Agile-Concurrent hybrid: A framework for concurrent product development using Scrum

Tena Žuže, Janez Kušar, Lidija Rihar and Tomaž Berlec

Abstract
Agile development became very popular at the beginning of the 21st century when the Manifesto for Agile Software Development was released. Since then, it has been predominant in the software industry and has been increasingly transferred to the development of physical products due to its great success. There are many studies on Agile-Stage-Gate hybrids that combine agile Scrum and the traditional Stage-Gate model, however no research has been found that addresses a direct integration of Scrum into a concurrent product development model in a similar way. In this paper, we therefore examine the possibility of introducing Scrum into the concurrent product development and propose a Scrum framework for an Agile-Concurrent hybrid. We propose that the framework for concurrent development remains unchanged: the stages overlap and the track-and-loop approach is preserved, while Scrum is proposed for the execution of day-to-day work. The main advantage of the proposed hybrid is that after each iteration the customer reviews the results of an entire loop, not just of one stage, which enables a broader understanding of the progress and facilitates a more extensive feedback. A quicker resolution of discrepancies, and a faster and more effective response to change is thus ensured. In the paper, the needed organizational changes and potential implementation issues are also discussed.

Keywords
concurrent product development, agile, Scrum, hybrid model, teamwork

Introduction
Today’s market is highly volatile, competition is fierce, and products have become increasingly complex. The traditional sequential product development has been replaced by concurrent development. The main characteristic of concurrent development is the overlapping of development stages (Rihar et al., 2010; Yang et al., 2012). With the introduction of concurrency, development times and costs are reduced, and the products are more innovative and of higher quality (Ganapathy and Goh, 1997; Kušar et al., 2004).

The use of overlapping activities has many advantages, however, it also leads to new risks due to incomplete information and uncertainty (Yang et al., 2012). The incompleteness of information and uncertainty can be managed using agile principles, such as constant customer collaboration and iterative development cycles. One of the most popular agile methods is Scrum, which was primarily designed for software development, but is increasingly being transferred to the field of physical product development (Sommer et al., 2015).

In recent years, hybrid models have been increasingly developed that integrate Scrum into the traditional Stage-Gate model and bring many benefits to the physical product development process, such as a faster response to change, better communication, a faster launch of products, decreased customer complaints, and more (Cooper and Sommer, 2016a, 2016b). So far, no research has been reported in the literature that would combine Scrum and the concurrent product development model in a similar way.

The main purpose of this paper is to link the research on concurrent engineering and agile product development, and to examine the possibility of introducing Scrum into the concurrent product development model.
A framework for the Agile-Concurrent hybrid is proposed that combines the benefits of the agile Scrum and the concurrent track-and-loop principle.

The rest of the paper is composed as follows. Section “Literature review” is a literature review on concurrent engineering and agile development. Also, some existing hybrid models are presented and the needed modifications for applying Scrum into physical product development are outlined. Section “Agile development and Scrum” presents a Scrum framework for the proposed Agile-Concurrent hybrid, and discusses some potential implementation issues. The paper ends with discussions and conclusions that outline the main advantages and limitations of the model and present directions for future research.

Literature review

Concurrent engineering

In traditional product development, the stages are strictly sequential—each stage only starts, when the one before has completely finished, resulting in a lack of integration and coordination between different functional areas (Valle and Vázquez-Bustelo, 2009). Due to a lack of communication and understanding between product design, production, and customers’ needs, many quality problems arise, development times are very long and costs are high (Valle and Vázquez-Bustelo, 2009). As a result, traditional development has been replaced by a new practice, named concurrent engineering.

The term concurrent engineering first appeared in the late 1980s. Winner et al. (1988) defined the concurrent engineering as “…a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support. This approach is intended to cause the developers from the very outset to consider all elements of the product life cycle, from conception to disposal, including quality, cost, schedule, and user requirements.”

Sohlenius (1992) indicated the increase in competitiveness as the major endeavor of concurrent engineering by reducing flow times, lowering costs, and increasing quality. Product and process planning must go hand in hand and should largely be carried out in parallel. For lowering the complexity of new product development, Prasad (2001) proposes decomposing it into smaller concurrent sets, which are easier to deal with and solve. He proposes a four-stage product, process, and methodology systematization method for an effective product realization, and stresses the importance of exploiting the so called 7Ts resources: talents, tasks, teamwork, techniques, technology, time, and tools.

