

A Reflection on the Interrelations Between Business Process Management and Requirements Engineering with an Agility Perspective

Citation for published version (APA):

Aysolmaz, B., Gürsul, M., Kirchner, K., Laue, R., Mertens, R., Reher, F., Schönreiter, I. M., Turban, B. M., & Weissbach, R. (2018). A Reflection on the Interrelations Between Business Process Management and Requirements Engineering with an Agility Perspective. In E. Teniente, & M. Weidlich (Eds.), *Business Process Management Workshops. BPM 2017. : Lecture Notes in Business Information Processing* (Vol. 308, pp. 669-680). Springer. https://doi.org/10.1007/978-3-319-74030-0_54

Document status and date:

Published: 17/01/2018

DOI:

[10.1007/978-3-319-74030-0_54](https://doi.org/10.1007/978-3-319-74030-0_54)

Document Version:

Accepted author manuscript (Peer reviewed / editorial board version)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Download date: 28 Mar. 2024

A Reflection on the Interrelations between Business Process Management and Requirements Engineering with an Agility Perspective

Banu Aysolmaz¹, Mehmet Gürsul², Kathrin Kirchner³, Ralf Laue⁴, Robert Mertens⁵,
Felix Reher⁶, Irene Schönreiter⁷, Bernhard Turban⁸, and Rüdiger Weißbach^{9*}

¹ Maastricht University, School of Business and Economics, the Netherlands
b.aysolmaz@maastrichtuniversity.nl

² STM Defence Technologies Engineering and Trade Inc., Ankara, Turkey
mehmetgursul@gmail.com

³ Berlin School of Economics and Law, Germany
kathrin.kirchner@hwr-berlin.de

⁴ University of Applied Sciences Zwickau, Germany ralf.laue@fh-zwickau.de

⁵ HSW University of Applied Sciences, Hameln, Germany mertens@hsw-hamelnde

⁶ University of the West of Scotland, Paisley, United Kingdom
felix.reher@uws.ac.uk

⁷ Technical University Dresden, Germany irene@schoenreiter.de

⁸ Univ. of Applied Sciences RheinMain, Wiesbaden, Germany
bernhard.turban@hs-rm.de

⁹ Hamburg University of Applied Sciences (HAW), Hamburg, Germany
ruediger.weissbach@haw-hamburg.de

Abstract. The paper points out some aspects of the interrelations between business process management, agility, flexibility, and requirements engineering. It shows some possibilities for agile development of business processes and for the development of flexible processes for changing requirements.

Keywords: Business process management, requirements engineering, agile, flexible processes, business process life cycle.

1 Agility and Flexible Processes

Agile principles [1] (AP) are common in Software Engineering (SE) and became more important in Business Process Management (BPM) in the last years [2]. The terms of agility as well as flexibility are widespread and mutual in BPM literature [3–5]. The definitions of flexibility have a great similarity to those of agility. Unfortunately, agility is often misused as a synonym for flexibility [3]. Agility can have varied interpretations in the field of BPM. On the one hand, it can refer to organizational agility, which can be an outcome of BPM deployment in an organization. It can also refer to the usage of agile approaches to BPM deployment efforts. Additionally, “agile BPM” may refer to business process management systems (BPMS) that incorpo-

* Authors are listed in alphabetical order to reflect their equal contributions.

rate agile development methodology in process automation. We proclaim the use of different terms for the clear differentiation of the actually intended contents. In our understanding, “agility” denotes agile methods according to AP, while “flexibility” refers to the adaptability, responsiveness, and context dependency of the processes.

In this paper, we present various aspects of the interrelations between BPM and requirements engineering (RE) with an agile perspective. In the rest of this chapter, we discuss on flexibility and organizational agility, the concept of flexible business processes (BPs), and how to apply agile methods in BPM. In Chapter 2, we look into the relationship between SE, RE, and BPM. Chapter 3 focuses on agile development processes and BPM. In Chapter 4, we discuss on some organizational circumstances that require agility and flexibility in processes. Lastly in Chapter 5, we conclude the paper.

