INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED21
16-20 AUGUST 2021, GOTHENBURG, SWEDEN

WHEN WORLDS COLLIDE – A COMPARATIVE ANALYSIS OF ISSUES IMPEDING ADOPTION OF AGILE FOR HARDWARE

Peterson, Matthew (1);
Summers, Joshua (2)

1: Clemson University;
2: The University of Texas at Dallas

ABSTRACT

The objective of this paper is to explore challenges identified for implementation of scrum for hardware development intersect with agile principles found in the literature. A series of comparative analyses are done at the textual level, through logical intersections, and through thematic analysis. It is shown that there are five underlying themes found across two sets of scrum challenges (constraints of physicality and the 13 principles). These five themes include: flexiblity, chunkability, scalability, endurability, and teamability. These five themes further are found related to the defining principles of the agile manifesto. Using this understanding, future efforts will include empirical case study work to determine the impact that these have on application of scrum methods and tools. Additionally, guidelines should be developed to help hardware product engineers in applying scrum.

Keywords: Design methods, Case study, Early design phases, Collaborative design

Contact:
Summers, Joshua
Clemson University
Mechanical Engineering
United States of America
joshua.summers@utdallas.edu

Cite this article: Peterson, M., Summers, J. (2021) 'When Worlds Collide – A Comparative Analysis of Issues Impeding Adoption of Agile for Hardware', in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.606

1 MERGING OF TWO WORLDS - AGILE FOR HARDWARE

Innovation is strategically important to companies, affecting both their ability to introduce new products and determining the competitive position of those products within the marketplace. Additionally, product development is a risky proposition and there is no guarantee of success for new initiatives (Motte, 2015). As such, organizations are constantly seeking methods to ensure success and enhance creativity within the product development process (Marxt and Hacklin, 2005; Pahl et al., 2013). Within software development, the agile method is cited for improvements to the development process and enhancing creativity (Dingsøyr et al., 2012; Highsmith and Cockburn, 2001; Palmqvist and Trifunov, 2019). Innovative companies, such as SpaceX (Rigby et al., 2018), appear to be using agile methods in the development of hardware systems to iteratively deploy creative new products and gain a competitive position with the market. Agile tools and methods are recently being explored for use in hardware development (Böhmer et al., 2017; Garzaniti et al., 2019; Wagner, 2014), extending the software development approaches to a new domain. To this end, researchers have identified constraints of physicality (Schmidt et al., 2017) and thirteen challenges to agile for hardware (Ullman, 2019a, 2019b). This paper uses several methods to compare challenges from each paper, determining if they are the same or different, and allocates them across five major themes. The results of this comparison provide a consolidated view of the challenges in implementing Agile for hardware and can inform future research efforts.

2 WHAT ARE AGILE AND SCRUM

This section provides a brief overview of Agile and one of the practices, Scrum. Agile is an iterative development methodology with origins in the American software development industry. It is more of a philosophical approach to development versus a rigid methodology, such as a traditional systems engineering (Haskins and Forsberg, 2011; Kapurch, 2007) or product development approach (Jänsch and Birkhofer, 2006). Published in 2001, the Agile Manifesto, provides four core values and twelve guiding principles, emphasizing people and interactions, working software, collaboration with the customer, and ability to respond to change (Fowler and Highsmith, 2001). Key aspects of the method include decentralized decision-making, recurrent experimentation and prototyping, rapid feedback, routine delivery of usable product, and increased collaboration both within teams and with the customer. The primary benefit of agile is enabling companies to respond faster to change within a dynamic product development environment. Several practices exist for the implementation agile, including Scrum, Kanban, eXtreme Programming (XP), and the Scaled Agile Framework for Lean Enterprises (SAFe) (Heimicke et al., 2019).

Scrum is one of the practices implementing agile (Cristal et al., 2008; Permana, 2015; Schwaber and Sutherland, 2011). It defines three critical team roles and responsibilities (product owner, development team, and scrum master), four essential meetings and cadence (sprint planning, daily scrum, sprint review, and retrospective), and the generation of three required artefacts (product backlog, sprint backlog, and the increment). The product owner establishes the vision for product development, and captures required features in the product backlog. The product owner works with the scrum master and development team to allocate features from the product backlog to a sprint backlog, for implementation in the coming sprint. A sprint is a defined period, typically two to four weeks, during with the development team promises to produce an increment incorporating features in the sprint backlog. During sprint planning, the development team works with the scrum master to estimate effort to achieve the features in the sprint backlog, ensuring they are achievable within the sprint timeline. The daily scrum is a short meeting, typically less than fifteen minutes, during which time the development team brief what work they have completed, what they are working on next, and any roadblocks impeding progress. At the end of the sprint, the development team produces the increment and highlights their achievements in the sprint review. The sprint retrospective captures lessons learned for the next sprint.