Unlike the traditional development, where the stages are strictly sequential, stages in the concurrent development overlap. Yang et al. (2012) define overlapping as the process of starting a downstream activity before completing the upstream activity. The information builds up gradually and is constantly transferred between activities (Kušar et al., 2014). Overlapping reduces project duration, increases coordination, communication, and interactions between different departments (Yang et al., 2014). The communication and information exchange can be facilitated through the use of a DSM (Design Structure Matrix), which helps identifying both the needed intensity and frequency of information flow among elements (Yassine and Braha, 2003).

A track-and-loop principle has been developed to perform the interactions between the overlapping stages (Prasad, 1996). The type of a loop is indicative of the number of consecutive stages that are involved in executing each loop. Winner et al. (1988) suggest using 3-T loops, which means that three consecutive stages overlap and interact.

In concurrent product development, the project teams are multidisciplinary, which contributes to a good understanding of the broader problem and early detection of potential discrepancies. In the project team, the team leader is a standing member, while the other members change depending on the development loop underway (Kušar et al., 2004). Concurrent engineering also recognizes the benefits of both customer and supplier integration in the process, so that the customer requirements can be finalized more quickly and efficiently (Bhuiyan et al., 2006; Rihar et al., 2010). For concurrent development to be successful, an effective communication and active collaboration between all the stakeholders is a must (Prasad, 1996; Rihar et al., 2010). This can be achieved through frequent meetings and constant information exchange, using shared data environments and different ICTs (Bhuiyan et al., 2006).

Agile development and Scrum

Companies have become increasingly aware of the fact that in today’s fast-paced and competitive market, flexibility and speed are of utter importance. The traditional product development no longer allows for the development of successful and competitive products (Takeuchi and Nonaka, 1986). We are facing a transition from deterministic-normative development processes to agile processes (Schuh et al., 2017).

Since 2001, when the Manifesto for Agile Software Development was written (Beck et al., 2001), agile development has dominated the software industry. The basis of the manifest are four core values: (1)
individuals and interactions over processes and tools; (2) working software over comprehensive documentation; (3) customer collaboration over contract negotiation; and (4) responding to change over following a plan.

Agile development is an incremental iterative approach. Instead of an in-depth planning at the beginning of the project like in traditional approaches, agile development embraces changes and adapts the project plan accordingly (Ullman, 2019). The definitions on agility and agile development still differ, but most authors agree that it is “... an approach that seeks flexibility, simplicity, iterations in short period of time, and incrementally add value” (Azanha et al., 2017).

There are many different agile methods available for software development, among the most popular being: Scrum, Kanban, eXtreme Programming (XP), Crystal methods, and others (Atzberger and Paetzold, 2019). All the methods share the same idea of continuous value delivery and responsiveness to changes, but differ in the depth of guidance and breadth of the life cycles. The most frequently applied agile method is Scrum (Gartzen et al., 2016). In comparison to other methods, Scrum is relatively simple and straightforward. Many authors also recognize Scrum as the agile method having the most applicability to physical products (Cooper and Sommer, 2016b; Ullman, 2019).

In Scrum, the development process is split into time boxes called sprints. The duration of sprints is fixed, typically 2 to 4 weeks (depending on product complexity and risk assessment) (Schwaber, 1997), and each sprint results in a useful intermediate version of a product to be reviewed by the customer.

Each sprint begins with spring planning. Project requirements, listed in the product backlog, are divided to individual tasks and assigned priorities. The highest priority tasks are transferred to the sprint backlog and must be completed in the selected sprint. The deliverables of the sprint need to be clearly defined to allow customer feedback. During the planning stage, the project backlog is also updated as necessary if the customer has new requests or if environmental changes occur. Note that the requirements are usually not listed directly, but are rather defined through user stories. User stories describe, how the final product will fulfill the user’s requirements in practice (Cooper and Sommer, 2016a).

Scrum is known for daily 15-min stand-up meetings called daily scrums, held in the same location and at the same time each day. At a daily meeting, team members present what they accomplished on the previous day, what they will be working on today, and discuss possible obstacles in achieving the goal. The sprint progress is presented on the Scrum board which is constantly updated. The progress can be also monitored via a burndown chart (Cooper, 2016; Cooper and Sommer, 2016a).

At the end of each sprint, the team delivers the result of the sprint to the customer. The customer reviews the product, tests it as needed, and provides feedback and any changed or new requests. Regular customer involvement provides for a continuous review of the actual progress, which prevents misunderstandings in subsequent stages. This step is called a sprint review.

