Applying Distributed Cognition in Teams to Agile Processes

BLAKE BARTENBACH, University of Nebraska at Omaha, USA

Prioritizing user stories can be a cognitively challenging and draining due to the differing perspectives, technical backgrounds, and domain knowledge of individuals involved. This is furthermore complicated by the lack of an objective prioritization methodology or framework to support the process. This paper aims to identify the state of new research in the field of Distributed Cognition and whether or not it can be applied to common Agile processes for enhanced effectiveness. We summarize three papers concerning the origin of Distributed Cognition, the development of a framework to apply Distributed Cognition Theory to enhance work performed in teams, and most importantly, the application of this framework to a real-world Agile requirements prioritization setting. We find that the research in the area is sparse, and that much more research is needed. However, new studies have begun to show promising applications of this framework to Agile team settings.

Additional Key Words and Phrases: agile, dicot, dc, distributed cognition, prioritization, scrum

ACM Reference Format:
Blake Bartenbach. 2021. Applying Distributed Cognition in Teams to Agile Processes. 1, 1 (December 2021), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Prioritizing customer requirements and communicating the value of stories with stakeholders is not an uncommon challenge to anyone who has served as a product owner, proxy product owner, or even a Scrum master of a Scrum team. The attendees of a backlog refinement and/or sprint planning session can have vastly different perspectives, knowledge, backgrounds, and experience — not to mention individual stakes in the work being performed. Without a common framework for prioritization, this makes prioritizing stories and tasks challenging as the relative requirements of tasks are subjective. This is the problem Buchan, Zowghi, and Bano set out to solve.

In a theory, the customer or designated stakeholder could be deferred to to resolve every single one of these conflicts. In practice, however, the customer is rarely available to attend every single Scrum backlog refinement session. Furthermore, the customer (nor any single individual) is unlikely to possess all of the technical and domain knowledge required to actually make an informed decision to deconflict the priorities.

For example, if a database team attends a sprint planning meeting, they may be inclined to view the stories pertaining to the database as the highest priorities, as that is the component that they interface with on a daily basis. However, a network team may feel strongly that the network tasks are a higher priority. Management can attend without the technical insight and background to understand either of these tasks, and request that a metrics reporting task takes precedence. This

Author's address: Blake Bartenbach, bbartenbach@unomaha.edu, University of Nebraska at Omaha , Omaha, Nebraska, USA, 68198 6001 Dodge Street.

XXXX-XXXX/2021/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

example illustrates a common example of the challenges of a refinement session when working within service teams.

Applying the principles of DC (Distributed Cognition), the DiCoT framework, and the knowledge gleaned from previous experiments using the DiCoT framework, the main question we seek to answer in this paper is "Can the criteria identified by DiCoT analysis be distilled down to a (relatively) simple set of rules for prioritizing product backlog items?" The aim of this research is to identify such a set, if one exists, or at the very least determine the feasibility of identifying such a set.

2 BACKGROUND

Although originally geared towards software projects, many teams in today's industry have adapted Agile methodologies similar to Scrum as their project management framework. While Scrum does a great job of defining roles and ceremonies to empower teams to work more effectively, it does not prescribe the most effective way to run these ceremonies in detail. Although having any regular prioritization meeting is arguably better than none, in the interest of streamlining and improving this process some researchers have turned to Cognitive Science for answers. In this paper, we review these experiments with a skeptical lens to determine to scientific validity of applying this branch of science to a popular event in Scrum/Agile frameworks known as Agile Requirements Prioritization. In this meeting, at least from a software perspective, most often software developers, their scrum master, their product owner, and anywhere from zero to many stakeholders attend a time-boxed meeting that typically lasts an hour to discuss the priorities of the project. During this meeting decisions are made about which tasks and features need to be worked sooner, and which can be worked later. The goal of this meeting is to refine the product's backlog of work into a coherent, list of tasks that reflect the ordered priorities agreed upon by the stakeholders and development team of the product.

3 SUMMARIES

3.1 Distributed Cognition

3.1.1 Problem. Distributed Cognition (DC) theory is a branch of cognitive science that aims to understand cognition beyond the basic sense in which it is usually viewed. Normally, we view the concept of cognition as something that is limited to a single individual's "skin or skull" as the paper puts it. DC aims to understand cognition that is distributed among individuals in groups. More specifically, the goal of DC in Hutchins' paper was to identify how tasks that are commonly only done by a single individual could be distributed to a group.

