers, software development project teams, the IT architecture department as well as contiguous business units that contribute to the overall procedure and qualitatively analysed these (Mayring, 2014).

4 Continuous Delivery Method

Following, we outline the research results. In section 4.1, we focus on the analysis of the OEM’s SRA procedure, identify deficiencies out of a MS project perspective and derive agile assessment requirements. Section 4.2 presents the development approach of a Software Readiness Template (SRT) that supports MS projects and clarifies how to apply it. Finally, section 4.3 outlines the continuous delivery
checks and the method of how to make the SRA procedure agile ready and, ultimately, integrable within the organization’s operational infrastructure.

4.1 Software Readiness Assessment Analysis

All software projects within the organization have to complete SRA before going live. The objective of the SRA is to secure high software quality and adherence to the organization’s governance and compliance guidelines. The SRA is an internally constructed online tool and consists of a questionnaire of 171 questions, which is structured in accordance to the categories of architecture (33/171), test (63/171) and operations (75/171). Within these categories, it is further divided into different criteria (see Table 1). The majority of the questions are open-ended, and multiple request for more than one answer, such as the first question of the assessment: “Have the functional requirements for the system been defined and documented?” As the focus of this article are the methodological aspects on transforming SRA procedures, the questions are quoted exemplary.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Test Planning and Test Controlling</td>
<td>Operation Preparation</td>
</tr>
<tr>
<td>Security and Compliance</td>
<td>Test Specification</td>
<td>Monitoring</td>
</tr>
<tr>
<td>Interfaces</td>
<td>Test Execution</td>
<td>Business Continuity</td>
</tr>
<tr>
<td>Monitoring planning</td>
<td>Test Evaluation and Test Closure</td>
<td>Organisation</td>
</tr>
<tr>
<td>Infrastructure (Hardware &amp; Software)</td>
<td>Test Data and Test Environment</td>
<td>Release Management</td>
</tr>
<tr>
<td></td>
<td>Defect Management</td>
<td>Service Level Management</td>
</tr>
<tr>
<td></td>
<td>Code Quality</td>
<td>Configuration Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incident / Problem Management</td>
</tr>
</tbody>
</table>

Table 1. Criteria of SRA Checks

For each software development project an SRA is initiated with a set-up. The SRA is structured along five project development phases: “project setup” (8 questions), “exploration” (33 questions), “sprint” (87 questions), “go live” (21 questions), and “transition into line” (22 questions). These phases are typically traversed in a subsequent order. Each answer to a question has to be evaluated with a respective fulfilment degree that ranges in quartiles from 0 to a 100 %. If a certain score is reached at the end of one phase, the project is ready to progress to the next development phase. Answering the entire questionnaire results in the calculation of a software readiness index (SRI) that has to reach a specified score to be formally accepted. Regardless of the outcome of the score, defined “must criteria” have to be met before any software goes live. So, for instance, if the overall SRI score is 95%, but one ”must criterion” was not met, the software will not be accepted to go live.

The SRA is monitored and supported by commissioning managers (CM) who fill the questionnaire with partial help from project team members. Thus, the independence of the assessment is secured. In some occasions, e.g. if a project is time-sensitive and further resources are needed, multiple CMs may work on one SRA and more may be hired on a contractual basis. For many answers, documents or some other form of verification needs to be created in order to be assessed as fulfilled, such as certifying the conclusion of interface contracts. Figure 2 illustrates the current process within a UML 2 activity diagram.

The SRA is so comprehensive that it is suitable for all of the OEM’s software projects. However, it is a one-size-fits-all solution as the questionnaire is a monolithic construct and therefore is not adaptable to individual projects’ needs. Regardless of the project development methodology, such as waterfall or agile, and scope, such as small or large, the SRA has to be filled in the same way. The SRA phases typically follow the course of a project’s progress. The organization's current process is best-suited for large-scale, long-running software development projects, but imposes a significant overhead for digital products which follow a minimal viable product (MVP) approach and grow incrementally over time (DevOps). Contrary to the agile principle of response to change, a defined functionality or feature cannot
quickly be implemented and tested under market conditions as verifications, documents and detailed answers have to be given ex ante in order to be ready for the release. The online tool offers no central storage management for documents or any other verifications. Documents and answers cannot be transferred from one project to another. So, for every project set-up the documents and verifications have to be re-created. MS projects that are developed agile, have a set of different characteristics, see Table 2, from which we derived SRA requirements that took into account the artefact design.