The sprint is closed with the sprint retrospective. The team reflects on the achievements of the cycle and considers possible improvements and changes in the next sprint.

In Scrum, three different roles are defined: Scrum master, product owner, and development team (Cooper and Sommer, 2016a). The Scrum master sees to it that the process runs smoothly and that everyone understands and obeys the rules of Scrum. The product owner arranges for the product development to be focused on the customer’s goals and requirements, directs the work of the team and ensures that all team members understand the tasks. The product owner’s job is also to define product backlog items. The development team is made up of professionals from different fields. Team members must work well together and be focused on the same goal, which is achieved through daily meetings and process transparency.

Another important feature of the Scrum team is self-organization (Hoda and Murugesan, 2016; Hoda et al., 2013). Team members are empowered with the autonomy to make collective decisions, they manage their own workload, and shift the work among themselves as they see fit (Hoda et al., 2013). The empowerment of the team members gives them a sense of ownership, and therefore motivates them to do their best (Cooper and Sommer, 2016b).

The use of Scrum and other agile approaches encourages the development team, including the customer, to produce the finished product as quickly as possible. Team communication and collaboration are improved, and team members’ productivity and morale are increased. Quick feedback from the customer significantly reduces the level of uncertainty, and the necessary changes are detected at the right time. This results in better project efficiency (meeting cost, time, and scope goals), and increased stakeholder satisfaction (Serrador and Pinto, 2015).

Agile physical product development

Traditional deterministic-normative processes are still widely used in the development of physical products, but they are slow, rigid and too bureaucratic (Schuh et al., 2017). In the last decade more attention has been
paid to transferring agile methods and good practices from the software industry to the development of physical products (Conforto and Amaral, 2010, 2016; Cooper, 2014, 2016; Cooper and Sommer, 2016a, 2016b; Schuh et al., 2016a, 2016b, 2017; Sommer, 2019; Sommer et al., 2013, 2015). The use of agile methods for the development of physical products has largely been made possible by modern technologies such as powerful computer tools, simulations, and 3D printing (Cooper, 2016).

Adopting agile methods does not necessarily mean abandoning traditional development models. In recent years, interest in the so-called hybrid models that exploit the benefits of agile development without abandoning the stability of traditional models (Ciric et al., 2018) has increased. According to Cooper and Sommer (2016b), a combination of agile approaches and traditional models has significant potential benefits for physical product manufacturers.

Boehm et al. (2012) propose a risk-driven framework for determining and evolving best-fit system lifecycle process, called ICSM (Incremental Commitment Spiral Model). The framework combines the benefits of both agile and traditional plan-driven methods; it empowers the team, introduces incremental and iterative development, favors the principle of concurrent rather than sequential work, and emphasizes evidence- and risk-based decision-making. In his paper, Cooper (2014) elaborates on a next generation of the Stage-Gate model that should be more adaptive, agile, and accelerated. Based on Cooper’s paper, Sommer et al. (2015) propose an industrial Scrum framework, a hybrid that combines Stage-Gate model on the strategic level, with agile Scrum implemented on the execution level. Conforto and Amaral (2010) propose a management framework entitled IVPM2 (Iterative and Visual Project Management Method) that combines the elements of Stage-Gate model, such as phase definition, standardized documents, and phase checkpoint reviews with agile approaches, such as iterative cycles, visual boards, and frequent meetings. Schuh et al. (2016b) suggest an adapted Scrum process model for highly iterative innovation processes for a fast realization of physical product ideas. They suggest a product lifecycle management (PLM) system, an integrated ICT infrastructure, interdisciplinary engineering teams and scalable manufacturing technologies as the key enablers. They also note, that parallel sprints, each focusing on a specific product increment, could potentially accelerate the development. In his book, Ullman (2019) presents a Scrum framework customized specifically for hardware design as a supplement to the traditional waterfall, serial design methodology.

The most frequently applied agile method for physical product development is Scrum. According to various authors, it has the highest potential for impact on physical product development of all the agile methods (Cooper and Sommer, 2016b; Ullman, 2019). In most cases, it is combined with the traditional Stage-Gate model. The studies show that the main advantages of these Agile-Stage-Gate hybrids are better voice-of-customer integration, faster response to changes, improved team communication and productivity, and faster product launch (Cooper and Sommer, 2016a; Ullman, 2019).