3.1.2 Proposed Solution. In his research, Hutchins did not identify a solution, but rather explained the concepts as he identified them, and proposed some interesting questions that warranted further investigation. In this original paper describing the concepts of Distributed Cognition, Huchins examines phenomena observed in the world which he believes cannot be explained simply by discussing the cognition of a single individual.

3.1.3 Validation. There was absolutely no validation performed in this paper, which is certainly a cause for concern. That being said, there have been several experiments performed, and a framework developed in the spirit of DC, that serve as validation for the concepts in the present. However, in this particular paper itself, there are many ideas proposed with little to no tangible, verifiable evidence.

3.1.4 Limitations. The main limitation of the original paper describing DC is, as discussed previously, it lacked scientific validation. Another potential limitation is the lack of a clear description of

what DC is exactly. Hutchins knew that he was on to something, but it wasn't clearly described or quantified well at the time of this writing. At present, we can find much more concise descriptions about the mechanisms Hutchins was attempting to describe in this paper.

3.2 DiCoT: A Methodology for Applying Distributed Cognition to the Design of Teamworking Systems

3.2.1 Problem. Many systems are traditionally designed by envisioning a single person sitting at their desktop or laptop computer, but in practice, we know this is not the way many systems are used. In certain circumstances, for example, an ambulance or emergency room, systems are simply workstations for whoever needs them. The workspaces don't have dedicated users like we think of in a typical office setting.

3.2.2 Proposed Solution. Distributed Cognition is a framework that aims to solve this problem by considering the bigger picture. Specifically, it aims to solve design problems, but the idea proposed by the authors of this paper is that Distributed Cognition may also apply to the distribution of labor. This is what is being coined as 'DiCoT' (Distributed Cognition for Teamwork).

This was attempted in a case study using the London Ambulance Service (LAS). Data was gathered through various different roles in the organization and analyzed using DiCoT. The core principles of DC were applied to the data gathered and two alternative redesigns of the system were proposed.

3.2.3 Validation. Because the proposed methods were not implemented at the time the paper was written, there was no validation done to determine the effectiveness of the proposed modifications. Instead the author proposes that it is simply a 'proof of concept."

3.2.4 Limitations. The largest limitation of the study was the lack of any concrete proof that the methods were successful. Although the ideas generated using the DiCoT framework look promising, there is no actual implementation to gauge whether or not they improved the workflow of a team.

3.3 Applying Distributed Cognition Theory to Agile Requirements Engineering

3.3.1 Problem. In this paper[2], the authors are trying to solve the problem of story backlog prioritization with stakeholders that can commonly occur in Agile settings.

Prioritizing requirements and communicating the value of stories with stakeholders is not an uncommon challenge to anyone who has served as a product owner, proxy product owner, or even scrum master. The attendees of a backlog refinement and/or sprint planning session can have vastly different perspectives, knowledge, backgrounds, and experience — not to mention individual stakes in the work being performed. This can create a challenge without a common framework for prioritization as the relative requirements of tasks can be vastly different depending on who you ask.

If the database team attends the refinement, they may tend to view the story that benefits the database as the highest priority, as it is what they are most familiar with and interface with regularly. However, the network team may feel strongly that the network tasks are a higher priority. Management can attend without the technical insight and background to understand either of these tasks, and request that a metrics reporting task takes precedence. This example illustrates a common example of the challenges of a refinement session when working within service teams.

3.3.2 Proposed Solution. To begin to solve the problem, the authors sought to determine two important things: 1. What aspects of Agile Requirement Prioritization are cognitively significant? 2. What principles from the DiCoT framework are important in the prioritization process? One solution to this problem is to utilize a common prioritization criteria. In the paper being discussed[2],