The tools and methods used in Agile and Scrum were created to address software development. While organizations have experimented with applying Agile for hardware, issues exist in applying it to this new domain.

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

3 TWO VIEWS ON THE COLLISION

Research groups are investigating Agile for Hardware (AfH). One effort has defined Constraints of Physicality (CoP) as a collection of limitations associated with implementing agile principles in the development of hardware (Schmidt et al., 2017). A second effort has identified thirteen guiding principles that should be considered when deploying scrum methods for hardware product development (Ullman, 2019b). This paper presents a comparison of these two views to determine overlaps and gaps.

3.1 The View from Constraints of Physicality (CoP)

Researchers have identified challenges from the Constraints of Physicality (CoP) through a directed network, highlighting cause-and-effect chains, referred to as backbones (Schmidt et al., 2017). CoP are a collection of difficulties in applying agile principles to hardware development and building a potentially shippable prototype in a couple of weeks. These challenges are interdependent, making it difficult for the researcher to know which to tackle first. Hereafter, references to this paper use the abbreviation CoP.

Analysis starts with a literature review identifying challenges and interdependencies relevant to CoP. Formation of a directed network has nodes representing challenges and edges representing their interdependencies. Edge direction allows identification of cause-and-effect. To establish a weighting of the edges six interviewees rate the significance of the cause-and-effect relationship, on a 10-point Likert scale. Applying the weighting results into the directed network exposes four backbones, separation, flexibility, scaling, and task breakdown. These backbones include 34 identified challenges from six referenced sources. Subsequent identification of these challenges in this paper follows this format, CoP-XX, where XX represents the challenge id from CoP. For instance, CoP-02 refers to "hard to separate deliverables for each iteration".

Researchers summarized findings from seven Danish organizations as they attempt to use Scrum for new product development over a three-year period (Ovesen, 2012). Themes from this paper become three of the four backbones in CoP. Other work compares AfH case studies from industry and academia, identifying challenges and common themes between these two different user groups (Ovesen and Dowlen, 2012). CoP challenges cite this reference twice, neither exclusively. It was recommended that a set of tools and approaches to embrace change and flexibility as an alternative to traditional waterfall product development be created (Smith, 2008).

Two of the CoP references arise from challenges with agile in software development but are deemed applicable to AfH. The top ten issues affecting the agile development community as identified at the XP 2010 conference are summarized in (Freudenberg and Sharp, 2010). CoP challenges cite this reference four times primarily in the scalability backbone, twice exclusively. In a more expansive study of agile challenges, researchers examined and synthesized 193 agile challenges collected at conferences during 2013 and 2014, finding seven themes involving organization, sustainability, culture, teams, scale, value and claims and limitations (Gregory et al., 2015). Results are compared against previous findings including (Freudenberg and Sharp, 2010). CoP challenges cite this reference six times primarily in the scalability backbone, four exclusively.

3.2 The View from Thirteen Challenges (13X)

Thirteen challenges (13X) is a supplement to a traditional design textbook that provides a detailed approach and recommendations on how to apply Scrum to hardware projects and integrate within traditional development paradigms (Ullman, 2019b). Two case studies are provided. The first is from Saab, highlighting development of the Saab Gripen E fighter using Scrum, specifically focusing on the oxygen delivery system. Additional information on Saab's approach found in (Furuhjelm et al., 2017; Steinkellner et al., 2009). The second case study is a student design team from Olin College using Scrum to design a prosthetic arm, iterating through several prototypes over an eight-week period. Appendix B of the supplement highlights thirteen challenges when applying Scrum to hardware projects. Additional context, figures, and examples are provided in (Ullman, 2019a). Subsequent identification of these challenges in this paper follows this format, 13X-XX, where XX represents the challenge as numbered in Appendix B of 13X. For instance, 13X-09 refers to "software testing is very different from hardware testing".

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

4 COMPARING VIEWS - LATENT SEMANTIC ANALYSIS

A comparison of challenges from these papers provides insight if the challenges are the same or different, resulting in a consolidated view of the research, which is useful in selecting new research efforts. Three methods are used to perform this comparison, with varying results. The first, Latent Semantic Analysis (LSA) examines text to determine the similarity in meaning between passages (Deerwester et al., 1990). An online application1 allows comparison of text blocks providing a normalized similarity value in return. LSA helps the researcher by automating comparison of text; both large numbers of text blocks and large sizes of the individual blocks. To use LSA, each issue from from CoP and 13X must be consolidated into singular blocks of text with no carriage return.

