NordDesign 2018
August 14 – 17, 2018
Linköping, Sweden

Scaling agile practices on different time scopes for complex problem-solving

Jan Breitschuh, Albert Albers, Patrick Seyb, Sophie Hohler, Jonathan Benz, Nicolas Reiß and Nikola Bursac

Karlsruher Institut für Technologie (KIT) IPEK – Institut für Produktentwicklung /Jan.Breitschuh, Albert.Albers, Patrick.Seyb, Sophie.Hohler, Jonathan.Benz, Nicolas.Reiss and Nikola.Bursac)@kit.edu

Abstract

There can be no innovation without time and space for ideation and courage to endeavor new terrain. Approaching this terrain in a structured way whilst managing the risks linked to the uncertainty of the novel is a major advantage of agile process models such as Scrum. On the other hand, most companies in mechatronic product development organize their activities in Stage-Gate-Processes for good reasons. This paper thus aims at combining the benefits of traditional and agile process models. The core assumption is, that even in overall complex projects, only a certain amount of tasks really benefits from agile practices.

In order to identify these project elements, a differentiated view on project complexity is necessary. This differentiation is then integrated into a tool for analyzing task entropy as a measure of unknowingness and thus potential for agile approaches.

These agile approaches are considered to be timebound episodes of concentrated problem solving with restricted ressources. Thus, theories about human problem solving and multitasking serve as a fundament for the conceptualization of short-term agile workshops.

By restricting the duration of these workshops to two to five days, the barrier of practicing agile methods in arbitrary process landscapes is significantly lowered. The question arising is how proven agile practices can be scaled to these small time scopes while retaining the valuable, structuring elements such as fixed sprints and regular meetings.

On the fundament of ASD – Agile Systems Design, this paper presents guidelines for implementation of agile workshops on smaller time scopes based on three pillars: 1) using a structured agile process, 2) using methods and tools in an agile way and 3) providing an agile moderator. Subsequently, an exemplary implementation of the concept is shown.

This paper thus contributes to actually creating innovation by describing a systematic way to generate results whilst containing risks under conditions of limited ressources.

Keywords: ASD – Agile Systems Design, Scrum, agile moderation, EntropyCompass, problem-solving, ASD-FIT

1 Introduction

The velocity and unpredictability of change – often referred to as “increasing complexity” – is a significant challenge for traditional product engineering processes. Agile process models claim to provide useful guidance when facing rapidly changing projects and prove this claim empirically (Rigby, Sutherland, & Takeuchi, 2016; Versionone, 2017; Weckmüller, 2017). Unfortunately, industrial engineering of physical products faces some limits to “agilization” such as scaling to large multidisciplinary projects (e. g. automotive development), decentralization of teams, cycle time of production compared to software compilation and the common integration of multiple functions in a single component (Schmidt & Paetzold, 2016). Additionally, only a certain amount of activities in a project is deemed being subject to extensive change and may thus be challenging for traditional processes – put in other terms there are complex episodes within largely non-complex projects (Sanchez-Schmitz et al., 2006). However, the proven benefits of agile development practices in software engineering remain fascinating. This paper thus strives to combine the advantages of traditional / linear and iterative / agile process models for complex projects by switching between different process models. Therefore, guidelines how to design action systems that are capable to respond to changing complexity are derived and integrated into an existing process framework. Based on this framework, the detection of increasing change during a project will be enabled. Afterwards, the selection of a suitable process model whilst maintaining smooth and consistent transitions between the different process models will be implemented in the guidelines.

2 State of Research

Providing support for fluid adaption to changing project complexity states requires a thorough definition of the term „complexity“. Afterwards, linking states of complexity to action models and investigating the state‘s impact on the individuals working in the respective projects is necessary. This section lays out the fundament of project complexity definition, depicts the recognized Cynefin Framework for state categorization, describes a common framework for agile and traditional product development processes and touches the understanding of product development processes as individual problem solving.

2.1 Project Complexity

Complexity in product development has been a major point of research since several decades: In the context of agile product development (e. g. the principle of the Walking Skeleton (Basili & Turner, 1975); Term by Alistair Cockburn) as well as in the context of traditional mechatronic product development (e. g. the discipline of Systems Engineering (Shamieh, 2011)). Since there are well established practices of handling complexity in product development projects, this section focusses on more fundamental ways to describe and understand complexity in a standardized way.

2.1.1 Dimensions of Complexity

Complexity of systems has a multitude of facets. Geraldi et al. condensed over 1.000 papers on complexity down to 25 papers on complexity of projects and extracted five dimensions of complexity (Geraldi, Maylor, & Williams, 2011). The five dimensions are defined as follows:

- *Structural Complexity* summarizes the number of system elements, their variety and interdependence.
- *Uncertainty* outlines novelty, ambiguity, experience and availability of information.
- *Dynamic* refers to the occurrence of changes.
- *Pace* aggregates urgency, criticality of time goals and the tightness of timeframes.

- *Socio-Political Complexity* arises from hidden agendas of individuals, conflicting interests, communication barriers or misunderstandings.