However, due to some discipline-specific differences and higher level of complexity, a direct transmission of Scrum toward physical product development is neither realizable nor beneficial (Schuh et al., 2016a). The first major difference is in immateriality of software, which enables a frequent delivery of a working product (a code), while it is virtually impossible to release functional and potentially marketable physical product this often (Cooper, 2016). Therefore, a deliverable or “done sprint” needs to be redefined. A deliverable in physical product development should be a version of a product, from a concept to a prototype—a protocept (Cooper, 2014), which can be demonstrated or is suitable for testing. A protocept can be anything from a computer 3D drawing, a virtual prototype, to a working model. The most important thing is to clearly define both the sprint deliverable and the sprint success criteria, and to set the goals realistically so that they are achievable during the sprint cycle (Ullman, 2019).

Second, agile software development exploits the modular architectures that allow the breaking down of the code into smaller independent modules. Physical products are much harder to modularize, since there are many inter-dependencies between different functions (Atzberger and Paetzold, 2019; Ullman, 2019). One possibility is to apply the DSM method that enables finding mutually exclusive or minimally interacting modules and also identifies the interactions or links between them (Yassine and Braha, 2003). DSM is a well-known and established tool in concurrent product development and could also be of a great help when introducing agile methods.

Third, when integrating Scrum, there are some project role changes needed, which may require extensive reorganization and training. The results, however, show that the reorganization can lead to improved product development performance and employee motivation (Sommer et al., 2013).

Another challenge in physical product development is forming a dedicated and co-located project team. For the best results, all the team members should only work on one project at the time (project organizational structure) and should be located in the same room, however, this is usually not possible in manufacturing companies. Often, compromises are made, so that
certain team members are dedicated for only a certain part of the project (Cooper and Sommer, 2016b). Instead of a project organizational structure, a matrix structure should be introduced. The problem of co-location can be solved using advanced ICTs and shared data environments, but if possible, co-located project teams are still highly recommended, since a face-to-face communication and productivity are facilitated (Cooper and Sommer, 2018).

Physical product development also requires more specialization than software development. While there is a great possibility that several individuals can execute the same task when developing software, in physical product development the expertise is not that universal across functions (Ullman, 2019). This can result in various bottlenecks. Therefore, the need for employing or training the so-called T-shaped specialists is recognized, who are experts in their own field, and at the same time have a general knowledge from other domains (Demirkan and Spohrer, 2015; Ullman, 2019).

Despite some challenges in integrating Scrum, companies are reporting promising results with the introduction of Agile-Stage-Gate systems, and to quote Cooper (2016): “... this new Agile-Stage-Gate hybrid model may be the most exciting and significant change to the new-product process since the introduction of gating systems more than 30 years ago.”

**Agile Concurrent product development**

In general, concurrent and agile development share some similar ideas and practices. First similarity is in a great importance of a high degree of a two-way communication between all the stakeholders. Secondly, both approaches recognize the benefits of multidisciplinary teams being established early on, and collaborating throughout the whole project (Bhuiyan et al., 2006). This prevents down-stream problems from occurring, as the incompatibilities are recognized early in the process. Testing, simulations, and prototyping are established tools in both approaches. Also, the benefits of customer and supplier integration are recognized to finalize the product requirements more quickly and effectively (Bhuiyan et al., 2006; Dahmas et al., 2019).

However, agile development has also some specific characteristics that could benefit concurrent development. First and foremost, agility emphasizes adaptability and embracing changes even in the late project stages. Active customer integration and frequent incremental value delivery allow for a better understanding of requirements and rapid response to changes. The scope is no longer fixed, it is rather adapted when new information and knowledge is available. Thus, the customer is provided the product he or she actually needs. The second important difference is in the extent of planning. While in concurrent development a project plan is made at the beginning of the project, an agile approach keeps planning to a minimum and adapts the plan iteratively. One could therefore think of concurrent approach as a macroplanning and of agile approach as a microplanning. Concurrent development can also be seen as a process-oriented approach that is mostly focused on project efficiency: shorter development time, lower cost and better quality, while an agile approach is more people-oriented, and emphasizes stakeholders’ success and satisfaction.

**Framework for Agile-Concurrent hybrid using Scrum**

In this section, we present a framework for Agile-Concurrent hybrid, where we integrate Scrum into a concurrent product development model. We propose that the concurrent development model remains unchanged and provides a formal structure of the process. The stages overlap and the track-and-loop principle is applied. Also, a changing team composition depending on the loop underway is preserved.