Three attempts were made to use the LSA method. Text segments from all three attempts are compared to the same body of text from (Ullman, 2019a, 2019b) which is separated into thirteen segments. The first attempt compared each of the 32 challenge nodes from CoP to the thirteen challenges from 13X. Results from this comparison show a strong correlation between the number of characters in the 32 challenge nodes and the similarity rank. In the context of this comparison this indicates larger text samples produce higher results regardless of similarity of meaning between the two passages.

A second attempt increases the size of the text from CoP by combining all of the challenge nodes under a backbone into a consolidated block of text, four in total. However, this comparison also shows a bias towards larger text samples despite the increase in size of the block text. The third attempt uses text from the CoP referenced sources, noted in section 3.1, to increase the amount of text available for comparisons. Unfortunately, the results of the third LSA comparison show little differentiation in scoring between the CoP backbones despite the increase in text length. These disappointing results from the LSA method most probably arise from limitations in the vocabulary available to the analysis engine. Comparison results indicate common engineering terminology such as architecture, modular, scrum, shippable, iterative, and prototyping have no definition within the LSA algorithm that would allow accurate comparison between the texts.

Table 1. Summary of results from three attempts at LSA

Attempt:	First	Second	Third
13X input	13 blocks of text	13 blocks of text	13 blocks of text
CoP input	32 challenge nodes from table, synopsized from referenced papers	4 backbones made up of consolidated text from challenge nodes	4 backbones made up of text from referenced papers
Results	Strong correlation between text length of CoP input and similarity value	Strong correlation between text length of CoP input and similarity value	No differentiation between results

The LSA effort provides a foundation for subsequent analysis. The third LSA analysis establishes a more detailed understanding in the underlying issues for each challenge node and backbone from the CoP. This increased depth in understanding enabled an effective manual comparison between the issues highlighted in both papers. It is interesting to note that most of the references provided in the CoP challenge nodes come from (Ovesen, 2012). Three of the four backbones (task-breakdown, deliverable-separation, and flexibility) are key analytical themes of this paper. Alternatively, challenge nodes from the fourth CoP backbone, scalability, arise predominantly from (Freudenberg and Sharp, 2010; Gregory et al., 2015).

5 COMPARING VIEWS - VENN DIAGRAM

The objective of this analysis is to create a Venn diagram to provide an easy to understand, visual comparison between the issues raised in both papers. Venn diagrams are routinely used to visualize the relationship between two sets of data (e.g. set A and set B). Data falls into three areas, the intersection between A and B in the centre and A-not-B or B-not-A on either side. For this analysis highly similar

[1] http://lsa.colorado.edu/ last accessed 11/29/2020.

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

issues between the papers are allocated to the intersection area and issues only mentioned in one paper and not the other are allocated to A-not-B area or vice versa. Organizing issues into common themes is helpful to determine if the papers share any mega-trends. To facilitate comparison the issue hierarchy is borrowed from the CoP paper, with the nodal challenges (issues) supporting the challenge backbones (categories). This analysis borrows the CoP backbones as an initial set of categories and seeks to allocate the combined 45 challenges from both papers and indicate how they are shared between the two papers. Therefore, this analysis performs an initial allocation of the 13X challenges to a category, which is initially based on the CoP backbones. After initial allocation of the 13X challenges, they are compared to the CoP challenge nodes to determine which of the three Venn diagram zones each issue will fall. Finally, consideration is given towards adding categories for issues that do not logically within the initial category set.

Figure 1 depicts the results of the analysis described above. The light grey oval represents the set of information (categories and issues) from the CoP paper and the dark grey oval those from the 13X. Similarly, the smaller, light grey, italicized text are issues (nodes) from CoP and the black text challenges from 13X. Bold, black, underlined text identify the seven categories that contain all 45 issues from the two papers.