These five dimensions can be seen as constitutional for describing the term “complexity” in a differentiated form. Especially for the research field of product development, various authors have argued, that uncertainty is at the core of product development complexity (e. g. Meboldt, Matthiesen, & Lohmeyer, 2013) and traced this down to a lack of knowledge and a lack of definition (Albert Albers, Ebel, & Lohmeyer, 2012). At this point it may be a trivial insight, that projects differ from another regarding sources of complexity.

2.1.2 The Cynefin Framework for Categorization

Since different complexity levels require different courses of action, the Cynefin Framework (Snowden & Boone, 2007) provides a system of determining the state of a system and choosing corresponding ways to act from a leadership point of view. Therefore, four general system states are defined (see Figure 1). The fifth state “disorder” is not discussed in this paper since it is not subject of regular process models but rather a state of indefiniteness.

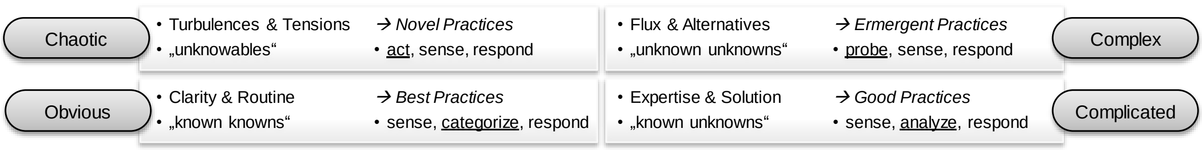

Figure 1 Four basic Cynefin Domains

The framework links the domains to suggestions for courses of action and danger signals for habits inside the domains. In the “obvious” domain a mostly straight forward process control is suggested (sense, categorize, respond). “Complicated” systems usually require extensive involvement of experts (sense, analyze, respond). “Complex” systems in contrast lack stability and are prone to significant change, which results in unpredictability of the systems behavior and outcomes. In such systems there is no point in end-to-end planning; rather getting started and letting alternative solutions emerge is the key to success. “Chaotic” systems extend this by high time pressure, which results in a focus on fast action in command-and-control chains.

It is important to understand, that by definition no complex or chaotic system can be predicted. They are like paper planes: You can tell where it was and develop theories about the why in hindsight. However, because of reciprocal interactions between various variables of the system (air pressure, temperature, humidity, swirls, material yield, …) which are all time-dependent it is impossible to prognose the exact system state in advance. Simply put, you can’t tell where the plane will land even if you highly standardize the take-off procedure. This effect can be observed in any complex product development: The solution is (at least partly) emergent and can only be planned incrementally. This issue and its implications for project management has been discussed in detail in the context of agile practices (Larman & Basili, 2003; Pich, Loch, & Meyer, 2002).Over time or by occurrence of incidents, transitions between the domains can happen and an adaption of the work procedures becomes necessary (O’Connor & Lepmets, 2015; Snowden & Boone, 2007). The transitions between the domains are tighlty connected to the availabilty or absence of knowledge (Albert Albers, Ebel, et al., 2012; Puik & Ceglarek, 2015; Snowden & Boone, 2007). Additionally it can be assumed that in real projects all domains exist in parallel, which corresponds to Albers‘ first hypothesis, that all product development projects are unique (Albert Albers, 2010).

Many authors contributed to detailing out the Cynefin framework ever since. Especially distinction between ordered and unordered systems (i. e. complicated and complex systems) were investigated from various perspectives. Unordered systems are deemed unpredictable. No matter how much knowledge exists at a point in time, no prognosis about the exact next state and risks of a complex or chaotic system can be made (Comes & Cavallo, 2013; French, 2013;

Mikkelsen, 2016). Additionally, effects and dependencies in unordered systems are considered nonlinear (Sheard & Mostashari, 2009; Van Beurden, Kia, Zask, Dietrich, & Rose, 2013).
Since its widespread application in industry and research, the Cynefin Framework can be considered a valuable and proven tool to determine a system’s state of complexity (i. e. a project’s state in the context of this paper).

2.2 Fusion of traditional and agile process models into ASD – Agile Systems Design

What happens in complex product development when starting with a „blank sheet of paper“, without resource restrictions, a lack of structured management and no proper validation of a products value proposition? This can be observed in the case of Juicero – a 130 Mio. $ investment in developing a 400 $ juice machine based on refill packs, which could be replaced by manual squeezing of the packs for 0 $ (see Figure 2**Fehler! Verweisquelle konnte nicht gefunden werden.**). The startup went bankrupt.