Scrum is proposed for the execution of the stages and loops to increase flexibility and responsiveness to changes. Some changes in day-to-day work organization are introduced (sprints, daily stand-ups, iterative planning, etc.), the customer actively collaborates and provides feedback on a regular basis, the scope is adaptable, and the project team is self-organizing and empowered to make decisions. Also, the team roles distinctive for Scrum are introduced.

The Scrum framework is shown in Figure 1, and the proposed Agile-Concurrent hybrid in Figure 2. The concurrent model is taken from Kušar et al. (2004), and as suggested by Winner et al. (1988), 3-T loops are applied. The deliverables of each stage and the milestones that represent the completion of an individual loop are also shown.

In the proposed hybrid, the project teams of individual loops are multidisciplinary. Within the project team, three Scrum development teams are formed (Figure 1), where each team is responsible for a deliverable of one of the stages participating in the loop. Effective communication and continuous information exchange between teams is achieved through daily meetings, shared data environments, and other appropriate ICTs. The communication and integration can also be facilitated through the use of a team-based DSM, which helps identifying the needed intensity and frequency of information flow among teams (Yassine and Braha, 2003).

A certain team member must be trained as Scrum Master so that he or she can manage and direct the development process, provide information, and ensure
compliance with the Scrum rules. In the Agile-Concurrent model, the project manager assumes the role of a product owner. The project manager (now product owner) directs team work, ensures that product development is focused on the customer’s goals and requirements, and is responsible for designing the
product backlog. With the introduction of Scrum, day-to-day decision-making is largely outsourced to self-organizing development teams.

One distinctive characteristic of the proposed Agile-Concurrent hybrid is the changing composition of a project team. In the proposed model, the product owner and the Scrum master are standing members of the project team, while the development teams change depending on the loop underway. With a changing team composition, both the needed specialization and multidisciplinary is ensured throughout the whole project. For example, in the feasibility loop, the development teams of the first three stages participate: goal-setting team, concept development team, and design team. In the design loop, the concept development team, and the design team still co-operate, while a process planning team joins the project team instead of the goal-setting team. In the same way, the composition changes in all the following loops.

With the introduction of Scrum, product development stages are divided into sprints that are 2 to 4 weeks long. The sprints of three consecutive stages within a single loop combine in the Agile-Concurrent hybrid into a so-called Sprint loop. A product increment, or the sprint deliverable, is defined similarly as in Agile-Stage-Gate hybrid: it represents some intermediate product version, a protocoept, that can be shown to the customer to seek feedback. The main difference of Agile-Concurrent hybrid is, however, that the project team works on three deliverables concurrently. Each development team is responsible for the deliverable of its own stage, while all the development teams together also share the responsibility for the result of the whole loop. For example, in the feasibility loop, the development teams work concurrently on the goals, concept, and design (see Figure 2). And while the goals are the main deliverable of the first loop, the concept and the design are also being developed (e.g. first sketches) and can be reviewed by the customer. The same goes for all the other loops.

At the beginning of each sprint, the entire project team executes planning of a loop sprint. Based on the product backlog, the highest priority tasks are transferred to the sprint backlog. The project team is self-organizing and internally divides the loop tasks between individual development teams and precisely defines their deliverables. A task-based DSM can be used to find the independent tasks that can run totally parallel, tasks that need to be executed sequentially, and coupled tasks that need to be constantly coordinated (Yassine and Braha, 2003). The entire project team is located in the same room where there is a single Scrum board with a displayed burn-down chart of the loop. With advanced communication technologies available, collocation is not a must, but if possible, it is still highly recommended.

All members of the project team participate in daily Scrum. The work completed is reviewed, and the daily goals and potential problems are presented. All development teams are constantly updated on the progress of the entire loop and discrepancies between individual deliverables can be detected and resolved on a daily basis. At the end of a sprint, the customer reviews the results of the entire loop and provides feedback. If needed, the product backlog is updated. Each sprint closes with a joint sprint loop retrospective, where the entire project team reviews the cycle's achievements and revises the achievement of the plan and the set goals of the loop.