Figure 2. UML 2 Activity Diagram of SRA Procedure

Table 2. SRA Demands for Agile Product Development

<table>
<thead>
<tr>
<th>MS Projects</th>
<th>SRA Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum viable product development</td>
<td>Minimal viable assessment</td>
</tr>
<tr>
<td>Automatic performance scaling</td>
<td>Flexible assessments that adapt with scaling</td>
</tr>
<tr>
<td>Continuous delivery of new features</td>
<td>Assessment as an iterative process</td>
</tr>
<tr>
<td>Use of standardized constructs:</td>
<td>Standardised assessment criteria that projects</td>
</tr>
<tr>
<td>Automated test pipeline up to production</td>
<td>can accept to comply, build upon and</td>
</tr>
<tr>
<td>API gateway for communication</td>
<td>revisit (reusability)</td>
</tr>
<tr>
<td>Development environment</td>
<td></td>
</tr>
<tr>
<td>DevOps enabled processes</td>
<td></td>
</tr>
<tr>
<td>Agile product delivery of build, measure, learn</td>
<td>Reactive software assessments</td>
</tr>
</tbody>
</table>

In order to get the SRA agile ready, we follow a two-step process. First, we aim to reduce the answering effort for MS projects and, secondly, integrate the SRA into the OEM’s agile workflow.

4.2 Software Readiness Template

Independent of their size and scope all MS projects share certain architectural features (Wolff, 2017). Following the SRA analysis (Sec. 4.1), document screening and analysis and interviews we identified that these projects rely on standardized technical constructs and organisational implementations. Hence, a set of SRA questions should be answerable by referring to the basic conditions these standards provide. The first standards we determined are the characteristics of the MS architecture (see Sec. 2), inherently used by any MS project. Secondly, MS projects use an automated test pipeline (ATP), a defined tool-based workflow, an API gateway, that operates as a standardized communication entry point, and a development environment (DE) that provisions, manages and scales the microservices. Finally, MS projects follow a DevOps approach. After having identified these standards, we transferred the SRA questionnaire from the online tool to an excel spreadsheet, which serves as the working basis. Subsequently, we analysed each one of the 171 SRA questions, and parts of them, taking an MS perspective of whether required verification can be provided within a package, is automatable by the use of the ATP, the API gateway and DE, or simply not applicable anymore. Thereby, we defined the terms as seen in Table 3, and reinsured our evaluations with two CMs.
As the results in Table 3 show, over 50% of the 171 SRA questions could be answered in a pre-packaged way. Around 30% of the assessment verifications can even be automated, most notably queried test criteria by using the ATP and DE. The 15 questions that were identified as no longer applicable, mostly relate to the changed application operation through the DevOps teams. 56 SRA questions were identified to be neither packable nor automatable and would still need to be manually answered for every MS project. The majority of MS projects start small and grow over time, for instance, when new functional requirements are added. Thus, the projects are required to answer certain SRA questions once or iteratively, e.g. per sprint, depending on certain triggers. In total 18 triggers were identified by working through the software readiness questionnaire with 2 CMs and elaborating on which changes/alterations/conditions require a software project to update or readjust the given answer. Subsequently, we abstracted these triggers and categorized them within the dimensions: product, business, operations & infrastructure and governance & compliance, see Table 4.

Table 3. Evaluation criteria of the SRA questionnaire analysis

<table>
<thead>
<tr>
<th>Packable:</th>
<th>Automatable:</th>
<th>Not Applicable:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- SRA questions that can be answered by the basic conditions that the DE, the API gateway and ATP provide as templates</td>
<td>- SRA questions that are automatically answered for MS projects which use the DE, the API gateway and ATP for communication, automated staging, logging and testing</td>
<td>- SRA questions that are not applicable to MS development projects</td>
</tr>
<tr>
<td>- E.g. is the life-cycle for the infrastructure components checked (SRA question 31) by the persons responsible for the DE</td>
<td>- E.g. automated tests will be executed in the background for every code commit and can be viewed at any time</td>
<td>- E.g. responsibilities for hardware operations</td>
</tr>
</tbody>
</table>