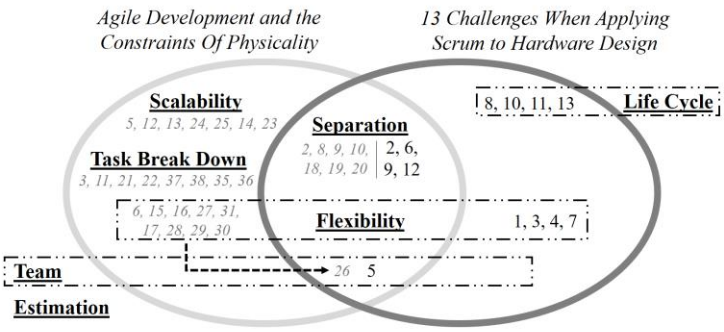

Figure 1. Venn diagram comparison of identified issues and categories

5.1 Flexibility Category

Flexibility is a key enabler for agile (Smith, 2008). One of the key tenets in The Agile manifesto is "Responding to change over following a plan". One of the twelve principles expands on this further states teams should "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage." With Agile, accommodating change is a benefit and advantage for organizations. This is antithetical to traditional measures of progress and decision making in hardware (Ovesen, 2012). Both documents agree that addressing the lack of flexibility will help in the implementation of AfH. However, they do so on primarily independent terms. CoP notes limits on flexibility from outside dependencies like certification and supplier lead times, challenges in keeping design options open, and the general difficulty in changing solutions in hardware development (Ovesen, 2012). Issues from 13X addressing flexibility include challenges with modularity (13X-01), adding features to complete designs (13X-03), simplification of design (13X-04), and cost (13X-07). As shown in Figure 1, the issues in 13X expand the set of challenges identified in CoP and present new research opportunities.

5.2 Separation of Deliverables Category

Agile emphasizes several key principles including early and continuous delivery, on a timescale of a couple of weeks to months, while ideally maintaining a constant pace indefinitely (Fowler and Highsmith, 2001). Scrum seeks to accomplish this by concluding each sprint with a "potentially shippable product increment" (Cristal et al., 2008; Permana, 2015). This can be challenging to accomplish with hardware when manufacturing, supply chain procurement, assembly, and test, can exceed these timelines and may facilitate routine delivery (Ovesen, 2012). Issues from both papers address this category. 13X-02 notes developing hardware in short sprints is a challenge and requires setting realistic sprint goals aligning well with CoP-18 regarding difficulty conceptualizing product iterations. 13X-06 notes that demonstrating hardware is not done simply by hitting a button to generate a build, aligning to CoP-19 regarding challenges with manufacturing. 13X-09 notes the testing of

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

large, complex, hardware programs such as aviation systems can easily exceed sprint timelines, aligning to CoP-20 around testing. 13X-12 notes the additional challenges when including the customer in demonstrating, prototyping and testing, aligning with CoP-08 and -09 which also highlight the challenges in evaluating and selling the customer on iteration delivery. Both documents agree that addressing the separation of deliverables is a challenge in the adoption of Agile for Hardware. The issues in 13X map well to CoP, adding context to but not expanding the set of challenges, as shown in the centre of the Figure 1 Venn diagram.

5.3 Task Break Down Category

The separation of deliverables category identifies issues in the delivering and evaluating product, the synthesis of design. The task break down category examines the analysis of design, breaking tasks into smaller efforts. While related, these categories are different, representing two sides of the same coin. With the scrum method, one of the responsibilities of the Product Owner is to establish key objectives for the development team. These objectives may not be achievable within the duration of a sprint and are broken into smaller tasks spanning multiple sprints (Cristal et al., 2008; Permana, 2015). CoP-03 notes this is challenging in hardware, causes arising in three primary areas. CoP-11, -21, -22, -37, and -38 note addressing this challenge is primarily an issue of changing mind-sets and breaking established practices rather than a technical change. CoP-35 notes that prioritizing these tasks is challenging. CoP-36 addresses issues with developing a sprint backlog. Challenges from 13X do not specifically address the task break down category or issues associated with changing mind-sets, establishing, and prioritizing the sprint backlog. Therefore, these challenges are shown only on the CoP side of the Venn diagram in Figure 1.

5.4 Scalability Category

Application of the agile method is effective for product development in student design projects (Ovesen and Dowlen, 2012; Ullman, 2019b). While easier to conceive on smaller projects, agile must be able to scale up to projects of larger scope, with multiple design teams, to be an effect development method for hardware. CoP identifies scalability as a one of the four main impediments (backbones) in applying agile for hardware (AfH).

Reasons cited for this include increased complexity when scaling (CoP-23), lack of a known process addressing scaling (CoP-14), understanding what agility means at the program level (CoP-13, -24, -25), and an absence of governance mechanisms if you could scale (CoP-12). Interestingly, (Freudenberg and Sharp, 2010; Gregory et al., 2015) are the source for these findings and primarily address challenges with the agile method in general and not specific to hardware, Scaling is not identified as a significant issue in (Ovesen, 2012) which is the primary source for most of the challenges identified in CoP.