The main advantage of Agile-Concurrent hybrid over Agile-Stage-Gate hybrids is in the fact that after each sprint, the customer reviews the result of an entire loop (the interim results of three successive stages). This gives customer a broader and better idea of the progress of the project and facilitates feedback that is more extensive. Uncertainty is thus significantly reduced and a quicker resolution of misunderstandings is enabled. With a more comprehensive insight into development, the customer can more rapidly formulate the final requirements, which leads to a lower number of required changes in subsequent stages, where corrections would be more expensive and time consuming.

**Potential implementation issues**

Scrum concepts give concurrent product development more flexibility and enable rapid response to change. However, implementing this new agile approach into a company can be very daunting (Cooper and Sommer, 2018).

First of all, all the challenges described in the section Agile Physical Product Development must be taken in consideration: the re-definition of deliverable, modularization, team role changes, forming a dedicated and co-located project team, and a higher need for specialization.

As people are in general very reluctant to change, another great challenge is to establish an agile mindset in the heads of all the stakeholders and willingness to accept this new working style (Atzberger and Paetzold, 2019). The employees must undergo extensive trainings, and in many cases, an outside expert must be employed.

For Scrum implementation to be successful, there is also a need for a full support and trust from the management (Bergmann and Karwowski, 2019). Cooper and Sommer (2018) state that management skepticism is one of the biggest challenges when implementing Scrum. As the project teams are empowered to make
their own decisions, management can feel as they have lost the power. This issue is also referred to as the "prince problem" (Atzberger and Paetzold, 2019).

Another potential challenge is to ensure a constant customer involvement and collaboration. Commonly, customers are used to work according to the traditional models and do not see the benefit of providing frequent feedback on interim results (Atzberger and Paetzold, 2019), so they might be reluctant to actively participate.

**Discussion and conclusion**

The success of agility in the software industry has triggered numerous attempts to transfer agile methods to the development of physical products. Over the last decade, research in this field has grown enormously and hybrids between agile and traditional models have been increasingly developed. The studies show that these hybrid models yield very promising results, such as faster and more flexible response to changes, improved productivity, and shorter time to market.

In this paper, the possibility of introducing Scrum into the concurrent product development model was examined. A conceptual framework for an Agile-Concurrent hybrid is proposed that combines the benefits of agile Scrum and the concurrent track-and-loop principle. Even though the model is only conceptual, it represents the first attempt of introducing Scrum into concurrent product development and thus provides a novel link between the research on agility and concurrency. The paper presents some basic ideas and guidelines that can facilitate the agile transformation of the companies practicing concurrent product development (e.g. team role changes, organization of work, etc.).

We propose that the framework for concurrent development remains unchanged: the stages overlap and the track-and-loop approach is preserved, while Scrum is proposed for the implementation of the stages and loops. Introducing Scrum into concurrent product development ensures constant customer collaboration and a faster response to changes. It is precisely this flexibility of the company to accept change (even in subsequent stages) as something positive and as an opportunity for progress that represents the main advantage of the introduction of agility and thus a great competitive advantage for the company.

The biggest advantage of the proposed Agile-Concurrent hybrid over other hybrid models is in the overlapping of the stages and the track-and-loop principle. Regular reviews of the results of an entire loop give customer a broader and better idea of the progress of the whole project, which facilitates more extensive feedback and faster formulation of the final requirements.

This research has, however, several limitations that need to be addressed. First, the framework needs to be tested in practice. The advantages of agility and concurrency theoretically complement each other, but this still needs to be empirically researched so that this assumption can be supported by data.

Second, it is necessary to investigate which projects this approach is the most appropriate for, and to analyze the benefits of introducing Scrum into a single loop. This approach is expected to be the most beneficial for radical development projects, where there is a high need for innovativeness, a high rate of change, and where the requirements and needs manifest during development itself and get more clear gradually. While we believe Scrum can be successfully integrated in all the development loops, the benefits should be most evident in the technical ones (e.g. design loop). For the other loops it might be reasonable to examine the trade-off between benefits and effort of implementing Scrum.

Third, Scrum generally demands dedicated and collocated project teams, however, in new product development, distributed teams can present a great competitive advantage. Therefore, future research needs to examine a possibility of introducing Scrum into a distributed concurrent product development. Both the potential obstacles and advantages of the distributed Agile-Concurrent hybrid need to be addressed.

With this paper, we hope to have provided a solid conceptual basis and some useful starting points for further discussion and research in the academic sphere and for the practitioners that would like to take on agile transformation. We believe that, if integrated successfully, the Agile-Concurrent hybrids could lead to significantly improved project success in terms of efficiency (time, cost, and quality), and also stakeholders’ satisfaction.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Ministry of Higher Education, Science and Technology of the Republic of Slovenia, grant no. 1000-15-0510, and by the Slovenian Research Agency, grant no. P2-0270.

