



# Right Thoughts & Right Action: How to Make Agile Teamwork Effective

by Torgeir Dingsøy, Diane Strode, and Yngve Lindsjörn

Teamwork is critical in many industrial sectors. When creating complex software solutions, most companies and public institutions organize work within cross-functional teams and follow the principles of Agile development. This approach to knowledge-intensive work seeks to empower team members, ensures that the most competent people make decisions, and manages uncertainty by allowing members to learn and adapt as work progresses.

Agile methods offer much guidance on teamwork. The “Principles Behind the Agile Manifesto” highlight self-organized teams and face-to-face conversations which, according to the principles, is “the most efficient and effective method of conveying information.”<sup>1</sup> Moreover, “a great development team,” according to a white paper on scrum teams, “trusts each other” and “pursues technical excellence.”<sup>2</sup>

Advice is abundant. For example, Google’s re:Work model offers advice to development teams in the form of five key factors for successful teams, including “psychological safety,” “structure and clarity,” and work that the team members consider meaningful.<sup>3</sup> There is also general advice from years of studies of teamwork and from empirical studies on Agile development teams. However, there has yet to be a model that draws together the knowledge from all these sources and specifically focuses on the effectiveness of Agile teamwork.

To fill this gap, we have developed an Agile Teamwork Effectiveness Model (ATEM).<sup>4</sup> Our model is based on a review of empirical studies on Agile development teams, general studies of effective teams and teamwork, and practitioner advice. We also incorporated findings from our own two case studies and 22 focus groups. Though primarily intended for colocated Agile software development teams, the increasing adoption of Agile methods outside IT departments may make the model valuable for other Agile workplaces.

## Why Do We Need a Team Effectiveness Model?

Team effectiveness refers to how team members interact to accomplish their project’s goals, while delivering quality work within budget and on schedule. Ineffective teamwork is detrimental — it can reduce job satisfaction, interfere with team learning, generate knowledge and skill silos, and generally impede progress.

Teamwork effectiveness models are based on accumulated empirical observations and reasoned arguments, and identify and describe key factors necessary for effective teamwork. Our model, tailored for Agile practitioners, offers insights into effective Agile teamwork and explains how certain Agile practices support it.

The ATEM builds on the Big Five model<sup>5</sup> of teamwork effectiveness. It consists of three coordinating mechanisms that facilitate and support five teamwork components critical for team effectiveness (see Figure 1). The ATEM includes observable behaviors that practitioners can use to evaluate teamwork effectiveness (see Table 1 — and Table 2, later in this article) and, if necessary, make informed decisions to improve it.

## Coordinating Teamwork

The ATEM coordinating mechanisms — shared mental models, mutual trust, and communication — facilitate and support each other and the five components. For example, a team needs a shared mental model of the work to be done before assisting a team member struggling with workload issues (redundancy); a team needs mutual trust when offering peer feedback to avoid hurt feelings; and a team needs communication to develop the shared mental model and mutual trust. Communication is also vital for all five components of teamwork.

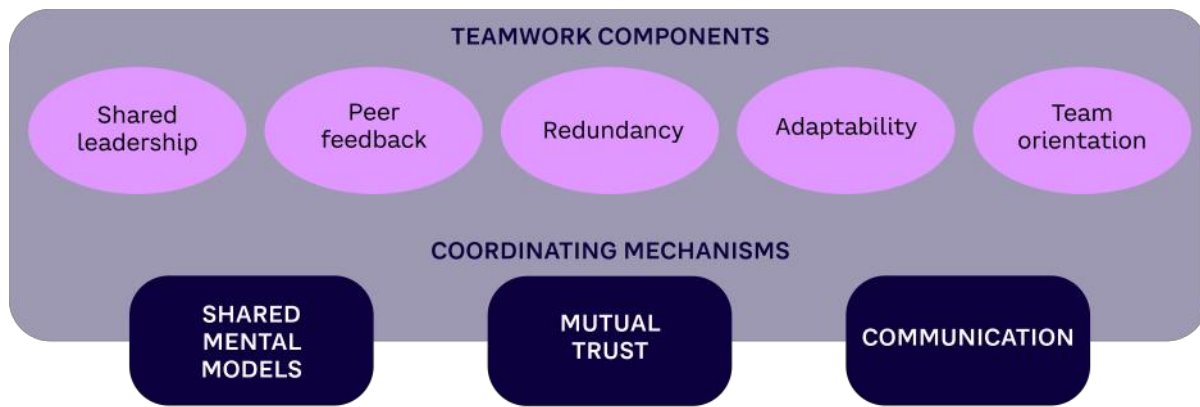


Figure 1. The Agile Teamwork Effectiveness Model (ATEM)

