

Final Section (4.3.5, 4.3.7, and 5. Discussion)

This part of the paper focuses on the challenges, limitations, and complexities of adopting Agile Project Management (APM) and the Scrum framework in a distributed academic research setting (the CECAN initiative). It moves from specific interview findings to a broader discussion of their implications.

Key Findings from the Interview Data (Sections 4.3.5 - 4.3.7):

1. **Debate on Research Suitability (4.3.5):** There is no consensus among researchers on Agile's universal applicability. Some contend it is best suited for applied research with stakeholder pressure and time constraints, viewing it as unhelpful for theoretical, basic research. Conversely, other researchers provided examples of successfully integrating Scrum practices into their own small-scale, diverse research teams outside of the CECAN case study.
2. **The Need for Flexible, Ad-hoc Adoption (4.3.6):** A key theme is that a rigid, prescriptive application of Agile is ineffective. Researchers emphasize the need for flexibility, adapting the methodology to constraints of time, labor, and resources. One expert criticizes "big box" methodologies for ignoring context.
3. **Institutional and Cultural Hurdles (4.3.7):** A significant, systemic barrier is the inherent non-agility of the traditional university system. A key issue identified is the lack of formal project management training for researchers and principal investigators, as these skills are not covered in PhD programs, forcing individuals to self-learn and complicating the adoption of new management frameworks.

Key Points from the Discussion (Section 5):

- **Overall Assessment:** The adoption of Scrum was partially successful, fostering new collaborative dynamics, but was challenging due to a lack of shared understanding and coherent application.
- **Synthesis of Challenges:** The discussion consolidates the interview findings into a list of eight key challenges, including the balance between efficiency and autonomy, limitations of online coordination, proliferation of Kanban boards, building trust, and the institutional culture of academia.
- **Connection to Literature:** The findings are linked to existing literature, confirming that Agile works best with self-organizing, flexible teams but can be counterproductive if applied too rigidly. It also notes that benefits like improved teamwork and trust, as well as limitations like difficulties for certain personalities and institutional inertia, are consistent with other studies.
- **Lessons Learned and Future Research:** The section concludes by suggesting that future studies should explore:
 - More open, less structured interpretations of APM vs. incorporating more Scrum practices.
 - The value of professional Agile facilitation (by non-researchers).
 - The effectiveness of Agile in co-located (using physical boards) vs. distributed (using digital tools like Trello) research teams.

2. Strengths and Weaknesses (4.3.5, 4.3.7, and 5. Discussion)

Strengths:

1. **Clarity and Thematic Coherence:** The findings are presented in a logical, thematic structure that is easy to follow. The transition from specific interview themes to a synthesized list of challenges in the Discussion is particularly effective.
2. **Grounding in Qualitative Evidence:** The arguments are strongly supported by direct, anonymized quotes from the researchers (e.g., RC2, RC6, RE4). This provides credibility and a rich, real-world context for the findings.
3. **Effective Synthesis:** The Discussion excels at pulling together the disparate threads from the results section into a coherent and insightful summary (the eight challenges), which greatly enhances the reader's understanding.
4. **Forward-Looking Conclusion:** The "lessons learned" segment is a strength because it moves beyond describing problems to proposing specific, actionable questions for future research, adding significant value to the paper.

Weaknesses:

1. **Weakness: Lack of a Dedicated Conclusion Section.**
→ **Issue:** This section includes the "Discussion" but not a separate "Conclusion." A formal conclusion is needed to explicitly state how the findings answer the original research questions and to what extent the research objectives were met. The current structure blends interpretation with final takeaways.
→ **Suggestion:** The paper would be strengthened by a distinct conclusion that directly addresses the research questions posed in the introduction and summarizes the main arguments without introducing new interpretation.
2. **Weakness: Findings Are Not Connected Back to the Methods.**
→ **Issue:** While the Methods section explains that a "grounded theory approach" was used to find themes, the Findings section does not show this process. It presents the themes (like "Types of research") as finished results, but doesn't show the evidence or the steps taken to build them. For example, it doesn't say how many researchers held each view, which would help the reader understand if a view was common or just from one or two people.
→ **Suggestion:** The findings would be stronger if they included more information about the evidence, such as how many interviewees mentioned a particular challenge, to give a better sense of how common each problem was.

3. Career Insights for Software Project Management

This study offers crucial, real-world lessons for a career in Software PM:

- **Agile is a Philosophy, Not a Prescription:** The most important takeaway is that successful Agile adoption requires adaptation. A skilled Software PM must tailor the framework to the specific context, team culture, and project type, rather than enforcing textbook rules.
- **Organizational Culture is a Key Success Factor:** The paper highlights that technical implementation is only half the battle. A Software PM must be adept at

navigating and influencing the wider organizational culture to create an environment where Agile principles can thrive.

- **The Critical Role of Facilitation and Training:** The challenges noted, especially the lack of training and the difficulty of the Scrum Master role, underscore that a Software PM's role is as much about coaching, facilitation, and building trust as it is about managing tasks and timelines.