Table 1. Synthesis Matrix

Idea	Hutchins[3]	Blandford, et al.[1]	Buchan, et al.[2]
Distributed Cognition as Cognitive Science	Originally proposed the new branch of science in an original paper.	Absolutely they see value in the application of this science, as they have created a framework to apply the principles of DC to Teams.	Yes, as they have applied DiCoT to Agile Requirements Prioritization.
Distributed Cognition as applicable to Teams	Not directly stated but strongly implied due to the nature of DC itself.	Blandford, et al. focused largely on the application of Distributed Cognition to Teams and created the DiCoT framework.	Buchan, et al. worked directly with a software development team to apply the principles of the DiCoT framework during their analysis.
Distributed Cognition in regard to Agile Processes	Agile was not specifically mentioned, as the original paper where DC was proposed as a branch of science took a very high-level view of cognition as it relates to groups.	Agile was not called out in the work done by the DiCoT researchers, as they were focused on the more generic case of simply "teams," but the principles definitely apply.	Yes, the conclusion of this research was that DC, specifically applied through the DiCoT framework's principles, is both applicable and beneficial in an Agile context.

a criteria was developed by applying DiCoT[1]: a methodology for applying Distributed Cognition (DC)[3].

The analysis identified six criteria for prioritization and three areas of distinct cognitive effort during a field study of two backlog refinement sessions for an undisclosed product.

3.3.3 Validation. There were some potentially identified threats to the validity of the study. Selection bias can not be ruled out, as the project studied was chosen by a single contact. External validity is also low, in the words of the authors, and applicability to other projects is likely to be inconsistent. Observer bias was also a possibility, as the team studied was aware that they were being analyzed during their backlog refinement meeting and may have behaved differently. The observer did attempt to mitigate the impact of this, however, by building a repertoire with the team prior to commencing the study.

3.3.4 Limitations. One limitation conceded to by the authors of the paper was that the process of applying the DiCoT framework and collecting data for analysis was very time consuming, and likely not practical to be conducted regularly.

3.3.5 Research Question. Can the criteria identified by this analysis be abstracted into a common framework? Is there some commonality between the requirements of all projects that can be quantified into a simple set of rules for prioritizing backlog items? The aim of this research is to attempt to identify such a set.

4 DISCUSSION

4.1 The Basis of Distributed Cognition

To begin to decide whether or not it is possible to extract a simple, repeatable formula to assist Agile teams with prioritization, we must first examine the credibility of the research done so far. Starting with the original paper where Distributed Cognition was first proposed by Edwin Hutchins in the 1990s, we must first ask ourselves an important fundamental question that sets the framework on which everything that follows will be built: Is Distributed Cognition simply a theory, or is it something concrete than can be quantified, defined, and clearly demonstrated?

In the opening section of the paper, Hutchins compares two published works, Mind in Society (Vygotsky, 1978) and Society of Mind (Minsky, 1985). Hutchins suggests that the titles of books are too similar to be merely a coincidence and that something larger might be happening in "systems of distributed processing."[3] This initial suggestion sets the tone for the rest of the paper quite well, as it is merely an unsubstantiated opinion that something could be at play. The issue here, is that Distributed Cognition has been described as an approach to cognitive science but this suggestion (as well as many of the others that follow) are not scientific.

In a paper referenced by Hutchins, Halbwachs (1925) declares that it doesn't make sense to discuss memory as a property of a single individual. Although this does pose some interesting questions and perhaps warrants further research, memory can certainly be discussed in the context of a single individual. The famous case of British musician Clive Wearing illustrates this counterpoint well. Wearing contracted herpesviral encephalitis and developed both anterograde and retrograde amnesia; that is, Wearing is both unable to form new memories, and unable to recall past memories. This is a case in where it certainly seems to make sense to discuss a single individual's memory.

Looking at this from a broader perspective however, it could also be argued that while it does make sense to discuss Wearing's memory as an individual, Wearing can also be viewed as a phenomenal example of Distributed Cognition at work. While Wearing's own memory has been rendered almost completely useless to him, he is able to rely on his social network to support him and provide him the memories he can no longer access himself. A good example of this is that Wearing's wife, children, and caregivers can direct him to the piano where he can play complex pieces and sight read. Although Wearing has no episodic memory, his procedural memory is still intact, and he was able to conduct a choir. Of course, these things were only possible because Clive was able to (unintentionally) leverage his social network to provide him proverbial paths to his lost skills and memories. This appears to demonstrate distributed cognition among these individuals. In specifically Clive's case, the cognition of others is required for many meaningful cognition to occur.