Conversely, 13X does not address scalability as an issue, using the development of the Saab Gripen E fighter as one of two case studies (Furuhjelm et al., 2017; Steinkellner et al., 2009). This is cited as a large scale and complex design project using scrum employing over 1000 engineers, operating within 100 scrum teams, on a three-week sprint cycle.

How can we reconcile the apparent disconnect between the papers? It is possible the scope is not generalizable to all of hardware development or that details on this program were not widely available when writing CoP. While 13X mentions the scale of the program, it provides limited details on integration of efforts. Much of the discussion surrounds a specific case study focusing on a ten-person team tasked with the development of the oxygen delivery system and integration of this system with ejection seat.

Other papers addressing agile development on this program focus on applications of a more limited scope involving modelling and simulation rather than the broader issues design (Furuhjelm et al., 2017; Steinkellner et al., 2009). Focusing on modelling and simulation means iterations may be virtual, not physical, eliminating many of the CoP challenges associated with creating and changing things in the real world. Still a significant achievement, but maybe not transferable to design domains lacking a strong correlation between simulation and real world results.

Only CoP addresses scalability issues, as shown on the Venn diagram in Figure 1. Further investigation is warranted to understand the applicability of the scalability challenge to reconcile the disconnect between these opposing views.

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

5.5 Full Product Lifecycle Category

Four challenges from 13X are interrelated and do not fit neatly into the categories borrowed from CoP. Consideration is given to an additional category aimed at developing agile methods across the product development lifecycle.

- As noted in 13X-10, realization of hardware development requires more than just design and must address the entire product development lifecycle, specifically noting manufacturing, supply chain, assembly, test, and quality.
- 13X-8 adds operations to the lifecycle topic, noting hardware faces unique challenges compared to software when addressing operational environments. For example, while a go cart and a snow mobile have similar functions they possess significantly different physical embodiments resulting from their intended operational environment. Today software code compiles easily to run in different environments, but this was not always the case and required development to address.
- Further extending the list of product lifecycle topics, 13X-13 adds problem definition. Noting that design without a clear understanding of the problem results in wasted effort, time, and money. The build-measure-learn action sequence of iteration is only effective if design teams understand how their efforts satisfy the high-level requirements characterizing the problem.
- Finally, 13X-11 recommends development of new approaches to manage requirements that leverage the flexibility of higher-level user stories from software with the measurable engineering requirements from methods like Quality Function Deployment (QFD). Providing for freedom of creativity within design teams while limiting scope creep requires new methods for hardware.

While not addressed specifically in CoP there is support from additional researchers. First, researchers noted the need to establish agility throughout the entire value chain and agile development efforts suffer from a mismatch between IT teams and the wider, more traditional, organizational structure (Gregory et al., 2015). Additionally, similar findings regarding requirements, even recommending specifying requirements at a higher level and adapting product visions, personas, use cases, and user stories are found in (Ovesen and Dowlen, 2012; Smith, 2008). However, only 13X directly addresses these challenges. This can be seen on the Venn diagram in Figure 1.

5.6 Team Composition Category

Team composition challenges are unique and merit their own category. Components of the Agile Manifesto and key principles specifically address the topic of teams. The skill sets for team members in agile software teams are complimentary or overlap to a large degree. This allows team members to help in other areas to tackle issues and complete a sprint. Both documents note this approach is a significant challenge for hardware, 13X-05 and CoP-26. The large number and specialization within disciplines and specialization make this a challenge. CoP mentions this issue is important but falls outside of the issues related to the physical nature of hardware. Furthermore, Ovesen (2012), includes team composition as a main theme in the analysis. Figure 1 shows team composition as a category, including 13X-05. The arrow depicts the reallocation of CoP-26 from the flexibility category to team composition, which is a better fit.

The reference documents identify other issues impacting team composition, primarily team distribution. Mention of this issue is made in both Gregory et al. (2015) and Freudenberg and Sharp (2010). In the later paper, four of the top ten areas for further research include team distribution, emphasizing the importance of this issue. Distribution of teams affects many programs and geographic colocation, as is the desired within the principles behind the agile manifesto, is difficult to achieve. This is the case with many large programs. For instance, NASA has different centres devoted to design, test, launch, and operations. Furthermore, with this is a timely topic during the COVID-19 pandemic.