1.1 Flexibility and organizational agility

A large body of knowledge around the definitions of agility and flexibility exist in literature. Eardley et al. suggest that flexibility is the ability to change direction rapidly or deviate from a predetermined course of action [6]. Doz et al. define strategic agility as “the capacity to continuously adjust and adapt strategic direction in a core business to create value for a company” [7]. Organizational agility is referred to as “the capacity of an organization to efficiently and effectively redeploy/ redirect its resources to value creating and value protecting (and capturing) higher-yield activities as internal and external circumstances warrant” [8].

Singh et al. indicate that an organization’s stimuli-response actions considered agile can be explained using a bi-fold idea of “magnitude of variety” change (named flexibility) and “rate of generating variety” change (named responsiveness) [9]. The “magnitude of variety change” refers to the architectural aspect of change and indicates the extent to which an organization can change and the amount of changes made in its products or practices. The other aspect, namely the “rate of variety change”, refers to the impermanence of change and charts the relationship of the change in variety with time. Thus, organizational agility consists of the amount of change the firm makes to its products or processes in response to environmental stimuli as well as the rapidity with which such changes are made. This flexibility and responsiveness create “a meta-capability” that deploys a dynamic balance between sensing opportunities, enacting complementarities, and capturing value over time [10].

Organizational agility is necessitated by changes in an organization’s environment such as technological shifts, talent pool skills shifts, resource limitations, emerging consumer markets, and changes in consumer expectations. An organization’s agile capabilities govern how it integrates, builds, and reconfigures its resources both internal and external in response to these changing environments. From the organizational perspective, the flexibility of processes is different from the organization’s ability to change processes. When processes are flexible, the organization can be more stable, at least until the number of process changes had changed the organization itself.

1.2 Flexible business processes

BPM needs to support, enact, and integrate organizational agility in order to dynamically manage BPs [11]. The aim of BPM is to improve the efficiency and effectiveness of the processes through process redesign and incorporate automation where feasible. BPM deployment typically involves implementing process-based, long-range business applications as part of a BPMS. Contexts change over time; thus, processes require continuous adaptation to the given context. This, in some cases, leads to obsolescence of the re-designed process in the implementation stage due to the rapidly changing requirements in a dynamic environment.

The key benefits of BPM can be summarized as “efficiency, effectiveness and agility” [12]. In the current dynamic and globalized environment, organizations need to be able to assimilate and counter changes on a real-time basis. The control on processes and the platform for rapid workflow modifications provided by BPM provides this agility. The feedback on operations management occurs in real time, with information secured and made available instantaneously, keeping productivity on track and efficient. In this way, company operations can react with greater agility, enacting operations change more easily and often. BPM empowers organizations to react better to times of quick change. The re-design of the workflow is then the instrument through which the association can react to this change. Therefore, BPMSs have to go beyond their classic features and incorporate, beside others, a contextual process management. Such systems have to support that process participants decide on the execution of activities as well as their order during process execution. Thus, it is not possible to fully prescribe such a process beforehand as it can be done for, e.g., a standardized purchasing process.

Some approaches have been developed to support flexible processes on design time as well as on runtime. During *runtime*, the ADEPT system offers the functionality of making dynamic changes during the execution time of a pathway. Running process instances can be migrated to new process model versions [13]. Based on ADEPT, MinAdept provides techniques for mining flexible processes [14]. Till now, the ADEPT concept does not include a concept for monitoring flexible processes.

For modeling variability at *design time*, several approaches exist: Declarative process modeling is an activity-centered approach in which constraints are used to prevent certain behavior [15]. During run-time, only allowed activities are shown in the work list of the user, and he decides about next activity to be executed. For specifying variants in procedural process models at design time, Hallerbach et al. identified two solutions in traditional tools [16]. The *multi-model approach* requires a separate model for each variant. The *single-model approach* makes use of one big model which covers all possible variants. Decisions that could be made at design-time appear as conditional branching that takes place at run-time. Both approaches have obvious disadvantages. The requirement is to have models that can be configured at design time. Torres et al. discusses two solution approaches [17]. *Behavioral approaches* model a superset of all variants and derive a particular variant by hiding or blocking elements. *Structural approaches* start from a “base” process model and derive variants by applying a set of change operations. Configurable Event-Driven Process

Chains [18] is an example of behavioral approaches, whereas Provop [19] and vBPMN [20] follow the structural approach.