Figure 2 Juicero Review (Image Source Bloomberg/Neowin)

ASD – Agile Systems Design strives to prevent such shortcoming products by combining the benefits of open problem space approaches and agile methods with the structure and guidance of traditional stage-gate processes. ASD is a human-centred framework that supports product development processes with agile ways of thinking, methods and process phasing. It thus stands in the tradition of Cooper’s hybrid Stage-Gate model (Cooper, 2014) but extends it with more flexibility regarding cross-gate iterations and process details tailored for fuzzy product development aspects. The following will describe the ASD framework in more detail.
The framework of ASD – Agile Systems Design has three ancestral constructs: The iPeM – Integrated Product Engineering Model, the mindset of PGE – Product Generation Engineering and the framework for continuous validation IPEK-X-in-the-Loop (Albert Albers, Behrendt, Klingler, Reiß, & Bursac, 2017; Albert Albers, Bursac, Heimicke, Walter, & Reiß, 2017). ASD – Agile Systems Design integrates the activities of product engineering from the iPeM with the strong focus on reference systems from PGE and the demand for early validation from IPEK-XiL (Richter et al., 2018). It thus constitutes a flexible yet standardized path of progression in product maturity (see Figure 3). For a better understanding of ASD’s uniqueness, the iPeM and PGE will be presented in greater detail.

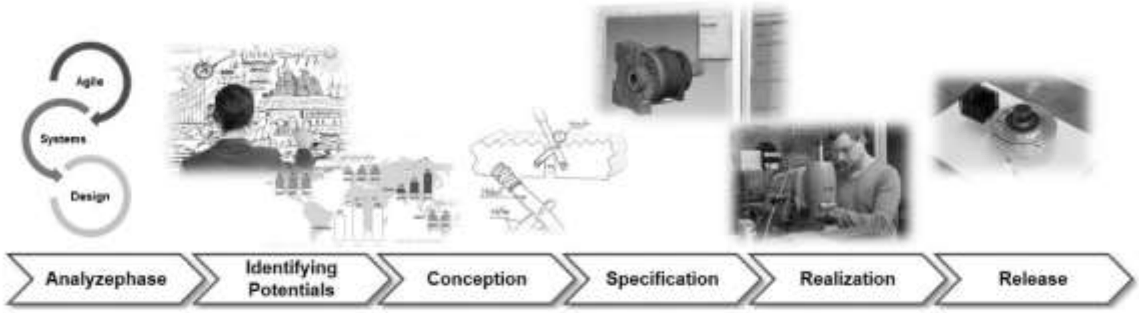

Figure 3 Process Elements of ASD – Agile Systems Design

The iPeM – Integrated Product Engineering Model is deeply rooted in the thought, that enabling, fostering and structuring communication is a strong enabler of project success. It thus strives to provide a common understanding of the mechatronic product development process for all the stakeholders (Albert Albers, Reiss, Bursac, & Richter, 2016). The iPeM focusses on naming activities and providing a basis for planning and evaluation of a projects progress. Unlike Cooper’s Stage Gate Model, it does not impose defined gates.

PGE on the other hand is rooted in an empirical observation: The vast majority of new or even disruptive products has predecessors (A. Albers, Bursac, Urbanec, Lüdcke, & Rachenkova, 2014). The concept of PGE – Product Generation Engineering derives two hypotheses, which are fundamental for ASD. On the one hand, new products and technical solutions are developed on the basis of existing reference products or reference solutions. These can be company-internal predecessor products or external competitive products, but also products from other sources. On the other hand, the development of a new product generation is carried out by a combination of the activities carryover variation (CV), embodiment variation (EV) and principle variation (PV), whereby embodiment and principle variation together represent the new development part of a new product generation (Albert Albers, Bursac, & Rapp, 2017).

On the basis of the iPeM and PGE, ASD has been developed with the goal of full scalability in mind: From multi-year development projects over medium-term sprints to short-term intensive workshops (Heimicke et al., 2018). Here, all scaling levels have common features such as the product profile, continuous validation based on development generations and the targeted sequence of analysis and synthesis activities (Albert Albers et al., 2018; Albert Albers, Behrendt, Klingler, et al., 2017). However, the concrete implementation may vary in order to improve acceptance (Lohmeyer, Albers, Radimersky, & Breitschuh, 2014). ASD structures the overarching phases by being static anchors in a dynamic work process.

2.3 Product Development from a Problem-Solving Perspective

Since product development processes are ultimately implemented in form of working individuals (Albert Albers, Breitschuh, & Lohmeyer, 2012; Breitschuh & Albers, 2014; Breitschuh, Sonnenschein, Fuchs, & Albers, 2016; Glock, 1998; Pahl, 1994) the perspective of individual problem solving will be investigated. Problems in product development are usually understood as being prone to dynamics and change and thus are defined as “complex problems” (Abele et al., 2012; Greiff & Funke, 2010). The requirement of repeatedly successful problem solving (more generally: acting) under changing circumstances can be interpreted as resulting from persons’ abilities and their actual usage. This is defined as “competence” (Weinert, 2002). Problem solving competence on the person side can be modeled as professional knowledge, knowledge to act, conditional knowledge, heuristics and self-regulation (Breitschuh et al., 2016). In the context of this paper, heuristics (Bursac, Rapp, Albers, & Breitschuh, 2017) and intuition (Campbell, 1988; Comes & Cavallo, 2013; Stewart, 2002) are of particular relevance for designing human centered process models.