**ORCID iD**

Tena Zužek [1] https://orcid.org/0000-0003-3389-0110  
Lidija Rihar [2] https://orcid.org/0000-0003-2272-351X
References

Atzberger A and Paetzold K (2019) Current challenges of agile hardware development: what are still the pain points nowadays? In: Proceedings of the 22nd international conference on engineering design, Delft, Netherlands, 5–8 August, pp. 2209–2218. Cambridge, England: Cambridge University Press.

Azanha A, Argoud ARTT, de Camargo Junior JB, et al. (2017) Agile project management with scrum: a case study of a brazilian pharmaceutical company IT project. International Journal of Managing Projects in Business 10(1): 121–142.

Beck K, Sutherland J, Fowler M, et al. Manifesto for agile software development. Available at: https://agilemanifesto.org/ (2001, accessed 21 August 2019).

Bergmann T and Karwowski W (2019) Agile project management and project success: a literature review. Advances in Intelligent Systems and Computing 783: 405–414.

Bhuiyan N, Thomson V and Gerwin D (2006) Implementing concurrent engineering. Research-Technology Management 49(1): 38–43.

Boehm B, Koolmanojwong S, Lane JA, et al. (2012) Principles for successful systems engineering. Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, pp. 117–126. New York City, US: IEEE.

Conforto EC and Amaral DC (2010) Evaluating an Agile method for planning and controlling innovative projects. Project Management Journal 41(2): 73–80.

Conforto EC and Amaral DC (2016) Agile project management and stage-gate model – a hybrid framework for technology-based companies. Journal of Engineering and Technology Management 40: 1–14.

Cooper RG (2014) What’s next? Ater stage-gate. Research-Technology Management 57(1): 20–31.

Cooper RG (2016) Agile-stage-gate hybrids. Research-Technology Management 59(1): 21–29.

Cooper RG and Sommer AF (2016a) Agile-stage-gate: new idea-to-launch method for manufactured new products is faster, more responsive. Industrial Marketing Management 59: 167–180.

Cooper RG and Sommer AF (2016b) The Agile-stage-gate hybrid model: a promising new approach and a new research opportunity. Journal of Product Innovation Management 33(5): 513–526.

Cooper RG and Sommer AF (2018) Agile-stage-gate for manufacturers. Research-Technology Management 61(2): 17–26.

Dahmas S, Li Z and Liu S (2019) Solving the difficulties and challenges facing construction based on concurrent engineering in yemen. Sustainability 11(11): 3146.

Demirkan H and Spohrer J (2015) T-shaped innovators: identifying the right talent to support service innovation. Research-Technology Management 58(5): 12–15.

Ganapathy B and Goh CH (1997) A hierarchical system of performance measures for concurrent engineering. Concurrent Engineering: Research and Applications 5(2): 137–143.

Gartz T, Brambring F and Basse F (2016) Target-oriented prototyping in highly iterative product development. Procedia CIRP 51: 19–23.

Hoda R and Murugesan L (2016) Multi-level agile project management challenges: a self-organizing team perspective. Journal of Systems and Software 117: 245–257.

Hoda R, Noble J and Marshall S (2013) Self-organizing roles on agile software development teams. IEEE Transactions on Software Engineering 39(3): 422–444.

Kušar J, Duhovnik J, Grum J, et al. (2004) How to reduce new product development time. Robotics and Computer Integrated Manufacturing 20(1): 1–15.

Kušar J, Rihar L, Duhovnik J, et al. (2014) Concurrent realization and quality assurance of products in the automotive industry. Concurrent Engineering: Research and Applications 22(2): 162–171.

Prasad B (1996) Concurrent Engineering Fundamentals Volume I: Integrated Product and Process Organization, 1st edn. New Jersey: Prentice Hall PTR.

Prasad B (2001) Product, process and methodology systematization to handle structural and computational complexity in product realization. Systems Research and Behavioral Science 18(6): 523–543.

Riha L, Kušar J, Duhovnik J, et al. (2010) Teamwork as a precondition for simultaneous product realization. Concurrent Engineering: Research and Applications 18(4): 261–273.

Schuh G, Gartz T, Soucy-Bouchard S, et al. (2017) Enabling agility in product development through an adaptive engineering change management. Procedia CIRP 63: 342–347.