In the case handling approach, activities can be executed based on data dependencies [21]. For example, if an activity is still running but already produced data necessary to execute the next step, the following activity can start. In the same vein, the concept of Procllets allows the division of a process into several process parts. These snippets can be executed one after the other or interactively [22]. *Case management*, covered by CMMN, requires modeling that can express the flexibility of a knowledge worker during run-time while selecting and executing tasks for a specific case [23]. Tasks are modeled and can be specified as either mandatory or discretionary during design-time and serve as recommendations during run-time.

1.3 Application of Agile Methods in BPM

Traditionally, the BPM deployment lifecycle emphasizes detailed up-front planning of process analysis, process design and modeling, process implementation, monitoring, and improvement activities. In other words, traditional BPM adopts a waterfall approach. Von Rosing et al. have proposed the use of the agile method through the various stages of the BPM lifecycle [24]. In an agile approach, analysis, planning, and architecture design are the beginning phases (corresponding to BPM's design and modeling phase). On the other hand, the build, test and deployment phases have a circular approach in multiple short iterations instead of the linear execution, monitoring and improvement phases of traditional BPM.

Agile BPM targets an initial high-level blueprint detailing the estimates for project releases, resources, risks, and cost and benefits. From this, a BPM deployment roadmap is outlined as to when and which requirements can be met as the project advances through small releases. In the next phase, instead of a detailed up-front process model, a high-level design for the "to-be" processes is developed at the start of the project that sets the foundation for the agile BPM project choices and options. This high-level design guides the detailed design in each iteration as specified by the stakeholders, within the budgeted time and cost parameters.

Agile BPM in the deployment phase is based on cooperative and information-aligned sharing of accountability and governance. Traditional BPM project governance uses a gated approach to release and monitor the fixed up-front project funding and outcomes. Since agile BPM breaks up the effort into short releases, the funding is also made available on the successful release of each iteration.

2 Relationship Between SE, RE, and BPM

2.1 The Software Part of BPM

Development of a BPMS is an essential step in the BPM lifecycle. A BPMS is typically developed in implementation stage, which helps businesses to automate and manage BPs and roles [25]. A BPMS is one of the most recommended investments for process improvement [26]. Since the BPM life cycle includes the development of

a software system as an essential step, SE approaches are naturally utilized as part of the BPM activities. The development of a BPMS may differ from the traditional software development life cycle due to the use of process automation tools driven by process models rather than code [27]. Still, many SE practices are implemented during the development of a BPMS [28]. For example, process analysis is seen as an essential RE activity [29]. **Table 1** presents the common SE activities in the first column [30], and provides a summary of similarities and differences of these activities for SE and the software part of BPM (i.e. development of a BPMS).

Table 1. SE activities and their relation to software part of BPM

Activities	Software Engineering	Software Part of BPM
Requirement	BPs can be used for starting point of software requirements elicitation [31].	Requirements can be elicited traditionally or an agile approach can be used [32].
Design	Developing UML diagrams	Developing detailed process models and enriching process models with execution-related properties [33]
Implementation	Various implementation methodologies can be used.	Configuring process automation tools [27].
Testing	Manual or automated tests.	Manual or automated tests, flow analysis, and simulation [33]
Optimization	Refactoring the code.	Process improvement through redesign of processes [34]
Integration	Integrating/communicating with other systems via services.	Integrating/communicating with other systems via services [35]

In addition to the activities, roles in SE and BPM are also comparable. Despite the view that BPM roles are mostly related to business [36], many roles perform similar functions, such as domain experts, analysts, and developers [31]. It is essential that technical people are also involved in the development of BPMSs, and process models provide a good communication environment with them and non-technical ones [27].