In real industrial working environments, multi-tasking stemming from interruptions of concentration must be considered. A distinction can be made between self-imposed interruptions and external interruptions. While the first ones may be supporting to problem solving by increasing arousal, the latter are rather displacing attention from current thoughts and decreasing performance (Adler & Benbunan-Fich, 2012). Especially retaining the same mental problem representation (i. e. a common topic of different tasks) is crucial for successfully switching between different tasks (Salvucci, Taatgen, & Borst, 2009). However, when accuracy is the primary measure for success, task focus with only little switching should be implied. External interruptions should be avoided (Adler & Benbunan-Fich, 2012).

2.4 Interim Conclusion

No current process model is suited for all Cynefin domains. Rather the different domains suggest different process models (command-and-control in chaos, agile in complex, sophisticated traditional in complicated and straightforward processes in obvious). Over the lifespan of systems or projects, transitions between Cynefin-Dimensions occur. Influence factors are known from complexity theory, but not yet included in an overarching construct. ASD – Agile Systems Design can already model transitions from a process point of view, but by now gives only abstract guidance to individuals and teams on how to respond when transitions occur. The perspective of problem solving competence enables a differentiated view on describing peoples' activities under various circumstances of product development complexity. This is crucial for a truly human centered product development process.
Concluding, there is need for actual working concepts that adapt to the current Cynefin domain, fit into any existing process landscape (traditional or agile on any scale) and support adaption of individual ways of problem solving.

3 The EntropyCompass

In order to enable detection of domain transitions, the Cynefin framework is extended by a more differentiated view on complexity. In the following, the various aspects of complexity from section 2.1.1 will be assigned to the Cynefin framework from section 2.1.2.
First, general distinctions between ordered and unordered states are integrated. These include the system‘s behavioural predictability (probabilism, occurrence of dynamics / change), the linearity of its behaviour and the socio-political aspects of complexity. In the next steps impact factors in domain transition are integrated. These include factors for transition from obvious to chaotic (project pace, lack of definition) as well as converse factors (availability of knowledge). Finally, structural complexity is defined as a factor influencing the size of the system rather than its Cynefin domain. For instance increasing the number or known system elements raises risks of forgetting elements (i. e. generating unknown elements). Since this is a conjuction of liklihoods rather than a causal connection, structural complexity is concerned an important impact factor on risk but not on the Cynefin domain.
The integration of all these aspects leads to the term „complex“ being a single domain as well as multiple impacting dimensions on transition. Additionally, „uncertainty“, which is commonly used to describe the differences between Cynefin domains is defined as only one impact factor among others. However, Richard Tainter has introduced a different formulation: Retaining complex states of social systems requires energy (Tainter, 1996). The comprehension of acting (in complex systems) as a process of energy exchange opens the path for another measure related to this topic: Entropy. From a thermodynamic perspective, entropy can be defined as a measure of unknowingness to conclude from a systems‘ macroscopic state the microscopic state of all its elements (Becker, 1966, p. 62f, 253). This interpretation is also

applicable to the Cynefin domains and finally results in the definition of the EntropyCompass as shown in Figure 4.