TEAMWORK COORDINATING MECHANISM	TEAM BEHAVIORAL MARKERS
<p><b>Shared mental models</b></p> <p>“An organizing knowledge structure of the relationships among the task the team is engaged in and how the team members will interact.”</p>	<ul style="list-style-type: none"> <li>• Anticipates and predicts each other’s needs</li> <li>• Shares common understanding of: goals, tasks, work process, product, individual skills, and expertise</li> </ul>
<p><b>Mutual trust</b></p> <p>“Shared belief that team members will perform their roles and protect the interests of their teammates.”</p>	<ul style="list-style-type: none"> <li>• Adheres to information sharing</li> <li>• Is willing to admit mistakes and accept feedback</li> <li>• Supports team social climate</li> </ul>
<p><b>Communication</b></p> <p>“The exchange of information between a sender and a receiver irrespective of the medium.”</p>	<ul style="list-style-type: none"> <li>• Follows up on progress of tasks</li> <li>• Visualizes project information</li> <li>• Facilitates informal communication</li> </ul>

\*Definitions revised from Big Five model (Salas et al.); behavioral markers revised from ATEM

Table 1. Coordinating mechanisms with behavioral markers in the ATEM

*Shared mental model* refers to the common understanding that develops among team members over time as they become familiar with each other and the situation. Elements of a shared mental model might be: knowledge of the skills/expertise held by team members and knowledge about the situation, the product, the development environment, the process to follow, and the way the team prefers to interact.

An effective shared mental model thus helps coordinate a team, enables a smooth working environment, and ensures fewer interruptions, since team members will spend less time learning from each other by observing or asking (and answering) questions and more time on completing tasks.

To evaluate a team’s shared mental model, it’s crucial to check whether team members can anticipate and

predict each other’s needs accurately. Do all the team members show knowledge of, or have the opportunity to find out, the goals, the current tasks, and the work process? Are they familiar enough with the product — the stories, design, architecture, and codebase? Are they aware of each other’s skills and expertise? When gaps in knowledge are found, focus on practices to fill those gaps. These include keeping teams together over time and integrating newcomers quickly, holding effective daily stand-up meetings and retrospectives, promoting social activities like group lunches, utilizing pair programming and task boards, and creating an environment where team members can readily express their knowledge, issues, and concerns and promote a common understanding.

*Mutual trust* refers to the shared belief that team members will perform their work roles and protect the

interests of their teammates.<sup>6</sup> This is critical for a team's psychological safety<sup>7</sup> (as in the re:Work model), which refers to a climate where people are comfortable being and expressing themselves.

Mutual trust is especially important when teams are self-organizing and rely on empowered members. A lack of trust prompts more formal information sharing within the team. To support mutual trust, the team climate must enable its members to admit mistakes and accept feedback. An unsupportive climate can trigger greater individual orientation, more formal communication, and an increased hierarchical project structure.

*Shared leadership occurs when a team organizes itself to achieve goals.*

How can a lack of mutual trust be identified? A poor social climate is one where several conflicts are brewing and team members are actively avoiding each other. Issues remain unresolved and keep reappearing. A lack of mutual trust can occur when information transfer is formal and not open and when the team members are unwilling to admit mistakes and accept feedback.

*Communication* refers to the exchange of information within the whole team and between individual team members — and is essential in any type of teamwork. A lack of open communication may hinder the sharing of knowledge along with the experience necessary for staying abreast of task progress, for offering feedback and support, for discussing issues and identifying solutions, and for understanding requirements/architectures and translating them into software.

Informal communication is the most effective way for Agile teams to share and discuss progress of tasks, so team members are often colocated to stimulate informal and open communication. This makes it easier to visualize information using project boards.

