Patterns for Remote Agile Teams

BITA ZARIPOUR and MICHAEL WEISS, Carleton University, Canada

During the COVID 19 pandemic, development teams were forced to work remotely. Many of them had little experience in working remotely. There were also obstacles related to missing or inadequate infrastructure to support collaborations between distributed team members. This paper identifies patterns for remote agile teams, especially teams that have been forced to work remotely.

CCS Concepts: • **Software and its engineering** → **Patterns**; **Agile software development**.

Additional Key Words and Phrases: remote teams

1 INTRODUCTION

During the COVID 19 pandemic, development teams were forced to work remotely. Many of them had little experience in working remotely. There were also obstacles related to missing or inadequate infrastructure to support collaborations between distributed team members. This paper identifies patterns for remote agile teams, especially teams that have been forced to work remotely. Figure 1 shows a map of the patterns.
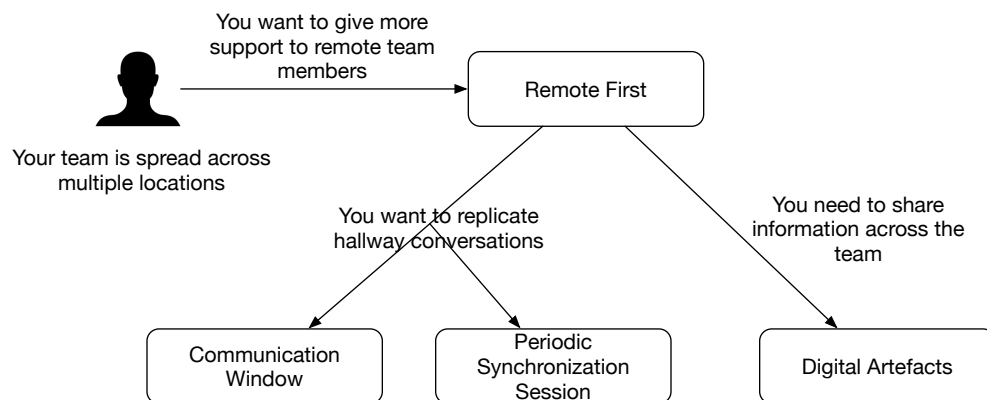

Fig. 1. Patterns for remote agile teams

The patterns were mined from examples in the literature. As described by Iba and Isaku [2012], this process involved brainstorming contexts, challenges, and solutions to those challenges, and mapping them visually to

detect commonalities between them. These "seeds" of common elements were subsequently fleshed out into patterns. These patterns were documented using the Alexandrian format.

The audience for these patterns includes organizations forced into working remotely.

2 PATTERNS

2.1 Remote First

. . . your team is spread across multiple locations. You want to give more support to remote team members.

* * *

In teams with co-located and remote members, remote members may be disadvantaged.

Working remotely may lead team members to miss out on important points in meetings, which can lead to delays in performing their daily tasks. Remote interaction also limits the transmission of verbal expressions and body language. In co-located teams, issues can often be resolved quickly by face-to-face interaction, whereas in remote teams misunderstandings are more likely to occur. In distributed teams, inadequate IT infrastructure can, furthermore, create barriers to collaboration.

Therefore,
Treat all communication as if it was remote, whether it is actually local or remote.

* * *

By treating all communication as remote, remote and co-located team members are treated the same. Remote team members will no longer feel left out from the conversation.

There are also drawbacks to this solution. It may require an upgrade to the company IT infrastructure to support remote collaboration and to the local infrastructure of each remote team member. The interaction between co-located team members may feel less natural, and switching to a Remote First approach will require those team members to adapt. Some team members may also consider remote interaction to be inferior to face-to-face interaction, so including remote members requires a trade-off between the richness of face-to-face interaction and the benefits of being more inclusive. At Debitoor [Lous et al. 2018], team members participate in daily stand-up meetings via online platforms from their workstations, effectively making everyone remote.

2.2 Communication Window

. . . your team is spread across multiple locations. You want to replicate the hallway conversations that can take place when team members are co-located for the whole team.

* * *

Remote team members do not have the same opportunities as co-located team members to interact informally and build trust and shared understanding.

Informal channels are a conduit of information that is important for building a fabric of trust among team members and provide opportunities for chance encounters through which team members can learn about one another and their role in the project. On-site team members may not trust remote team members as much as other on-site members, if they do not see them regularly. Remote team members may also consider themselves second-class members of the team. Bringing the team together physically is not always possible.

Therefore,
Create a communication window through which all team members can interact.