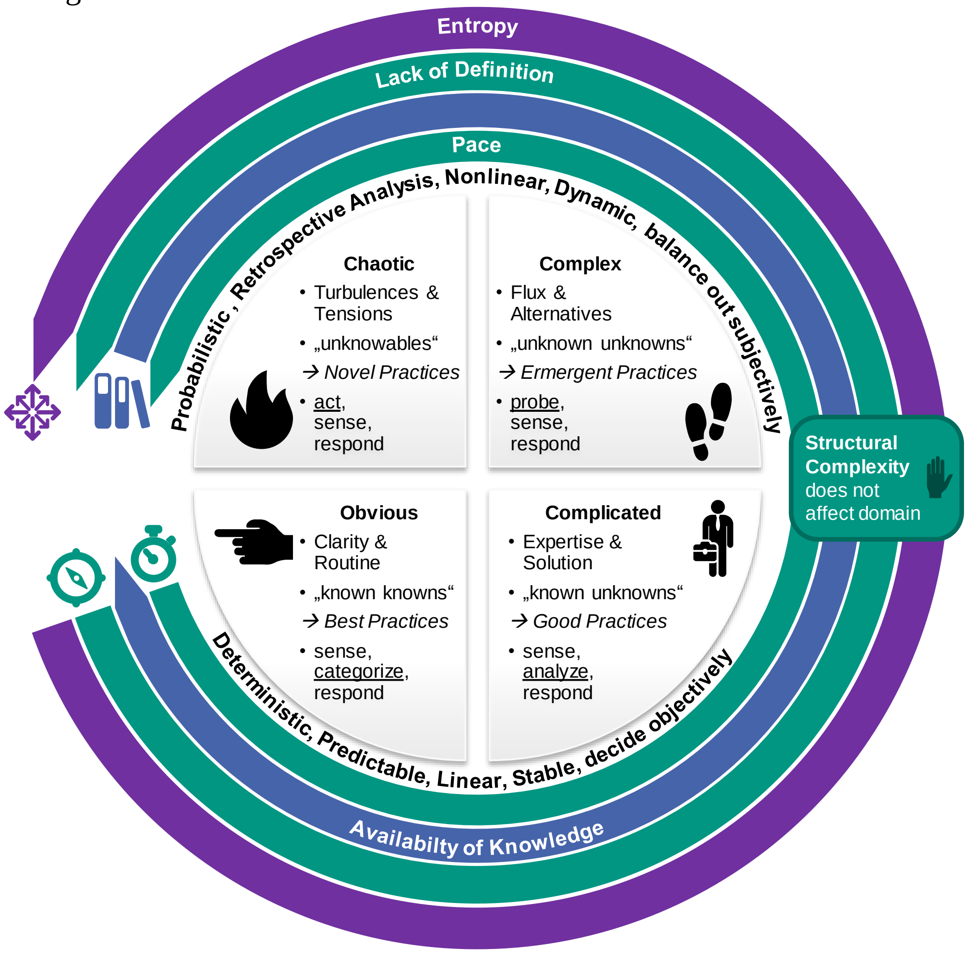

Figure 4 EntropyCompass

The EntropyCompass is designed as a vocabulary to describe and analyze the project situation with its impact factors on domain transition liklihood on the one hand and as a classification scheme for procedures to cope with the entropy level present on the other hand (see section 2.1.2). For instance, O‘Connor and Lepmets suggest using Scrum for Cynefin‘s “Probe – Sense – Respond“ course of action in complex states and rely on intense and structured communication in complicated states (O’Connor & Lepmets, 2015).

The EntropyCompass can thus be used to differentiate project tasks in terms of their potential benefit from agile practices: Obvious tasks can be handled without special effort, complicated tasks benefit from intense communication, complex tasks can be supported by agile process models and chaotic tasks must be handled with focus on risk containment. Either way, the EntropyCompass helps to identify the current state of tasks and projects. This in turn enables the detection of potential for transitions between traditional and agile ways of working.

4 The ASD-FIT Module: Focused, Intense, Teamwork

In order to provide smooth transitions between agile and traditional process models, a common basis for both process types and their interfaces must be used. The ASD-Framework has already proven this ability (Albert Albers, Bursac, Heimicke, et al., 2017). This section describes guidelines on how to apply the ASD-Framework as a fascilitator for consistent Cynefin domain transitions. This special application of the ASD-Framework is called ASD-FIT because of its main attributes: Focused, Intense, Teamwork.

4.1 Goal and setting of the ASD-FIT Module

The overall goal of ASD-FIT is to apply agile process models where helpful, retain straight forward processes where possible and integrate thorough planning where necessary.

The first goal is thus to focus on methods to support acting in complicated and complex states (compare section 3). Complicated states can be supported by timely focused and efficient integration of scarce expert knowledge. Complex states on the other hand are more deemed episodes, so that agile support should be limited in time. In contrast, obvious states require no extensive agile processing and chaotic states are beyond the scope of this paper. With the EntropyCompass, the current state and transition likelihoods can be discovered.

Since demands for process model transitions change over time, ASD-FIT must fit into any existing process landscape. This requires clear process boundaries (input, output, procedure guidelines), maximum flexibility in actual working style (i. e. ASD aspect agile) and explicitly taking into account existing work (i. e. ASD aspect PGE).

Third, ASD-FIT must support the containment of risks which correlate to the domain transition causes. Planful use of resources under uncertain boundary conditions can be achieved by timeboxing and focused workshops instead of multiproject work. In order to support risk containment, fostering communication, early validation and decision-making is crucial. Acceptance of the approach and its results is improved by requiring final results during the committed timebox.

Since a flexible way of work is to be combined with strict process guidelines, a Scrum based approach was chosen. Due to the requirement, that ASD-FIT must fit in any process landscape, the duration of an ASD-FIT workshop is limited to a maximum of five days. This necessitates scaling of Scrum (or more general agile) elements to small timescales.

4.2 Maxims and Pillars of ASD-FIT

In order to provide a common ground for Scrum and various agile and traditional practices on small timescales, guiding maxims were developed. Such maxims have proven valuable in product development teamwork (Mussgnug, Boes, & Meboldt, 2015). Since the focus of ASD-FIT is on complicated and complex domains, where human brainpower is the ace up the sleeve (expertise and improvisation), the maxims *Communication*, *Focus* and *Transparency* provide strong alignment with human centered design processes. These maxims are supplemented with pillars for realization and linked to design specifications for ASD-FIT modules (see Figure 5).