A lack of communication can be identified by checking whether a team has open and informal communication. For a colocated team, examine, for example, the physical environment: Does the entire team work in a shared open plan office? Are there team rooms and open spaces that enable easy access to each team

member? Does the team share information openly using physical or virtual boards to visualize task progress, allocation, and dependencies? Does the team hold regular and unscheduled meetings to openly discuss issues and solutions?

## Components of Effective Agile Teamwork

After establishing the coordinating mechanisms, a team must then establish the five key components for effective teamwork: shared leadership, peer feedback, redundancy, adaptability, and team orientation (see Table 2). According to our focus group material, development teams are not sufficiently aware of peer feedback and redundancy.

*Shared leadership* occurs when a team organizes itself to achieve goals. In Agile teams, shared leadership can be evenly distributed, making everyone a leader, or rotated among team members based on expertise. Agile teams can also adopt a servant leadership style. In this approach, the servant leader initially empowers the team, then steps back. When the team has to interact with parties outside the team, such as stakeholders or others in the organization, the servant leader takes responsibility.

Shared leadership is important because it creates effective Agile teams that are empowered and self-organizing, allowing them to make decisions among themselves to set and meet the goals of the team and organization.

Shared leadership is evident when the team solves problems together, determines performance expectations and acceptable interaction patterns, and seeks and evaluates the information that affects how the team functions. Using Agile practices and automated tools, the team synchronizes and combines individual team member contributions and makes decisions together regarding the roles team members will take, the plans for product development, and the purpose and frequency of meetings. If these behaviors are lacking, then shared leadership should be improved.

*Peer feedback* is the ability to give accurate feedback to team members based on a common understanding of the team environment. This component is important to correct any errors made by team members, especially in stressful situations. In a development team, errors might include a misunderstanding of user needs,

TEAMWORK COMPONENT	TEAM BEHAVIORAL MARKERS
<p><b>Shared leadership</b></p> <p>“The ability of the team to direct and coordinate their activities, assess team performance, assign tasks, develop team knowledge, skills, and abilities, motivate one another, plan and organize, and establish a positive atmosphere.”</p>	<ul style="list-style-type: none"> <li>• Facilitates team problem-solving</li> <li>• Determines performance expectations and acceptable interaction patterns</li> <li>• Synchronizes and combines individual team member contributions using Agile practices combined with automated tools</li> <li>• Seeks and evaluates information that affects team functioning</li> <li>• Uses Agile values and methodologies to determine team member roles</li> <li>• Uses Agile values and methodologies to determine frequency and type of preparatory meetings and feedback sessions</li> <li>• Servant leader facilitates boundary-spanning function</li> <li>• Agile team practices provide planning function</li> </ul>
<p><b>Peer feedback</b></p> <p>“The ability to develop common understandings of the team environment and based on those understandings to give accurate peer feedback to team members.”</p>	<ul style="list-style-type: none"> <li>• Identifies mistakes and lapses in other team members’ actions</li> <li>• Contributes regular feedback regarding team members’ actions to facilitate self-correction</li> </ul>
<p><b>Redundancy</b></p> <p>“The ability to anticipate other team members’ needs through accurate knowledge about their responsibilities. This includes the ability to shift workload among members to achieve balance during high periods of workload or pressure.”</p>	<ul style="list-style-type: none"> <li>• Recognizes workload distribution problem in team</li> <li>• Shifts work responsibilities to underutilized team members</li> <li>• Supports completion of whole task (or parts of tasks) by other team members</li> </ul>
<p><b>Adaptability</b></p> <p>“The ability to adjust strategies based on information gathered from the environment through the use of backup behavior and reallocation of intra-team resources. Altering a course of action or team repertoire in response to changing conditions (internal or external).”</p>	<ul style="list-style-type: none"> <li>• Identifies cues that a change has occurred, assigns meaning to that change, and develops new plan to deal with the change</li> <li>• Identifies opportunities for improvement and innovation for habitual or routine practices</li> <li>• Remains vigilant to changes in the internal and external environment of the team</li> </ul>
<p><b>Team orientation</b></p> <p>“The propensity to take others’ behavior into account during group interaction and the belief in the importance of team’s goals over individual members’ goals.”</p>	<ul style="list-style-type: none"> <li>• Considers alternative solutions provided by teammates and appraises that input to determine what is most correct</li> <li>• Performs increased task involvement, information sharing, strategizing, and participatory goal setting</li> <li>• Sticks together and remains united</li> </ul>