Table 4. Trigger Categories and Triggers

<table>
<thead>
<tr>
<th>Trigger Category</th>
<th>Product</th>
<th>Business</th>
<th>Operations &amp; Infrastructure</th>
<th>Governance &amp; Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>New functional req.</td>
<td>Scaling and performance</td>
<td>API specification changes</td>
<td>New security req.</td>
<td></td>
</tr>
<tr>
<td>Personal data use</td>
<td>Extreme scaling changes</td>
<td>Change in test cases</td>
<td>Personal data use</td>
<td></td>
</tr>
<tr>
<td>New service or mainten. req.</td>
<td></td>
<td>New operation critical defects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Data types</td>
<td></td>
<td>DevOps team changes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New functional req. that necessitates user training</td>
<td></td>
<td>Emergence of new interface</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Based on these preliminary SRA evaluations, we formulated generalized target picture answers out of the perspective of an MS development project and checked each proposal with a commissioning manager. As a result, we created a template of the SRT answers that could be provided to MS projects as a working basis for their SRA procedure, see Figure 3. Utilizing the SRT, MS projects get immediate support within the prevalent procedure to fill the SRA since the answers can be copied and adapted to application specific needs. Subsequently, project teams can derive tasks and integrate them within their project management tool. In accordance with agile principles, the SRT template displays if the question has to be answered once or recurring.
Continuous Delivery Checks (CDC)

The SRT is the foundation for the conceptual development of the Continuous Delivery Package (CDP). The CDP is a pre-checked bundle of basic conditions that MS projects accept and commit to comply with, therefore automatically answering a great amount of the SRA questionnaire, identified in the previous analysis. According to the MVP approach, MS projects start small and are advanced with every sprint or when new functional requirements are added to the service. In order to reduce the time-to-market for MS projects, the CDP is provided to them. Projects have to check if the basic conditions specified in the CDP are sufficient for them and if they can ensure to comply with them at their current development state. For larger projects that cannot agree to commit to the basic conditions, the respective SRA criteria have to be specified and evaluated as usual.

The CDP is formed by the identified infrastructure components, architectural criteria or procedures that all MS projects share and is manifested in form of bundled documents, templates, agreements, lists and links among others, see Figure 4. For instance, the 99.9% guarantee of the availability of a DE in its standard configuration should be enough for most MS projects in their initial phase.
The CDP is part of the Continuous Delivery Checks (CDC), which is a concept that integrates the SRA procedure within the OEM’s agile development methodology and conforms to the requirement to continuously deploy software, see Figure 5. Upon project set-up, a set of the SRA questions is automatically migrated into the project’s backlog. If MS projects agree to comply with the CDP, a significant amount of these backlog issues is answered with the pre-defined basic conditions. The residual amount has to be answered by the project in the respective sprints. The answering and acceptance of backlog issues by the CM satisfies the SRA when calculating the readiness index. The CDC are conducted iteratively in accordance to a project’s agile development state and whether a trigger requires a project to adapt an answer. If MS projects follow the Scrum methodology, sprint cycles can be a period determining factor. For the increasing variety of software development projects, e.g. MS projects, this configurable readiness assessment approach can reduce the evaluation effort as only criteria relevant to the project’s current circumstances and development state are considered. The CDC provide a solution for an agile SRA by making the existing procedure flexible to scale with the software.

![Figure 5. The Continuous Delivery Checks Principle](image)

5 SRT Test and Validation by Software Development Projects

In accordance to Frank’s (2007) evaluation process model, we first chose a Human Risk and Effectiveness Strategy for validating a design science research artefact (Frank, 2007; Venable et al., 2016). As this strategy entails an empirical validation in the natural context, it is well-suited for case research (Venable et al., 2016). In order to validate the artefact, the SRT was given to the commissioning managers of three MS projects to perform the company’s SRA, see Table 5. In this context, the CMs received an introduction about the structure, how the document is best used and the intentional future implementation of the CDC. After the application of the SRT, we conducted semi-structured interviews and qualitatively analysed their feedback, following the summarising content analysis proposed by Mayring (2014). The CDC methodology relies on the correctness and usability of the SRT. The SRT is an artefact, which MS development projects should be able to immediately use, as general, applicable answers are provided. We discuss and interpret the results via a cross-case analysis (Khan and VanWynsbergh, 2008).

The MS projects were of different scope and addressed distinct services, as can be seen in Table 5. Project A intended to develop a service that shows which vehicles are in stock in a certain region to support the point-of-sale representatives. Project B wanted to centralize the authorization and authentication management for digital products. Project C developed a remotely connectable infotainment service. Apart from project A, only one CM primarily worked on the SRA of their respective MS. The interview data was analysed along the dimensions usefulness, correctness, completeness, and comprehensibility.

Generally, the projects remarked the SRT to be useful for MS projects to a certain degree, however their assessment differed, whereas the project context affected the amount of answers that could be copied. The SRT answers were assessed to be partially correct and due to the generalist aspiration provide room for interpretation. Therefore, Project A and B had to adapt many of the suggested template answers.
Project A and C noted most of the SRT answers to be complete. Project B did not disclose any information regarding this dimension.