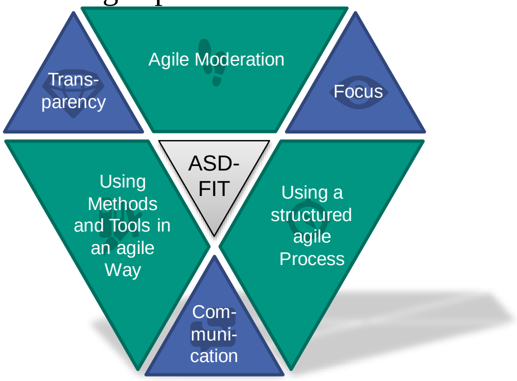

Figure 5 Overview of ASD-FIT maxims (corners) and pillars (edges)

Maxim "Focus": *Focus* means overcoming multitasking that stems from external distraction in the first place. In the second place, by requiring working modes of "one topic at a time" combined with a policy of "I finish my part and then select the next task", self-induced task-switching is managed in a transparent way. This provides a solid basis for successful application of peoples' problem-solving competences.

Maxim “Transparency”: *Transparency* is the basis for trust which in turn is basis for sustainable focus. Transparency in ASD-FIT has various dimensions. First, it is committed that after the agreed timeboxes, the task is done, and participants don’t have to put effort in it any more. Second, it is committed that everyone involved in the workshop is focussed on their defined and clarified role and do their work in cooperation with all the others. Third, the visualisation of goals, progress, tasks and timeslots manifests transparency.
Maxim “Communication”: Intensive teamwork includes the participation of decisionmakers, which helps to keep iteration goals without requiring premature freezes. This can only take place in a personal, face-to-face setting with regular and timed *Communication*. Consequently, ASD-FIT suggests using one common room and communication spaces – even facing the fact of globally distriubted development.
Pillar “Structured agile Process”: Guarding a structure whilst enabling flexibility is crucial to decrease management effort and focus on getting tasks done. Thus, the ASD phases, that constitute overarching product development maturity states are combined with a Scrum working mode on small timescales. A KanbanBoard for project management ensures transparency.
Pillar “Using Methods and Tools in an agile way”: Ensuring availability and usability of various methods of product development fosters divergent and convergent thinking tasks. On the one hand, the *InnoFox* is used as a system for situation and demand-oriented method suggestion(Reiß, Bursac, Albers, Walter, & Gladysz, 2016). On the other hand, the *KreaDesktop* is a computer interface, which enables fast and intuitive access to resources needed in most workshops such as recherche data, idea management software, digitalization of analogue artefacts and virtual collaboration.
Pillar “Agile Moderator”: The more entropy a system of project has, the more important human heuristics and intuition become (see section 2.3). The ASD-FIT modules rely on a standardized set of moderator responsibilities, tasks and form sheets. However, mental agility, the ability to gather and structure information is crucial for successful moderation. The pace and intensity of ASD-FIT modules require professional and dedicated moderation. Otherwise a severe loss in efficiency and effectiveness may occur – a guarantee for waste of scarce resources. The role of the agile moderator is thus to define the methodological path to the goals, monitor the work process and bring possible derivations from the goals and impediments to the process to a discussion in the team.

4.3 Scaling the Format for short term DayResults

One implementation of ASD-FIT are very short-term intensive workshops called *2DayResults*. In this format, work is carried out in sprints of approximately 90 minutes after the ASD procedure. This is especially suitable for project episodes demanding for intense face-to-face collaboration. The concept differs from pure Scrum by its very short iterations whilst retaining the structured communication.
In this context, particular importance is attached to achieving goals quickly. The necessary high degree of parallelization is realized in first place by carefully designed, interacting roles (see Figure 6).

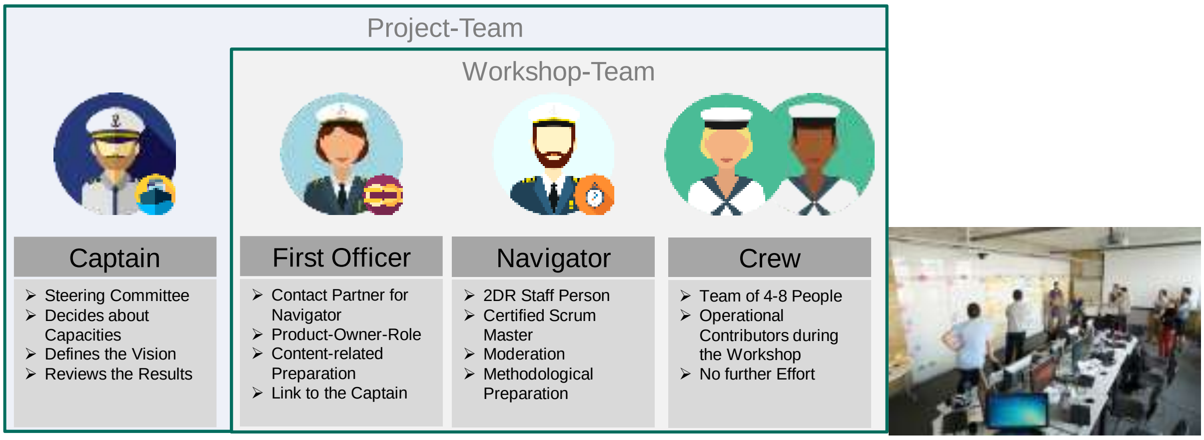

Figure 6 2DR Project Team and Role Description and actual workshop situation

By parallelization of the tasks and working on them in small teams or in individual work, the different problem-solving competencies can be integrated in a way that is appropriate to the situation. The PGE approach is also of central importance for these intensive workshops: The aim is to "maximize the share of knowledge work" by using existing documents, forms or building blocks of already achieved results.
The concept has been tested in academic contexts (e. g. developing and writing a research project proposal, improving a complete lecture) and industry contexts (e. g. business strategy and portfolio planning with 50 employees of a medium-sized enterprise).