As shown in Figure 1, both papers address issues related to team composition in similar terms.

6 COMPARING VIEWS - N^2 MATRIX

Venn diagram analysis is useful in understanding which issues are similar, how similar they are, and form logical groupings. However, it is not illustrative in how each issue relates to all other issues. An N^2 diagram allows comparison of each issue to every other issue, allowing identification of broader trends versus the Venn diagram. The population of issues is made up of all thirteen issues from 13X and the four backbone issues from CoP, creating a 17x17 comparison matrix. This comparison identifies if an issue from 13X is already in CoP or it is a new issue and to which backbone it aligns

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

best. Starting with results from the Venn diagram analysis, comparing issues results in a numeric value Four comparison values are possible: 9 indicates the text blocks are referencing the same issue; 3 indicates the issues are related but not the same; 1 indicates a loose alignment between the issues; and 0 indicates no similarity between the issues. Using the Venn diagram, issues within the same category are evaluated with respect to each other first, and then compared to all other issues. This facilitates an orderly and fast comparison. The N matrix is initially ordered based on the listing of issues from CoP and 13X but can be rearranged to group similar items, as indicated in the evaluation scores, to show broader trends across the data. Values for the comparison of an issue to itself are left blank. Assigning this comparison a value skews the results by de-emphasizing similarity with other data (i.e. we know the issue agrees with itself, we want to know how it compares with other data).

6.1 N^2 Matrix Results

Figure 2 shows the results for the N^2 analysis. The comparison between issues is non-directional based on relationships between factors, thus resulting in a symmetric matrix. The colours vary in a gradient, with darker grey indicating a strong relationship and the lightest grey indicating a weak relationship. Comparison cells left blank indicate that no relationship is identified. Cells indicating a comparison of an issue to itself are coloured black. A geometric rating system (1, 3, 9) is used to provide quantitative values. Summation values for the rows and a count of non-zero values appear in columns on the right of the figure. The non-zero count is a measure of the similarity of the issue to other issues. Those with a higher count are more similar to other issues than those with lower count values. The sum value indicates the strength of those similarities.

Issues	13X-1	13X-3	13X-4	*CoP-06*	13X-7	*CoP-02*	13X-12	13X-2	13X-9	13X-6	*CoP-03*	13X-13	13X-11	13X-10	13X-8	*CoP-05*	13X-5	Sum	Count
13X-1		3	3	3	3	1	1	1	1	1	1		1			1		20	12
13X-3	3		3	3	3	1		1			1							15	7
13X-4	3	3	**Zone 1**	3	3	1		1	**Zone 4**		1			**Zone 5**			1	16	8
CoP-06	3	3	3		3	1					1		1				3	18	8
13X-7	3	3	3	3			1		1	1				1				16	8
CoP-02	1	1	1	1			9	9	9	9	3	1					1	45	11
13X-12	1				1	9		3	3	3	1		1	**Zone 6**				22	8
13X-2	1	1	1			9	3	**Zone 2**	1	1	3	1	1	1				23	11
13X-9	1				1	9	3	1		1	3	1	1		1			22	10
13X-6	1				1	9	3	1	1		1			1				18	8
CoP-03	1	1	1	1		3	1	3	3	1		1					1	17	11
13X-13						1		1	1		1		3	3	3			13	7
13X-11	1			1			1	1	1			3	**Zone 3**	3	3			14	8
13X-10					1			1		1		3	3		3		1	13	7
13X-8									1			3	3	3				10	4
CoP-05	1															**Zone 7**	1	2	2
13X-5			1	3		1					1			1		1	**Zone 8**	8	6

Figure 2. N^2 comparison results

Results at the issue level are as follows:

- Most connected issues: 13X-1, modularity, with 12 connections, 13X-2, with 11, and 13X-9 with 10. These also have a high value for sum. Indicating these issues are well connected and influential. Addressing these issues has the greatest potential for the researcher to make a significant improvement in the adoption of AfH.
- Least connected issues: 13X-8 with 4, and 13X-5 with 6. They also have the lowest sum value across these connections, indicating these issues are not well connected, not influential, and are at the low end of priority for further research.
- Most connected backbones: CoP-02 deliverable separation and CoP-03 task breakdown. CoP-02 also has the highest sum total from those connections, 45, nearly twice the value for the runner up.
- Least connected backbone: CoP-05 scalability

Columns and rows have been rearranged to group together similar issues and expose broader trends in the analysis. Black boarders and identifiers indicate zones of interest for further analysis. Organizing the matrix in this manner shows three zones with strong internal similarity, zones 1, 2, and 3. Also present are impact zones, 4, 5, and 6. These zones show the effects of zones 1, 2, and 3 on each other.