2.2 The “Core” BPM

The activities at the intersection of BPM with SE do not constitute the main part of BPM. Actually, BPM is a discipline for which the focus is more on the humans and processes rather than technology. Technology, or the BPMS to be developed, is only a facilitator to improve processes. Process improvement can be achieved in many other ways such as implementation of various redesign heuristics [34]. BPM as a holistic approach covers six core factors; strategic alignment, governance, methods, information systems, people, and culture [37]. When we use the term “core BPM”, we refer to all these factors other than the software development part of BPM in this paper. These encompasses all the activities in BPM lifecycle including process analysis, design, implementation, monitoring, and improvement [38].

3 Agile Development Processes and BPM

3.1 Agile Values and Terminology

In recent years, a number of agile methods have emerged, all based on the core values stated in the Manifesto for Agile Software Development [1]. The Manifesto expresses the core values in the form of “Important Part” and “Less Important Part”. It should be noted, however, that the Manifesto does not deem the less important part to be unimportant. The following **Table 2** shows the applicability of the values to BPM.

Table 2. Agile values and their applicability to BPM

Value #	Important part	Less important part	Software Part of BPM	Core BPM
1	Individuals and interactions	Processes and tools	+~	~
2	Working software	Comprehensive documentation	~	~
3	Customer collaboration	Contract negotiation	++	+
4	Responding to change	Following a plan	+	??

The two rightmost columns in this table reflect our evaluation about the applicability of the agile values to both the software part of BPM as well as the core BPM. Value #1 cannot easily be adapted, since individuals and interactions are controlled by management who usually want to have formalized overviews of implementation progress. This problem can be assumed to be even harder to solve when core BPM facets, such as employee workflows, are considered as more parties are involved. This is, however, more a question of company culture than BPM itself, since software developed in a BPM context is intertwined with BPs that might be regulated by laws and that have to be transparent to management. Value #2 could be the value that is most difficult to apply for both the software development as well as the core part of BPM. BPM is closely connected to business concerns [28]. Hence, Value #3 should not pose any serious problems in neither part, as customer collaboration plays an essential role in BPM anyway. It could, however, be more problematic when third party systems or workforce personnel are involved. Responding to change (Value #4) can easily be followed in the software part of BPM, provided that is not hindered by problems resulting from the realization of Value #2. If documentation has to be updated and complex change processes involving many stakeholders have to be run before implementation changes, fast responses to change can become difficult. Realizing Value #4 in the core BPM part might prove even more difficult as the workforce is affected by change as well. Possible strategies could be to decompose work steps into minimalistic actions that could easily be recombined or to qualify an agile workforce with employees who can handle constant change. Both strategies might bring their own hurdles and would have to be explored in different organizational contexts.

