



Vincent I. Prado Claudio
 Advisor: Héctor J. Cruzado, PhD, PE
 Graduate School

Abstract

This project took place at a large-scale defense contractor team using Scrum methodologies. The project aimed to improve predictability and reduce effort estimation errors by focusing on user story writing and sprint planning practices. A key finding was the importance of establishing clear definitions for "ready" and "done" for user stories, incorporating a modified INVEST method, and utilizing story writing templates. To address uncertainty during sprint planning, scales were devised to measure technical uncertainty and Subject Matter Expert (SME) availability. A decision matrix was created based on these scales to guide story inclusion within a sprint. The implemented guidelines have the potential to enhance communication, reduce rework, and improve overall project predictability.

Problem

This project addressed managing uncertainty in software development for a geographically dispersed defense contractor team. Despite widespread adoption, Agile methodologies present practical challenges. A significant hurdle arises from the lack of comprehensive upfront requirements compared to Waterfall methods. This ambiguity can lead to underestimating task complexity and, as experienced by the project team, force the withdrawal of a month's worth of work. To mitigate these issues, this project aimed to decrease development process uncertainty, with the goal of improving task completion time estimates and facilitating the creation of more predictable sprint cycles.



Methodology

The initial phase in the methodology involved data collection and analysis from user stories. The objective was to identify recurring patterns that contribute to ambiguities within the stories. The analysis focused on key factors including acceptance criteria, technical complexity, estimated completion time, and the author's area of expertise. The second phase involved a literature review focused on knowledge sharing, requirement uncertainty, and Scrum-based user story writing. This review aimed to glean insights that would later inform the development of uncertainty-reduction guidelines. The methodology culminated in the formulation of conclusions based on the findings from the initial two steps. Subsequently, techniques and guidelines were established to mitigate uncertainty within the team's development process.

Background

The ability to categorize uncertainty allows project managers to select appropriate development approaches [1]. Agile methodologies offer several practical strategies for managing uncertainty. For example, shorter iterations are typically recommended when there's high uncertainty, as they enable faster feedback loops [2]. This allows teams to address requirement ambiguities and correct course quickly, thereby reducing uncertainty. Additionally, fostering knowledge sharing within the team is crucial. Team members can leverage various methods like presentations, code reviews, or pair programming to share expertise in specific areas [3]. This collaborative learning approach can significantly reduce knowledge gaps, a major source of uncertainty. Finally, by prioritizing tasks with high learning value, team members gain essential knowledge early on, enabling them to create well-defined requirements and navigate unforeseen challenges with greater confidence [4].

Effective user story writing depends on establishing clear definitions for "ready" and "done" [5][6]. The "ready" criteria determine when a story is prepared for development. Conversely, the "done" criteria specify when a story is complete. Furthermore, to enhance user stories, the INVEST method is highly recommended [7].

Results and Discussion

In the context of this large-scale project, a specific definition of "ready" for user stories was established. This definition comprised two key criteria: comprehensive content and **INVEST** compliance (with adaptation). The team's definition of "done" for user stories encompasses two primary aspects. First, all modifications outlined in the "expected changes" section of the story must be implemented. Second, the definition encompasses any additional steps required to finalize the story, even if not directly related to the addressed feature or issue.