7 AFTER WORLDS COLLIDE - KEY FINDINGS AND FUTURE RESEARCH

The preceding analysis compared issues from CoP and 13X using three methods with increasing insight. Overall, these comparisons produces a consolidated set of issues and allows examination of broader themes. The N^2 matrix analysis provides both a quantitative comparison from issue to issue, identification of broader themes, and evaluation of similarity or effects between issues. Grouping of

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

similar issues in Figure 2 identifies zones 1, 2, and 3 as broader themes. Based on the Venn diagram and N^2 analyses, the scalability backbone, zone 7, and issues in team composition, zone 8, have little similarity to other issues and represent themes as well. Therefore, this analysis identifies five themes as shown in Figure 3.

Flexibility includes all of the issues identified from the CoP backbone and adds 13X-1, 3, 4, and 7. "Chunkability" combines the highly similar issues from the task breakdown and separation backbones. As noted above, these issues represent the analysis and synthesis of attacking development in sprints. As seen in Figure 2, the set of issues here is not expanded as 13X-2, 6, 9, and 12 are covered in CoP. The "Endurability" theme encompasses the full product lifecycle category identified above and includes 13X-8, 10, 11, and 13, which are new issues. Scalability includes issues from CoP only. Resolution of the apparent disconnect in the significance of this issue warrants further investigation. "Teamability" includes issues identified in both papers, CoP-26 and 13X-05. It also includes the issue of team distribution, identified in CoP references. Count and sum values for each zone identified in Figure 2 allow calculation of a cell average. Cell averages above three indicate some of the issues in this zone are identical. The weight of the connection lines and corresponding number between themes represent the similarity between these themes. For instance, Zone 2, chunkability, and zone 1, flexibility, have the highest rated similarity of 17. This indicates investigations into either theme have a higher potential to affect each other.

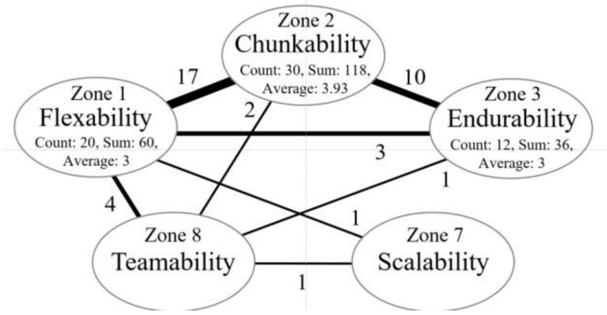

Figure 3. Key themes summarizing the challenges in applying agile for hardware

It is important to relate research back to the needs of the engineering designer. The agile manifesto outlines twelve facilitators for implementing agile. Table 2 maps these principals to the themes identified in Table 2 using the grading scheme from the N^2 matrix. A strong alignment is seen with a connection between each agile principal and at least one theme, and vice versa. This analysis validates the themes identified are important challenges in application of the agile method to hardware development. Future work includes developing guidelines to address themes and consolidated set of issues for use by the engineering designer in the application of agile.

Table 2. Mapping of themes to agile principals

ID	Principles	F	C	E	T	S
9	Continuous attention to technical excellence and good design enhances agility.	9	1	1	1	1
2	Welcome changing requirements, even late in development.	9		3		1
1	Highest priority is to satisfy customer through early and continuous delivery.		9	3		
3	Deliver working software frequently, with a preference to the shorter timescale.		9	3		
10	Simplicity--the art of maximizing the amount of work not done--is essential.		9			1
7	Working software is the primary measure of progress.	1	3	1	3	1
8	Promote sustainable development, maintain a constant pace indefinitely.		3	9	3	1
12	At regular intervals, the team reflects on how to become more effective.		3		9	
4	Business people and developers work together daily throughout the project.			1	9	
5	Build projects around motivated individuals.				9	
6	Face-to-face conversation, is the most effective way to conveying info.				9	
11	Self-organizing teams create the best architectures, requirements, and designs				9	

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press

REFERENCES

Böhmer AI, Hugger P and Lindemann U (2017) Scrum within hardware development insights of the application of scrum for the development of a passive exoskeleton. In: 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), 2017, pp. 790–798. IEEE.

Cristal M, Wildt D and Prikladnicki R (2008) Usage of Scrum practices within a global company. In: 2008 IEEE International Conference on Global Software Engineering, 2008, pp. 222–226. IEEE.