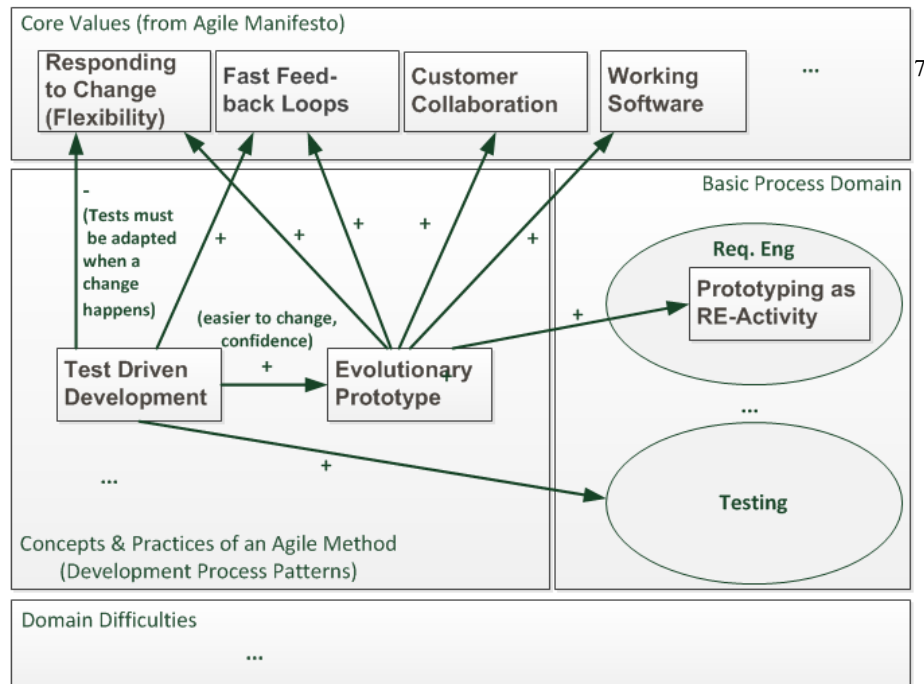


Fig. 1. Illustration on the interactions of agile development process patterns such as Test Driven Development or Evolutionary Prototype with other aspects.

3.2 Development of an agile method for BPM

Agile methods can be seen as a collection/language of process patterns [39, 40]. In this article, we use the term development process pattern (DPP) for what is referred to as process pattern in SE theory. Each practice of an agile method can be seen as an individual DPP, and all practices of an agile method can be rather seen as a DPP catalogue (or even pattern language) describing the patterns with their interdependencies/tradeoffs.

Fig. 1 shows a schematic illustration of agile DPPs and their interrelations/tradeoffs. The figure is separated into four dimensions:

At the top, the Core Values of agile methods expressed by the Agile Manifesto are listed. In the left-center area, the DPP-dimension shows typical agile methods to be analyzed for their tradeoffs. At the right area, the dimension of Basic Processes enlists the basic processes of any software development method (requirements engineering, design, implementation and testing as the basic process set every agile method must at least address in some way. At the bottom, the Domain Difficulties are indicated as a further dimension. As we will point out later, this dimension is an important aspect that must be considered in order to ensure an agile method can be applied in a certain domain because the DPPs used must also address these domain-specific difficulties.

In the DPP-dimension, the two DPPs “Test Driven Development” (TDD) and “Evolutionary Prototype” (EP) are shown as examples. The tradeoffs of each DPP are modeled by arrows, where arrows annotated with ‘+’ indicate a positive effect, whereas arrows annotated with ‘-’ indicate negative effects. In this way, e.g. TDD has

a positive effect on EP because changes can be made with less risk of undesired function side effects, as these would be discovered with automated testing, giving developers more trust in their code. These aspects are a fundamental basis to ensure that the EP DPP can work at all. It can also be seen that EP has positive effects to a lot of Core Values. The impact of TDD on the “Responding to Change” Core Value is two sided: TDD makes changes more work intensive as all test cases affected by the change have to be adapted but it also makes changes safer. TDD also has positive effects on other Core Values and EP and TDD also have positive effects on Basic Processes.

Fig. 1 gives an impression on how two DPPs interact with each other and the other dimensions discussed here. It could be argued that, if a working agile method with its DPPs is completely analyzed in this way, any aspect in any of the shown dimension should somehow be covered by positive effects, thus forming a synergistic network that is described above by the metaphor of interlocked cogwheels.

Concerning the implementation of BPs through IT-Systems (IT Business Alignment), a simple adaption of existing agile methods and techniques will not be successful because of the following factors:

- BPs are often essentially complex, so they need an essential amount of upfront planning (e.g. by modelling them)
- BPs are often highly interconnected with other BPs and different users, so it is very difficult to acquire fast feedback that is essential for agility
- BPs are often implemented into complex systems where workflow control is taken over by a workflow engine and the workflow triggers a number of other systems.
- Many BPs involve workforce personnel. People can hardly be involved in automated test procedures nor does it make sense to change people-based workflows too often and without proper change management.
- Legal constraints, business needs and organizational aspects do in some cases require higher degrees of documentation and planning.