At another point, Hutchins states that the environment that humans are thinking in are not natural environments, but "artificial through and through."[3] We again must view this through a critical lens and ask ourselves if this is truly a scientific fact. If human beings are living in an environment created by them without outside assistance, is that not a natural environment? Is the distinction between a natural environment and an artificial environment simply that humans have built their environment? If so, this is a meaningless distinction as birds build nests, beavers build dams, and ants build very complex systems of underground tunnels. Hutchins continues on to say that, "It does not seem possible to account for the cognitive accomplishments of our species by reference to what is inside our heads alone."[3] Whether or not it seems possible is scientifically irrelevant — what is important is whether or not it is factually true.

Although Hutchins proposes some interesting theories, we highlight that at the time of the original paper, many of the ideas proposed and claims made had no scientific references or field studies to support their validity.

4.2 Applicability to Teams

When we examine the research in the previous paper, it suggests many interesting ideas and areas of further research, but there is a lack of evidence to support the theories. Regardless, the idea of distributing cognitive processes across teams remains an interesting concept to be explored further.

The DiCoT framework opens with the claim that it has, "been developed and tested within a large, busy ambulance control centre."[1] Initially, this is an impressive statement and suggests that the framework has been shown to be effective in a field test. However, upon examination of the methods used in the study, it is revealed that what this is referring to is two half-day visits to the London Ambulance Service creating a proof-of-concept that had not been tested. Blandford and Furniss applied DiCoT using the information gathered but stated, 'Since these re-designs have not been implemented, it is not possible to validate the conclusions drawn, but this exercise has been a 'proof of concept'…'[1]

While this does not prove that the DiCoT framework does not work, it also does not provide any supporting evidence that the kind of analysis and artifacts created during the research can provide any real-world benefit. Regardless of the lack of tangible evidence of the method's success, many of the principles that the method was proposes logically and anecdotally make sense. The framework provides 18 principles that should be followed when designing any team process with distributed cognition in mind. These are abstract guidelines that can be utilized to (purportedly) enhance the effectiveness of a team. An example provided by the DiCoT paper is 'creating reminders of where a task was left off at', which follows the principle, 'Creating a scaffolding to simplify cognitive tasks.' Although this is clearly a guideline for cognition in the general sense, it is not clear how this particular principle applies to Distributed Cognition specifically.

4.3 Applicability to Agile Prioritization

When we look at the theory and framework from a scientific standpoint based upon the evidence we have thus far, one may draw the conclusion that there is no valid scientific evidence to warrant further consideration. However, interestingly enough what we find is that this is where everything seems to come together.

Albeit the research examined previously does not factually demonstrate anything aside from a theory, perhaps the greatest benefit for "connecting the dots" is examining a process that is known to be cognitively demanding, and looking at the best practice processes and procedures around it

Initially, it becomes obvious that simply attending two requirements prioritization meetings is not enough research to make any definitive claims about the effectiveness of applying the DiCoT framework principles to the Agile requirements engineering process. However, the research is very promising, and certain warrants more investigation. The research team applied all 18 principles of the DiCoT framework during their analysis, and uncovered very promising insight into why these meetings can be effective, and perhaps some pitfalls that can produce ineffective requirements engineering meetings.

One issue we can immediately identify from this field study, is that it was done very early in the requirements engineering process — before development had even started. The reason this particular information could cause concern is that this phase of pre-development requirements engineering is brief, certainly compared to the amount of time that the software is actively developed and sustained. Some factors that make this relevant to point out is that the development team and software testers were not present in the meetings attended by the researchers. Moreover, these roles being present provide different perspectives into the prioritization process and they were not accounted for in this study. The dynamics between meeting attendees will change over time as they develop a shared understanding of the priorities, and new techniques may need to be explored to

handle the later stages of requirements engineering. The researches do acknowledge this, however, and agree that further research is needed in the area.

The researchers identified several relevant cognitive artifacts: user story cards that were physically arranged by the attendees, a whiteboard, and a projected computer screen. These elements are certainly present in many Agile requirements prioritization meetings, but how important are they?