5 Conclusion and Outlook

This paper contributes to actually creating innovation by describing a systematic way to generate results in a very concentrated effort. The presented ASD-FIT concept for workshops provides an agile setting to best possible support people who are solving complex problems. It lays explicit focus on fitting into various process landscapes. Implementation has proven the concept's feasibility and benefit. Analysis of unknowingness with the EntropyCompass ensures application of ASD-FIT to tasks that benefit most. This in turn enables efficient use of the most valuable development resource: People.
This paper presented a short example of the implementation of ASD-FIT. Future work will describe applications in greater variety and give more detailed guidelines for implementation.

References

Abele, S., Greiff, S., Gschwendtner, T., Wüstenberg, S., Nickolaus, R., Nitzschke, A., & Funke, J. (2012). Dynamische Problemlösekompetenz: Ein bedeutsamer Prädiktor von Problemlöseleistungen in technischen Anforderungskontexten? *Zeitschrift für Erziehungswissenschaft*, *15*(2), 363–391.

Adler, R. F., & Benbunan-Fich, R. (2012). Juggling on a high wire: Multitasking effects on performance. *International Journal of Human-Computer Studies*, *70*(2), 156–168.

Albers, A. (2010). Five Hypotheses about Engineering Processes and their Consequences. In *Proceedings of the TMCE 2010*.

Albers, A., Behrendt, M., Klingler, S., Reiß, N., & Bursac, N. (2017). Agile product engineering through continuous validation in PGE – Product Generation Engineering. *Design Science*, *3*.

Albers, A., Breitschuh, J., & Lohmeyer, Q. (2012). Dependency Structure Matrix Modelling for Human Resource Allocation. Presented at the 14th DSM Conference.

Albers, A., Bursac, N., Heimicke, J., Walter, B., & Reiß, N. (2017). 20 Years of Co-creation Using Case Based Learning. In *ICICL Proceedings* (pp. 636–647). Springer.

Albers, A., Bursac, N., & Rapp, S. (2017). PGE – Produktgenerationsentwicklung am Beispiel des Zweimassenschwungrads. *Forschung im Ingenieurwesen*, *81*(1), 13–31.
Albers, A., Bursac, N., Urbanec, J., Lüdcke, R., & Rachenkova, G. (2014). Knowledge Management in Product Generation Development. *25. DfX Symposium*.
Albers, A., Ebel, B., & Lohmeyer, Q. (2012). Systems of objectives in complex product development. In *Ninth TMCE*.
Albers, A., Heimicke, J., Walter, B., Basedow, G. N., Reiß, N., Heitger, N., … Bursac, N. (2018). Product Profiles: Modelling customer benefits as a foundation to bring inventions to innovations. Presented at the CIRP Design.
Albers, A., Reiss, N., Bursac, N., & Richter, T. (2016). The integrated Product engineering Model (iPeM) in context of the product generation engineering. 26th CIRP Design.
Basili, V. R., & Turner, A. J. (1975). Iterative Enhancement: A Practical Technique for Software Development. *Transactions on Software Engineering*, *SE-1*(4), 390–396.
Becker, R. (1966). *Theorie der Wärme*. Berlin; New York: Springer.
Breitschuh, J., & Albers, A. (2014). Teaching and Testing in Mechanical Engineering. In F. Musekamp & G. Spöttl (Eds.), *Competence in Higher Education and the Working Environment.* (1st ed., Vol. 12, pp. 107–129).
Breitschuh, J., Sonnenschein, E., Fuchs, J., & Albers, A. (2016). Fachliches Problemlösen in der Maschinenkonstruktion. *Journal of Technical Education (JOTED)*, *4*(2).
Bursac, N., Rapp, S., Albers, A., & Breitschuh, J. (2017). Entscheidungsheuristiken in der PGEProduktgenerationsentwicklung. Presented at the 28. DfX Symposium.
Campbell, D. J. (1988). Task complexity: A review and analysis. *Academy of Management Review*, *13*(1), 40–52.
Comes, T., & Cavallo, A. (2013). Designing decision support systems at the interface between complex and complicated domains. In: Proceedings of the Nineteenth Americas Conference on Information Systems.
Cooper, R. G. (2014). What's Next?: After Stage-Gate. *Research-Technology Management*, *57*(1), 20–31.
French, S. (2013). Cynefin, statistics and decision analysis. *Journal of the Operational Research Society*, *64*(4), 547–561.
Geraldi, J., Maylor, H., & Williams, T. (2011). Now, let's make it really complex (complicated): A systematic review of the complexities of projects. *International Journal of Operations & Production Management*, *31*(9), 966–990.
Glock, F. (1998). *Konstruieren als sozialer Prozeß.* Wiesbaden: DUV, Dt. Univ.-Verl.
Greiff, S., & Funke, J. (2010). Systematische Erforschung komplexer Problemlösefähigkeit anhand minimal komplexer Systeme. Projekt Dynamisches Problemlösen. In *Kompetenzmodellierung. Zwischenbilanz des DFG-Schwerpunktprogramms und Perspektiven des Forschungsansatzes* (Vol. Beiheft, pp. 216–227).
Heimicke, J., Reiß, N., Albers, A., Walter, B., Breitschuh, J., Knoche, S., & Bursac, N. (2018). Agile Innovative Impulses in Product Generation Engineering: Creativity by Intentional Forgetting. In *Proceedings of the Fifth International Conference on Design Creativity*
Larman, C., & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. *Computer*, *36*(6), 47–56.
Lohmeyer, Q., Albers, A., Radimersky, A., & Breitschuh, J. (2014). Individual and Organizational Acceptance of Systems Engineering Methods - Survey and Recommendations. In *Proceedings of TMCE 2014*. Budapest.
Meboldt, M., Matthiesen, S., & Lohmeyer, Q. (2013). The dilemma of managing iterations in time-to-market development processes. In *Second International Workshop on the Modelling and Management of Engineering Processes (MMEP 2012)*.