These aspects could be seen as *domain difficulties* that should also be taken into account when considering agile methods. This follows the intuition that agile development is based on a set of assumptions that do not hold true in all kinds of projects [41]. Recent approaches have, however, successfully faced problems such as distributed development with organizational adaptations [42] or complex large scale projects with Large Scale Scrum [43]. So it might be possible to customize an agile method for BPM. There are some promising agile methods like TDD or EP that may at least in part be adaptable to implementation of BPs. However, this does not mean that this is a working agile method. To really establish an agile method, several of these principles must work together in a synergistic way addressing all important aspects to consider (Core Values, Basic Processes and Domain Difficulties). In this way, it might be possible to identify a set of development process patterns forming a promising agile method for the BPM-domain. To achieve this, **Fig. 1.** might indicate a kind of conceptual skeleton for evaluating the fit of agile methods to BPM.

4 Organizational Circumstances

4.1 Process Life Cycles and Disruptions

While the idea of process modeling and process management is to develop a stable environment for tasks, especially for transactional tasks, every process model has its life cycle. We can differentiate between planned and unplanned, but unavoidable process life cycles. These “unavoidable” life cycles result from their development process: Processes will be initiated, they have a phase of stability, and typically also a declining phase. In some cases, they have a “sudden death” caused by an unexpected change in the frame condition. The initial phase is the phase of establishing a process in an organization. Typically there are some uncertainties in this phase, for example an incomplete or wrong documentation or missing trainings. This phase is also a learning phase for individuals as well as for the whole organization. In the following phase of stability, the process can be operated successfully.

Typically over the long run major changes will influence the stability of the process. Some of these changes will only have a restricted influence, for example the substitution of one machine by another. In this case, the relevance and quality of the process description will decline. Other influences will be game changers and disrupt the traditional processes by replacing them.

In many organizations, “planned life cycles” as iterative development of processes are implemented. Those iterations refer to the Deming (PDCA) cycle [44] and reflect limited stability of the environment and internal learning processes. Changes in the Deming cycle are thought as an optimization of the existing, less as a reaction on (or the creation of) dramatic disruptive changes (caused for example by constitutional and legal changes or technologically driven changes).

4.2 Flexible Processes in the Post-Merger Integration

After mergers and acquisitions (M&A), most BPs are duplicated. During post-merger integration (PMI), an organization needs to decide how to proceed with doubled or redundant processes respectively and how to unify them in a common process map. An efficient process harmonization (PH) in a neutral and structured way with achieving the employee commitment promises a beneficial PMI. PH in the PMI context combines a common global management system across different regions or units with the allowances of defined variations at the process level [45].

Although PH has the highest relevance in the integration approaches absorption (acquired company is absorbed by acquirer) and symbiosis (evolution from existing), particular processes might be unified in preservation (acquired company retains independent) and holding (integration not intended) as well. Regardless of the various integration approaches, the need for action after an M&A is out of question. The question however is the intensity of PH.

While full benefit of process management is only apparent when the strategy, design, implementation and controlling of the processes are viewed holistically and coordinated with each other [46, 47], over-standardization with too strong regulations

would lead to a loss of necessary flexibility and competitive advantages. The right balance between standardization and individualization must be evolved. Process flexibility in context of PH means a combination of standardized processes with necessary variants - in other words: process flexibility respects specific and necessary process characteristics.

For such scenarios the “gate principle” is proposed as the best option. The “gate” variant is focused on a common (intermediate) output of a process necessary for the next process (step) (=gate), e.g. particular KPIs, reports, defined status etc. Process execution is secondary as long as the defined output is generated. The focus on a unified output allows a company keeping flexibility within a certain process combined with the advantage of an overall harmonized process map. In the long-term, the processes can be redesigned with the gate used as a requirement for agile process development. This stepwise unification gives flexibility in a tense situation, while it can be seen as an analogy of growing together to a unique organization.

5 Conclusion

In this discussion paper, we looked into the concepts of flexible processes in organizational environments, approaches for agile development of processes and management of flexible processes in an agile way, relationship between the areas of SE, RE, and BPM, and opportunities to incorporate agile principles in BPM. With the arising importance of agility in BPM field, this paper may inspire the researchers to initiate new works in these areas.