* * *

A communication window between on-site and off-site members increases the cohesion of a team. It provides an overall sense of activity across locations, and allows team members to interact casually outside of scheduled meetings. Communication windows can be ranked by the richness of interaction they enable: video > instant messaging, as well as by their frequency. A Periodic Synchronization Session provides a richer form of interaction, where team members get a chance to meet face-to-face, yet also less frequently.

There are different known uses. Lous et al. [2018] describe how at Debitoor monitors are set up between the two main offices to create a virtual window through which each office can receive a live-stream of the other office. A startup that works with one of the co-authors maintains a permanent web conferencing link with its remote employees. Many remote teams use instant messaging services like Slack to instantly share information with other team members and to see which team members are currently available for interaction.

2.3 Periodic Synchronization Session

… your team is spread across multiple locations. You want to replicate the hallway conversations that can take place when team members are co-located for the whole team.

* * *

Remote team members do not have the same opportunities as co-located team members to interact informally and build trust and shared understanding.

Informal channels are a conduit of information that is important for building a fabric of trust among team members and provide opportunities for chance encounters through which team members can learn about one another and their role in the project. On-site team members may not trust remote team members as much as other on-site members, if they do not see them regularly. Remote team members may also consider themselves second-class members of the team. Meeting team members only online is not sufficient.

Therefore,
Bring all team members together into the same location from time to time.

* * *

Bringing all team members together allows them to synchronize with each other. Annual conferences and quarterly synchronization sessions can periodically bring all team members into the same place [Šmite 2014]. At Automattic, the company that builds WordPress, developers work from home, but meet up from time to time at different places around the world of working sessions, as well as at an annual conference [Berkun 2013].

A liability of Periodic Synchronization Sessions is that they only happen infrequently. To enable more frequent interaction complement them by using Communication Windows.

2.4 Digital Artefacts

… your team is spread across multiple locations. You need a way for your team members to share project-related information, such as tasks and their status, or the rationale for decisions.

* * *

Project-related information is locked up in local artefacts (such as a physical Kanban board).

In agile projects, information is often shared face-to-face and informally. Because team members are co-located, they can simply write their notes on a whiteboard. Working remotely is more difficult because team members are not in close proximity to one another. It becomes more challenging for team members to be aware of how decisions were made. A lack of transparency can lead to poor team cohesion. You also do not know who may need to access the information in the future, so it is not sufficient for it to be shared among current team members.

Therefore,

Create digital versions of project-related information artefacts and share them with everyone.

* * *

Using digital artefacts increases the transparency within the team. An example of a digital artefact is a digital Kanban board that allows team members to track the progress of individual tasks [Neumann et al. 2021]. Digital artifacts can be accessed equally by remote as well as co-located team members. They also create a record of team activity, and can be used to onboard future team members. Open source projects are distributed by default, and thus have always practiced sharing information through digital artefacts such as project mailing lists, bug trackers, and wikis. These artefacts are also known as "informalisms" [Scacchi 2005]. They are both less formal than traditional documentation, but also archived in digital form and, therefore, searchable.

3 CONCLUSION

The patterns in this papers are, in many ways, children of the Covid pandemic. Throughout the pandemic, teams had to learn how to work remotely, often from one day to the next. For agile teams, this can be particularly challenging, as agile methods emphasize frequent interaction between team members, and often presuppose that teams are co-located. Some of the practices described existed prior to Covid, but had only been used by either decentralized teams in large companies; early leaders, companies like Automattic or Debitoor, who decided to forgo physical offices; as well as, of course, open source projects. As we are coming out of the pandemic, we can expect organizations to rethink their need for physical offices and permanently adopt practices like these.

ACKNOWLEDGMENTS

REFERENCES

Scott Berkun. 2013. *The Year without Pants: WordPress.com and the Future of Work*. John Wiley & Sons.

Takashi Iba and Taichi Isaku. 2012. Holistic Pattern-Mining Patterns. In *19th Pattern Languages of Programs Conference*. Hillside, 1–19.

Pernille Lous, Paolo Tell, Christian Bo Michelsen, Yvonne Dittrich, Marco Kuhrmann, and Allan Ebdrup. 2018. Virtual by design: How a work environment can support agile distributed software development. In *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*. IEEE, 97–106.

Michael Neumann, Yevgen Bogdanov, Martin Lier, and Lars Baumann. 2021. The Sars-Cov-2 pandemic and agile methodologies in software development: a multiple case study in Germany. In *International Conference on Lean and Agile Software Development*. Springer, 40–58.

Walt Scacchi. 2005. Socio-technical interaction networks in free/open source software development processes. In *Software Process Modeling*. Springer, 1–27.

Darja Šmite. 2014. Distributed project management. In *Software Project Management in a Changing World*. Springer, 301–320.