<table>
<thead>
<tr>
<th>Service</th>
<th>Project A</th>
<th>Project B</th>
<th>Project C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stock locator service (B2B)</td>
<td>Authorisation &amp; Authentication Mgmt. Service (B2C, B2B)</td>
<td>Remote vehicle services (B2C)</td>
</tr>
<tr>
<td>CMs</td>
<td>2 (internal &amp; external)</td>
<td>1 (internal)</td>
<td>1 (external)</td>
</tr>
<tr>
<td>Usefulness</td>
<td>To a certain degree helpful</td>
<td>To a certain degree, as general answers often still need to be adopted - but indicatively useful</td>
<td>A significant number of answers were copied and adapted to project needs</td>
</tr>
<tr>
<td></td>
<td>Conceptually yes, but without too lintegration for now an extra effort</td>
<td>For smaller projects more useful than for bigger ones</td>
<td></td>
</tr>
<tr>
<td>Correctness</td>
<td>Partially correct answers and additions were provided</td>
<td>Not wrong, but some proposals are too general and give room for interpretation</td>
<td></td>
</tr>
<tr>
<td>Completeness</td>
<td>Mostly complete, but additional processes, such as licence agreements, could be added</td>
<td>(No remark)</td>
<td>Complete, but additional standards may be identifiable</td>
</tr>
<tr>
<td>Comprehensibility</td>
<td>External CM: comprehended</td>
<td>Initial difficulty, but during the application learning effects</td>
<td>SRT functioning comprehended, however some compr. problems due to missing knowledge about the infrastr. tasks and processes</td>
</tr>
<tr>
<td></td>
<td>Internal CM: difficult to comprehend</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remarks</td>
<td>For proper usage, agile integration necessary</td>
<td>For proper usage, agile integration desirable</td>
<td>The SRT was helpful, but is not enough as the SRA still needs to be aligned to agile tools</td>
</tr>
<tr>
<td></td>
<td>External CM: useful</td>
<td>Further improvement by more categorisation and filter option</td>
<td></td>
</tr>
</tbody>
</table>

*Table 5. Overview of test MS projects and validation results*

However, both Project A and C provided suggestions that more standards might or could be supplemented. All projects stated to have had initial comprehensibility concerns, which could be overcome with progressing SRA answering (Project B and C). Project A’s internal CM had difficulty comprehending the SRT’s concept. The CM particularly noted that infrastructural components, such as the DE and ATP, could not give basic conditions, and instead, the MS projects must state them. The idea of reversing the operational readiness principle by providing restrictions and not asking for all eventualities was noted to be difficult to imagine. Contrary, the external CM stated to comprehend its intentional use, and noted, for proper usage, an integration within the organisation’s agile toolchain to be necessary for a full, potential benefit exploitation, as did the CMs of Project B and C. Based on the remarks, especially of the external CM, we adapted certain answers and provided additional filter features that would make the document more usable. The conceptual idea of the CDC was regarded to be primarily good and necessary in order to transform the current rigid process, but a managerial decision to actually transform the SRA was at the time of the project’s assessment missing, which caused insecurities to arise and employees to conduct the SRA procedure as usual parallelly.

6 Discussion

6.1 Scientific Implications

Previous studies on digital transformation have focused on current challenges and the identification of necessary fields of action (Baines et al., 2009, Gimpel and Röglinger, 2015, Bounfour, 2016). It was suggested that operational procedures need to be adapted, but methods that support their execution have
been scarcely researched. Building on these findings the present article presents a methodology on the operational level derived from an empirical case and therefore contributes to the identified demand (Gimpel and Röglinger, 2015). Further, the research on agile SRA is still in its infancy, but is gaining in importance (Nierstrasz and Lungu, 2012). Our study provides a foundation to the understanding of the far-reaching implications a setup of agile readiness assessment has. By identifying constructs and standards that all projects within an organisation use, and further by deriving restrictions and developing the necessary SRA answers, an approach is provided that may be applicable to other kinds of software development projects as well.

In addition, agile SRA requirements for MS projects were identified. Contrary to many studies on the strategic level (Hess et al., 2016, Khan, 2016) with regard to digital transformation, the novelty of this study is the provision of a transformation methodology by means of empirical data and the identification of impediments when trying to initiate its implementation. Finally, as for the research on microservice architectures this study provides further insights about their implementation difficulties. In sum, this article contributes to the understanding of how manufacturing organizations can transform their SRA procedure to suit agile development methods by providing a methodology that takes the existing operational infrastructure into account. This study demonstrates on an operational level that the decision to become a solution provider and its resulting consequences challenge an organisation’s operational procedures as a transformation is a process, stretching over a period of time. Consequently, existing software readiness assessment (SRA) processes have to be managed and maintained, while new ones have to be implemented. In this respect, the study highlights the notion of previous findings that digital transformation is a complex venture that affects many sections within an organisation (Hess et al., 2016).