Deerwester S, Dumais ST, Furnas GW, et al. (1990) Indexing by latent semantic analysis. Journal of the American society for information science 41(6): 391–407.

Dingsøyr T, Nerur S, Balijepally V, et al. (2012) A decade of agile methodologies: Towards explaining agile software development. Elsevier.

Fowler M and Highsmith J (2001) The agile manifesto. Software Development 9(8). [San Francisco, CA: Miller Freeman, Inc., 1993-: 28–35.

Freudenberg S and Sharp H (2010) The top 10 burning research questions from practitioners. Ieee Software 27(5). IEEE: 8–9.

Furuhjelm J, Segertoft J, Justice J, et al. (2017) Owning the Sky with Agile: Building a Jet Fighter Faster, Cheaper, Better with Scrum. Global Scrum Gathering, San Diego, CA.

Garzaniti N, Briatore S, Fortin C, et al. (2019) Effectiveness of the scrum methodology for agile development of space hardware. In: 2019 IEEE Aerospace Conference, 2019, pp. 1–8. IEEE.

Gregory P, Barroca L, Taylor K, et al. (2015) Agile challenges in practice: a thematic analysis. In: International Conference on Agile Software Development, 2015, pp. 64–80. Springer.

Haskins C and Forsberg K (2011) Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Incose. DOI: INCOSE-TP-2003-002-03.2. 1.

Heimicke J, Niever M, Zimmermann V, et al. (2019) Comparison of Existing Agile Approaches in the Context of Mechatronic System Development: Potentials and Limits in Implementation. In: Proceedings of the Design Society: International Conference on Engineering Design, 2019, pp. 2199–2208. Cambridge University Press.

Highsmith J and Cockburn A (2001) Agile software development: The business of innovation. Computer 34(9). IEEE: 120–127.

Jänsch J and Birkhofer H (2006) The development of the guideline VDI 2221-the change of direction. In: DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia, 2006.

Kapurch SJ (2007) NASA Systems Engineering Handbook. NASA Special Publication.

Marxt C and Hacklin F (2005) Design, product development, innovation: all the same in the end? A short discussion on terminology. Journal of Engineering Design 16(4). Taylor & Francis: 413–421.

Motte D (2015) Effectiveness of the systematic engineering design methodology. In: International Conference on Engineering Design (eds C Weber, S Husung, M Cantamessa, et al.), Milan, Italy, 2015, pp. 77–86. The Design Society.

Ovesen N (2012) The challenges of becoming Agile: Implementing and conducting scrum in integrated product development. Department of Architecture and Design, Aalborg University.

Ovesen N and Dowlen C (2012) The challenges of becoming agile–experiences from new product development in industry and design education. In: International Conference on Engineering and Product Design Education, 2012. London South Bank University.

Pahl G, Beitz W, Blessing L, et al. (2013) Engineering Design: A Systematic Approach (Seconded.). 3rd ed. London: Springer-Verlag London Limited.

Palmqvist O and Trifunov M (2019) Agile hardware development.

Permana PAG (2015) Scrum method implementation in a software development project management. International Journal of Advanced Computer Science and Applications 6(9): 198–204.

Rigby DK, Sutherland J and Noble A (2018) Agile at scale. Harvard Business Review 96(3): 88–96.

Schmidt TS, Chahin A, Kößler J, et al. (2017) Agile development and the constraints of physicality: a network theory-based cause-and-effect analysis. In: Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods, Vancouver, Canada, 2017, pp. 199–208. The Design Society.

Schwaber K and Sutherland J (2011) The scrum guide. Scrum Alliance 21: 19.

Smith PG (2008) Change: Embrace it, don't deny it. Research-Technology Management 51(4). Taylor & Francis: 34–40.

Steinkellner S, Andersson H, Gavel H, et al. (2009) Modeling and Simulation of Saab Gripen's Vehicle Systems. In: AIAA Modeling and Simulation Technologies Conference, 2009, p. 6134.

Ullman DG (2019a) 13 Challenges When Applying Scrum to Hardware Design. Machine Design, December.

Ullman DG (2019b) Scrum for Hardware Design: Supporting Material for the Mechanical Design Process. Corvallis, OR: David Ullman LLC.

Wagner S (2014) Scrum for cyber-physical systems: a process proposal. In: Proceedings Of The 1St International Workshop On Rapid Continuous Software Engineering, 2014, pp. 51–56.

https://doi.org/10.1017/pds.2021.606 Published online by Cambridge University Press