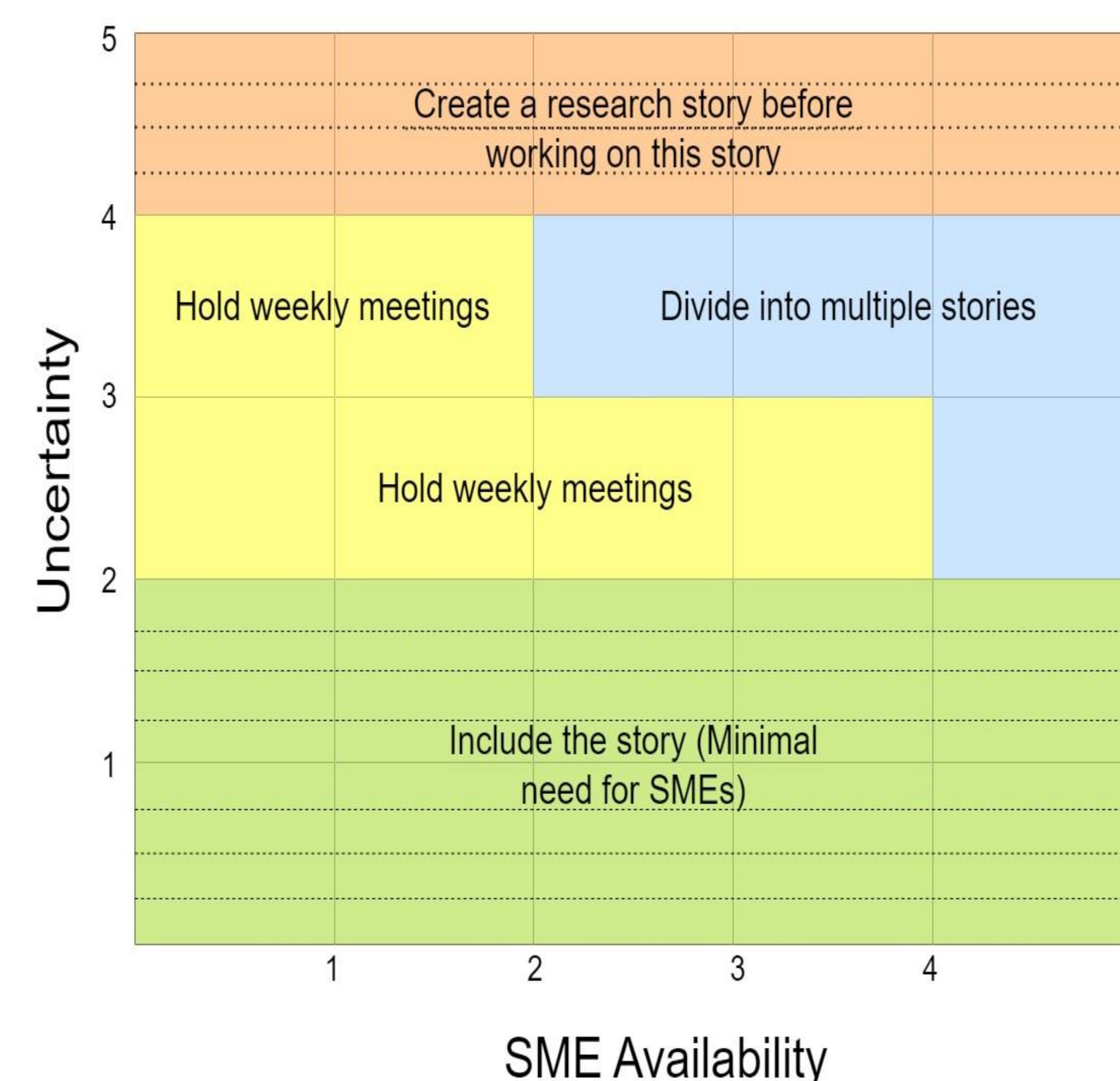
| Story Sections and Guide for Bug Fixes | | |
|--|--|--|
| Section Name | Section Description | Guiding Question |
| Description | This part will explain what the problem is. It does not have to contain the expected behavior. | What is the problem? |
| Expected Changes | Explain what the expected result is. This section should be part of the "definition of done". | What should happen to confirm the issue was fixed? |
| Steps to Reproduce | List steps to reproduce the issue. Make sure to include the dependencies. | What are the steps that should be followed to reproduce the issue? |

| Story Sections and Guide for Implementation of New Features | | |
|---|---|---|
| Section Name | Section Description | Guiding Question |
| Description | Explain why the story is needed. | What benefit will the story provide for the end-user? |
| Expected Changes | Explain what the expected result should be. | What new capabilities do these changes add to the system? What will the user be able to do that was not able to do in previous versions of the system? |
| Documentation | Add links to any documentation and/or design that was created for this new feature or any related story that was previously worked. | Has any previous work been done for this feature? Were designs and/or documentation created for this feature? |

A decision matrix to guide story inclusion within a sprint was developed. This matrix utilizes a scale for technical uncertainty for the Y-axis and a scale for SME availability for the X-axis.

| Attribute | Uncertainty Score |
|---|-------------------|
| Enhancement | 1 |
| Know how to build | 2 |
| Not sure how to build but have previous domain experience | 3 |
| Not sure how to build and no domain experience | 4 |
| Research is needed (New technology and/or no design) | 5 |

| Attribute | Uncertainty Score |
|---|-------------------|
| Enhancement | 1 |
| Know how to build | 2 |
| Not sure how to build but have previous domain experience | 3 |
| Not sure how to build and no domain experience | 4 |
| Research is needed (New technology and/or no design) | 5 |



Conclusions

This work presented guidelines developed to address uncertainty in the software development process of a geographically dispersed defense contractor team utilizing Scrum methodologies. The project focused on user story writing and sprint planning practices to enhance predictability and reduce estimation errors. Key findings included the importance of establishing clear definitions for "ready" and "done" user stories, incorporating the INVEST method with adaptations for large-scale projects, and utilizing story writing templates to ensure consistency. Scales to measure technical uncertainty and Subject Matter Expert (SME) availability were created to inform sprint planning decisions. The implementation of these guidelines has the potential to improve communication, reduce rework, and enhance overall project predictability within the defense contractor team.

References

- [1] Little, Tu, "Context-adaptive agility: managing complexity and uncertainty.", IEEE software, 22.3, 2005 pp. 28-35.
- [2] Highsmith, J. "Adaptive software development: a collaborative approach to managing complex systems", Addison-Wesley, 2013, pp.95
- [3] Dorairaj, S, et al., "Knowledge management in distributed agile software development.", 2012 Agile Conference, IEEE, 2012, pp. 64-73
- [4] Racheva, Z, et al., "Do we know enough about requirements prioritization in agile projects: insights from a case study." 2010 18th IEEE International Requirements Engineering Conference. IEEE, 2010, pp. 147-156
- [5] Atlassian, Definition of Ready (DoR) Explained & Key Components, Available: <https://www.atlassian.com/agile/project-management/definition-of-ready>
- [6] M. Rehkopf, User stories with examples and a template, Atlassian, Available: <https://www.atlassian.com/agile/project-management/user-stories>
- [7] K. Drozd, How to use Confluence and Jira Software for agile sprint planning & refinement, Atlassian, Available: <https://www.atlassian.com/agile/tutorials/jira-confluence-sprint-refinement>