What the team found, is that the meeting tended to consist of a repeating process: an unprioritized user story is taken from a group of unprioritized items, and the attendees would discuss and eventually identify the position of the element in relation to the already prioritized elements. The team also identified the cognitively demanding aspects of the process:

(1) Explaining and reasoning about stakeholders' perspectives on the value of the story.
(2) Agreeing on a position for the item in the prioritized backlog.
(3) Reasoning, questioning, and clarifying the intent of user stories to develop a shared understanding of the value provided.[2]

The team further identified six prioritization criteria used to determine the position of a story in the backlog:

(1) The strategic value of the story to the case organization.
(2) The strategic value of the story to the end users.
(3) The negative impact of not implementing the user story.
(4) The cost and effort versus the benefit of implementing the story.
(5) The negative impact on internal stakeholders with dependencies affected by the story.
(6) The negative impact of dependencies between the particular story being examined and other user stories.[2]

Furthermore, the team was able to identify and highlight exactly which processes directly connected to principles identified in the DiCoT framework. The role of the user story cards alone highlighted many principles of the DiCoT framework. Not only did they function as information radiators, information buffers, information filters, information transformers, and attention coordinators, but they also allowed the facilitator to utilize them as behavior triggers to move the meeting along.[2] Furthermore, they acted as visual indicators of the meeting progress (stories remaining to be prioritized). This connection with the DiCoT framework principles strongly suggests that these user story cards are an important part of the Agile requirement prioritization process.

The designated facilitator of the meeting was identified as an important element for the success of the meeting. A skilled facilitator will be able to identify when a consensus has been reached, and move the meeting along to the next backlog item to be discussed. However, the team cautions that this position of power can easily allow the facilitator of the meeting to exert more influence than others over discussion items.

The role of face-to-face contact (typically highlighted by the Scrum framework) was determined to be integral to enhancing the ease of communication in the meeting. The team identified a wide variety of body language used throughout the meetings such as hand gestures to provide emphasis, pointing to particular items, heads nodding or shaking in agreement/disagreement, and other nonverbal cues.

The application of the DiCoT framework and analysis performed also identified the meeting room to be an important piece to be considered. The cognitive artifacts used (the whiteboard, user story cards, and projected computer screen) were integral to the flow of information and recording decisions made during the meeting. The team also highlighted the importance of having a distraction-free area for the team to focus on the meeting.

Lastly, the diversity of the attendees was identified as an important aspect for conducting effective requirements prioritization. This was demonstrated by several events where a previously high

priority story was discarded due to new information, and a previously low priority story became high priority for the same reason. Having a variety of perspectives allows the team to make much more informed decisions about priorities and the state of work yet to be performed.

The overall results of the research were promising. The connected well-known ceremonies to principles defined in the DiCoT framework, and uncovered the reasons why they are effective. The team proposes that the DiCoT principles could be used as a checklist that can be used to analyze the effectiveness of a current Agile requirement prioritization process, or as a guide that can be consulted if the meetings are not going well. The results of the assessment can be used to facilitate positive change in the process.

5 CONCLUSION

The beginnings of the field of Distributed Cognition hinged largely on "gut feeling" rather than material evidence. The confusing results of initial attempts to quantify and define what principles govern how these processes work and can be utilized to improve real-world processes created a rocky start for the application of DC. However, due to the continued research in the field, we are beginning to see how the dots connect and can improve real-world scenarios.

Although the results of the London Ambulance research were not able to be tested, we can clearly map productive Agile requirements prioritization processes to the principles defined in the Distributed Cognition for Teams framework. This shows a promising future for the field of DC with respect to team interactions. Not only can this framework be used to produce new effective team processes, it can be used to analyze and improve existing ones.

In answering whether or not a simple framework could exist that could be applied to all Agile requirements prioritization meetings to prioritize backlog items, the answer is that we simply don't know. We have research that has identified six distinct criteria that was used to determine the value of a backlog item, but more research is needed to determine whether or not this is a complete set (and certainly whether it can be applied in a general fashion). What we do know, is that these meetings rely heavily on the participants having a shared understanding of the different aspects discussed in the meeting. This takes time to develop simply through conversation and discussion of the various topics being discussed. However, we can safely draw the conclusion that it is likely helpful and productive to utilize the DiCoT principles as a generic checklist to ensure that these meetings are being run as productively as they can be — or at the very least, identify areas for improvement.

REFERENCES

[1] Ann Blandford and Dominic Furniss. 2005. DiCoT: a methodology for applying distributed cognition to the design of teamworking systems. In *International workshop on design, specification, and verification of interactive systems*. Springer, 26–38.

[2] Jim Buchan, Didar Zowghi, and Muneera Bano. 2020. Applying Distributed Cognition Theory to Agile Requirements Engineering. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer.

[3] Edwin Hutchins. 2000. Distributed cognition. *International Encyclopedia of the Social and Behavioral Sciences. Elsevier Science* 138 (2000).