Schuh G, Gartz T, Basse F, et al. (2016b) Enabling radical innovation through highly iterative product expedition in ramp up and demonstration factories. Procedia CIRP 41: 620–625.

Schuh G, Riesner M and Diels F (2016a) Structuring highly iterative product development projects by using HIP-indicators. In: Proceedings of the 2016 IEEE IEEM. 2016 IEEE international conference on industrial engineering and engineering management, Bali, Indonesia, 4–7 December, pp. 1171–1175. New York City, US: IEEE.

Schwaber K (1997) Scrum development process. In: J Sutherland, C Casanave, J Miller, et al. (eds) Business Object Design and Implementation. London: Springer, pp. 117–134.

Serrador P and Pinto JK (2015) Does agile work? – a quantitative analysis of agile project success. International Journal of Project Management 33(5): 1040–1051.

Sohlenius G (1992) Concurrent engineering. Annals of the CIRP 41(2): 645–655.

Sommer AF (2019) Agile transformation at LEGO group. Research-Technology Management 62(5): 29–29.

Sommer AF, Hedegaard C, Dukovska-Popovska I, et al. (2015) Improved product development performance
through agile/atage-gate hybrids. Research-Technology Management 58(1): 34–45.
Sommer AF, Slavensky A, Nguyen VT, et al. (2013) Scrum integration in stage-gate models for collaborative product development – a case study of three industrial manufacturers. In: Proceedings of the 2013 IEEE IEEM. 2013 IEEE international conference on industrial engineering and engineering management, Bangkok, Thailand, 10–13 December, pp. 1278–1282. New York City, US: IEEE.
Takeuchi H and Nonaka I (1986) The new product development game. Harvard Business Review 64(1): 137–146.
Ullman DG (2019) Scrum for Hardware Design. Oregon: David Ullman LLC.
Valle S and Vázquez-Bustelo D (2009) Concurrent engineering performance: incremental versus radical innovation. International Journal of Production Economics 119(1): 136–148.
Winner RI, Pennell JP, Bertrand HE, et al. (1988) The Role of Concurrent Engineering in Weapons System Acquisition, IDA Report R-338. Alexandria, VA: Institute for Defense Analysis.
Yang Q, Yao T, Lu T, et al. (2014) An overlapping-based design structure matrix for measuring interaction strength and clustering analysis in product development project. IEEE Transactions on Engineering Management 61(1): 159–170.
Yang Q, Zhang X and Yao T (2012) An overlapping-based process model for managing schedule and cost risk in product development. Concurrent Engineering: Research and Applications 20(1): 3–17.
Yassine A and Braha D (2003) Complex concurrent engineering and the design structure matrix method. Concurrent Engineering: Research and Applications 11(3): 165–176.

Author biographies

Tena Žužek is a PhD student and a young researcher in the Laboratory for Manufacturing Systems and Production Process Planning at Faculty of Mechanical Engineering, University of Ljubljana. In 2015, she received a bachelor’s degree in physics and in 2017, a master’s degree in mechanical engineering from University of Ljubljana. Her research interests include project management, concurrent engineering, agility and risk management. e-mail address: Tena.Zuzek@fs.uni-lj.si

Janez Kušar is an associate professor at Faculty of Mechanical Engineering, University of Ljubljana, Slovenia. He took his PhD in 1999 in the field of production planning and control. His research fields include: project management, especially in individual and small-series production, material flow optimization, production planning and control systems, and projects of transition from sequential to concurrent engineering. e-mail address: Janez.Kusar@fs.uni-lj.si

Lidija Rihar graduated from Faculty of Mechanical Engineering in Ljubljana on a field of production systems in 2009. In 2013, she defended her doctoral dissertation entitled Generalized model of concurrent product and process development. Since 2013 she is a research assistant at the Faculty of Mechanical Engineering in the field of production systems. e-mail address: Lidija.Rihar@fs.uni-lj.si

Tomaz Berlec, PhD is an assistant professor at Faculty of Mechanical Engineering, University of Ljubljana, Slovenia. He made his M. Sc. degree in 2003 and Ph.D. degree in 2008 in the field of production planning and control. His research fields include: lean production, teamwork, material flow optimisation in companies with individual and small-series production, lead times optimisation of operations and orders, production planning and control systems. e-mail address: Tomaz.Berlec@fs.uni-lj.si