Different trends are currently influencing the requirements for BPM. On the one side, flexibility and agility are important to develop modern business models and they need adequate methods for planning and implementing as shown in this paper. On the other side, stability, transparency and documentation are important for those business processes that have strong legal constraints (indemnification). The parallelism of both trends and their combination will be a challenge for further research.

6 References

1. Beck K, Beedle M, Van Bennekum A, et al (2001) Manifesto for agile software development.
2. Weißbach R, Kirchner K, Reher F, Heinrich R Challenges in Business Processes Modeling – Is Agile BPM a solution ?
3. Conboy K (2009) Agility from first principles: reconstructing the concept of agility in information systems development.
4. Sethi AK, Sethi SP (1990) Flexibility in manufacturing: a survey. *Int J Flex Manuf Syst* 2:289–328.
5. Chen Y, Wang Y, Nevo S, et al (2014) IT capability and organizational performance: the roles of business process agility and environmental factors. *Eur J Inf Syst* 23:326–342.
6. Eardley A, Avison D, Powell P (1997) Strategic information systems: An analysis of development techniques which seek to incorporate strategic flexibility. *J Organ Comput* 7:57–77.

7. Doz YL, Kosonen M (2008) Fast strategy: How strategic agility will help you stay ahead of the game. Pearson Education
8. Teece D, Peteraf M, Leih S (2016) Dynamic Capabilities and Organizational Agility. *Calif Manage Rev* 58:13–35.
9. Singh J, Sharma G, Hill J, Schnackenberg A (2013) Organizational agility: What it is, what it is not, and why it matters. In: *Acad. Manag. Proc. Academy of Management*, p 11813
10. Fourné SPL, Jansen JJP, Mom TJM (2014) Strategic Agility in MNEs. *Calif Manage Rev* 56:13–38.
11. Triaa W, Gzara L, Verjus H (2016) Organizational agility key factors for dynamic business process management. In: *Bus. Informatics (CBI), 2016 IEEE 18th Conf. IEEE*, pp 64–73
12. Rudden J (2007) Making the Case for BPM-A Benefits Checklist. *BPTrends* 1–4.
13. Dadam P, Reichert M (2009) The ADEPT project: a decade of research and development for robust and flexible process support. *Comput Sci - Res Dev* 23:81–97.
14. Li C, Reichert M, Wombacher A (2010) The Minadept clustering approach for discovering reference process models out of process variants. *Int J Coop Inf Syst* 19:159–203.
15. Pesic M, van der Aalst WMP (2006) A Declarative Approach for Flexible Business Processes Management BT - *Business Process Management Workshops: BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006. Proceedings.* In: Eder J, Dustdar S (eds). Springer Berlin Heidelberg, Berlin, Heidelberg, pp 169–180
16. Hallerbach A, Bauer T, Reichert M (2009) Issues in Modeling Process Variants with Provop. In: Ardagna D, Mecella M, Yang J (eds) *Bus. Process Manag. Work. BPM 2008 Int. Work. Milano, Italy, Sept. 1-4, 2008. Revis. Pap.* Springer Berlin Heidelberg, Berlin, Heidelberg, pp 56–67
17. Torres V, Zugal S, Weber B, et al (2013) A Qualitative Comparison of Approaches Supporting Business Process Variability. *Bus Process Manag Work - BPM 2012 Int Work Tallinn, Est Sept 3, 2012 Revis Pap* 560–572.
18. Rosemann M, van der Aalst WMP (2007) A Configurable Reference Modelling Language. *Inf Syst* 32:1–23.
19. Hallerbach A, Bauer T, Reichert M (2010) Capturing Variability in Business Process Models: The Provop Approach. *J Softw Maint Evol Res Pract* 22:519–546.
20. Döhning M, Zimmermann B, Karg L (2011) Flexible Workflows at Design- and Runtime Using BPMN2 Adaptation Patterns. In: Abramowicz W (ed) *Bus. Inf. Syst. 14th Int. Conf. BIS 2011, Poznań, Poland, June 15-17, 2011. Proc.* Springer Berlin Heidelberg, Berlin, Heidelberg, pp 25–36
21. van der Aalst WMP, Weske M, Grünbauer D (2005) Case Handling: A New Paradigm for Business Process Support. *Data Knowl Eng* 53:129–162.
22. van der Aalst WMP, Barthelmess P, Ellis CA, Wainer J (2001) Proclets: A Framework for Lightweight Interacting Workflow Processes. *Int J Coop Inf Syst* 10:443–482.
23. OMG (2014) Case Management Model and Notation (CMMN) Version 1.0.
24. Von Rosing M, Von Scheel J, Gill AQ (2014) Applying agile principles to BPM. *Compleat Bus Process Handb Body Knowl from Process Model to BPM.*
25. Pourshahid A, Amyot D, Peyton L, et al (2009) Business process management with the user requirements notation. *Electron Commer Res* 9:269–316.
26. McCoy DW, Cantara M (2010) Hype Cycle for Business Process Management.