Mikkelsen, M. F. (2016). How to cope with complexity?-A review of project complexity literature using the Cynefin framework as theoretical lens. Presented at the IRIS (Information Systems Research Seminar in Scandinavia), Ljungskile, Sweden.

Mussgnug, M., Boes, S., & Meboldt, M. (2015). Principles guiding Teams in New Product Development. In *Proceedings of EPDE 2015*.

O'Connor, R. V., & Lepmets, M. (2015). Exploring the use of the cynefin framework to inform software development approach decisions (pp. 97–101).

Pahl, G. (1994). *Psychologische und pädagogische Fragen beim methodischen Konstruieren*.

Pich, M. T., Loch, C. H., & Meyer, A. D. (2002). On uncertainty, ambiguity, and complexity in project management. *Management Science*, *48*(8), 1008–1023.

Puik, E., & Ceglarek, D. (2015). The quality of a design will not exceed the knowledge of its designer; an analysis based on Axiomatic Information and the Cynefin Framework. *Procedia CIRP*, *34*, 19–24.

Reiß, N., Bursac, N., Albers, A., Walter, B., & Gladysz, B. (2016). Method Recommendation and Application in agile Product Development Processes. Design Conference

Richter, T., Heimicke, J., Reiß, N., Breitschuh, J., Albers, A., Gutzeit, M., … Bursac, N. (2018). Pitch 2.0 - Concept of early Evaluation of Product Profiles in Product Generation Engineering. In *Proceedings of TMCE 2018*.

Rigby, D., Sutherland, J., & Takeuchi, H. (2016). Embracing Agile. *Harvard Business Review*.

Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009). Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. *SIGCHI conference on human factors in computing systems*.

Sanchez-Schmitz, G., Perez-Soltero, A., Barcelo-Valenzuela, M., Martin-Rubio, F., & Palma-Mendez, J. T. (2006). Selection and Control of Knowledge Intensive Core Processes to Improve Knowledge Management Initiatives. *INTERNATIONAL JOURNAL OF KNOWLEDGE, CULTURE AND CHANGE MANAGEMENT*, *6*(1).

Schmidt, T. S., & Paetzold, K. (2016). Agilität als Alternative zu traditionellen Standards in der Entwicklung physischer Produkte. *27th DfX Symposium.*_

Shamieh, C. (2011). *Systems Engineering for Dummies, IBM Special Edition*.

Sheard, S. A., & Mostashari, A. (2009). Principles of complex systems for systems engineering. *Systems Engineering*, *12*(4), 295–311.

Snowden, D. J., & Boone, M. E. (2007). A leader's framework for decision making. *Harvard Business Review*, *85*(11), 68.

Stewart, T. A. (2002). How to think with your gut. *Business 2.0*, (November 2002).

Tainter, J. A. (1996). Complexity, problem solving, and sustainable societies. In R. Costanza, O. Segura Bonilla, J. Martínez Alier, & International Society for Ecological Economics (Eds.), *Getting down to earth: practical applications of ecological economics* (pp. 61–76). Washington, D.C: Island Press.

Van Beurden, E. K., Kia, A. M., Zask, A., Dietrich, U., & Rose, L. (2013). Making sense in a complex landscape. *Health Promotion International*, *28*(1), 73–83.

Versionone. (2017). *11th annual State of Agile Report*.

Weckmüller, H. (2017). Agilitätsbarometer 2017 - So agil sind Unternehmen in DACH. *Personalmagazin, Sonderveröffentlichung*.

Weinert, F. E. (2002). *Leistungsmessungen in Schulen*. Weinheim [u.a.]: Beltz-Verl.