6.2 Managerial Implications

On a managerial level, we can learn from this study that deciding to implement a new architecture pattern in order to improve software development processes, has far more reaching consequences than can be previously envisioned. Setbacks have to be accepted and key stakeholders need to be aware that transforming processes can result in additional efforts. That is why it is crucial to include, if possible, all contributing parties. Further, manufacturers that aspire to be leading digital service providers, have to be able to enable fast and flexible product development if they want to stay competitive in the market. Processing IT projects in accordance to these requirements poses difficulties to OEMs as their current structures cannot operationally handle these requirements. It could be observed that the SRA procedure is a reflection of the organization’s product-centric structure, which is oriented around the development and advancement of physical goods. Software development projects are processed in a rather waterfall environment, where it is required to answer to all eventualities ex ante rather than respond to change. As Hess et al. (2016, p. 2) point out, “managers often lack clarity about the different options and elements they need to consider in their digital transformation endeavours”. This study demonstrates these different elements and the methodology provides instructions of how to adapt agile assessment procedures.

7 Conclusion

In this study, we conducted an empirical investigation with an automotive OEM that introduced microservice (MS) architectures as part of a transformation initiative to enable fast digital product development. Thereby, we investigated the question of how to transform the organisation’s software readiness assessment (SRA) procedure under consideration of its current operational infrastructure. We therefore analysed the existing process by reviewing literature, screening documents, conducting interviews, and by identifying current deficiencies and agile assessment requirements. Based on these results, the SRA was transferred into a working document to provide a basis for the development of an applicable software readiness template (SRT) for MS projects. On the foundation of the SRT we created the concept of Continuous Delivery Checks (CDC) by providing an initial framework of basic conditions of constructs and processes all MS projects use. The intention of the CDC is to reduce the answering effort.
and makes the organisation’s SRA flexible, as well as integratable within its agile toolchain or its operational infrastructure. It allows DevOps teams to administer the SRA as part of their tasks of the respective sprint within one project management tool. The SRT was validated by three MS projects and respectively adapted. We observed the SRT to be a useful intermediary step within the transformation process to an agile software assessment procedure, but its successful implementation and adoption depends on the integration of all contributing departments and cannot be executed independently. In this context, we noticed employees to have difficulties to change their mindset, though majoritarian they supported the concept of the CDC. In addition, without the implementation of the CDC and a managerial mandate the created artefact was not as beneficial to the MS projects, as it could have been.

The CDC is an approach to transform business operations and to contribute to propelling the development of digital products and services, thus leveraging the application of digital technologies, such as MS, in order to enhance customer value, both being focal points of companies leading digital transformation (Berman, 2012). Further, the approach demonstrates a way on how to integrate assessment processes within agile development processes, which increases the application benefit of the organisation’s agile tools in use. After all, the potential of agile project development supporting technologies cannot be exploited if an organisation’s operations remain unaltered and unintegrated. The methodology has been positively received by the person responsible for the SRA advancement and management. By the time of writing, a mandate within the organisation was created based on the findings of this study, which promotes the advancement and implementation of the CDC or agile SRA.

However, the findings of this study are subject to the following limitations. As is typical for case study methods research, not all variables could be controlled, despite the high dependence of observations and artefacts on the study context, thus complicating replication (Recker, 2013). The case research was conducted with one manufacturing company and the question is whether the application in a different organisation, or set of organisations, from different industries would have resulted in a different methodological approach. Further, the validation of the SRT with different MS projects could have possibly yielded different feedback. In order to address these limitations, we endorse further empirical research on digital transformation ventures both with companies of the same, and of different, industries.

This article can provide the background for further research on the transformation of agile SRA in operational procedures in more broader terms. Theoretically, the unification or clear differentiation of software maturity, release readiness and software readiness assessments by providing definitions or characteristics are topics of future interest. Also, investigating the applicability of the developed procedure across different industries as well as the research on complementing methods and further advancement of the existing ones, provide many points of reference for future research as generally applicable tools still need to be developed. Additionally, the integration of agile SRA procedure within agile development methods is a topic of growing concern. The current study could be expanded on in additional software development projects, other than MS, generalizing the results to formulate comprehensive transformation frameworks.
References