27. Dumas M, van der Aalst WMP, ter Hofstede AHM (2005) *Process-aware information systems : bridging people and software through process technology*. John Wiley & Sons, New Jersey
28. van der Aalst WMP (2013) *Business Process Management: A Comprehensive Survey*. *ISRN Softw Eng* 2013:1–37.
29. Aysolmaz B, Leopold H, Reijers HA, Demirörs O (2017) A semi-automated approach for generating natural language requirements documents based on business process models. *Inf Softw Technol*
30. IEEE Computer Society (2014) *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide V3.0)*. IEEE Computer Society
31. Heinrich R, Kirchner K, Weißbach R (2014) Report on the 1st International Workshop on the Interrelations between Requirements Engineering & Business Process Management (REBPM). *IEEE RE'14*
32. Aukema C (2011) *Managing Requirements In Business Process Management Suite Projects*. University of Twente
33. Dumas M, La Rosa M, Mendling J, Reijers HA (2013) *Fundamentals of business process management*. Springer
34. Reijers HA, Mansar SL (2005) Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega* 33:283–306.
35. van der Aalst WMP (2009) *Process-Aware Information Systems: Lessons to Be Learned from Process Mining*. In: Jensen K, Aalst WM (eds) *Trans. Petri Nets Other Model. Concurr. II*. Springer-Verlag, Berlin, Heidelberg, pp 1–26
36. Swenson K BPM is not Software Engineering. <https://social-biz.org/2008/11/25/bpm-is-not-software-engineering>. Accessed 4 Oct 2017
37. Brocke J Vom, Rosemann M (2010) *Handbook on business process management 1*. Springer-Verlag
38. Weske M (2012) *Business Process Management: Concepts, Languages, Architectures*, 2nd ed. Springer Publishing Company, Incorporated
39. Turban B (2013) *Tool-Based Requirement Traceability between Requirement and Design Artifacts*.
40. Bozheva T, Gallo ME (2006) Defining Agile Patterns. In: Dutoit AH, McCall R, Mistrík I, Paech B (eds) *Ration. Manag. Softw. Eng.* Springer Berlin Heidelberg, Berlin, Heidelberg, pp 373–390
41. Turk D, France RB, Rumpe B (2014) Limitations of Agile Software Processes. *CoRR abs/1409.6*:
42. Ramesh B, Cao L, Mohan K, Xu P (2006) Can Distributed Software Development Be Agile? *Commun ACM* 49:41–46.
43. Larman C, Vodde B Large-scale scrum : more with less.
44. Moen R, Norman C (2010) Evolution of the PDCA Cycle. <http://www.westga.edu/~dturner/PDCA.pdf>.
45. Fernandez J, Bhat J (1AD) Addressing the Complexities of Global Process Harmonization. In: *Handb. Res. Complex Dyn. Process Manag.* IGI Global, pp 368–385
46. Stähler D (2006) Standardisierung als Erfolgsvoraussetzung im Geschäftsprozessmanagement. *Zeitschrift Führung + Organ. ZfO* 75:
47. Becker J, Kahn D (2003) The Process in Focus. In: Becker J, Kugeler M, Rosemann M (eds) *Process Manag. SE - 1*. Springer Berlin Heidelberg, pp 1–12