\*Definitions revised from Big Five model (Salas et al.); behavioral markers revised from ATEM

Table 2. Teamwork components with behavioral markers in the ATEM

developing code that is not aligned with the team’s design decisions, or introducing errors in the new code or the existing codebase. Self-correction based on internal feedback is likely to be more efficient, quicker, and more useful than correction from stakeholders

outside the team. Moreover, receiving late feedback on errors can generate extra work; a developer might have already moved on to new tasks and will need additional time to recall the context of the error.



A team can use peer feedback to identify its mistakes and offer regular feedback to its members so they can self-correct. Such feedback is typically given in meetings when demonstrating the product, when discussing improvements in a retrospective, or when working closely together (e.g., pair programming). Peer feedback can also be given indirectly through, for example, automated code testing. A lack of such arenas for feedback indicates a need to improve team effectiveness.

***Team orientation is important because it means the team is focused on working together to achieve a single goal, rather than on their individual goals.***

*Redundancy* means that team members have insight into the workload of teammates and use that knowledge to rebalance and reallocate tasks when necessary. For redundancy, knowledge of each team member's responsibilities is needed. When teams lack redundancy, they cannot readily reallocate their workloads. If workloads rise, some members will work too much, and others not enough. Over time, this can threaten a team's sustainable work pace.

Insufficient progress on tasks or the number of unfinished tasks worked on in parallel may signal a lack of redundancy. A team with good redundancy recognizes its workload distribution problems, shifts task responsibilities, and relieves team members under pressure by reallocating tasks or subtasks to other members.

*Adaptability* refers to a team's ability to adjust its behavior and actions to accommodate changes in the internal or external environment by using the available resources. Adaptable teams accommodate changes that impact their efforts, and sidestep effects that harm their performance and hinder goal completion. For example, a team unable to adapt to changing requirements could waste time working on the wrong problem. When faced with a technological change, a team must be able to adapt by rapidly learning new technology or risk becoming less effective.

To evaluate team adaptability, check whether the team takes an interest in the internal or external future

events that could affect it. Does the team know the organizational strategy and plans so it can identify potential change? When changes occur, are there arenas to discuss them and plan suitable adjustments?

*Team orientation* refers to the consideration of other team members' behavior when the team interacts and the belief that the goals of the team are more important than the goals of the individual team members. Unlike other teamwork components, team orientation is about attitudes rather than behaviors.

Team orientation is important because it means the team is focused on working together to achieve a single goal, rather than the individual goals of each team member. An individual orientation puts the benefits of teamwork at risk. Without team orientation, heavily interdependent tasks will become harder to complete, and changes to plans may cause the team to struggle. Moreover, rigid roles at the team level can result in team members not engaging in certain tasks and goals; for example, a dedicated test role on a development team might lead to others not taking responsibility for the tasks associated with this role.

As an attitudinal dimension, team orientation might be easier to evaluate based on feedback. A lack of team orientation is evident when team priorities are not considered while choosing work tasks, when there is a lack of interest in offering peer feedback, when assistance is not given to team members with too much workload, or when there is a lack of cohesion.

## How ATEM Can Improve Teamwork

The ATEM is useful in three ways:

1. In colocated Agile teams, facilitators, Agile coaches, and team members can better understand their teamwork by regularly reflecting on how they meet each factor in the model (e.g., during retrospectives). They can use the behavioral markers to evaluate their teamwork and help identify challenges and ways to improve their teamwork.
2. Parts of the model may be relevant in contexts beyond single, colocated Agile software development teams. What would differ among a very small team, a distributed team, a multi-team, and teams doing safety-critical development? A very small team will find it reasonably easy to develop the

coordinating mechanisms; a distributed team will need to invest more time to develop an infrastructure that enables good coordinating mechanisms; a team in a multi-team setting would need to ensure that it develops effective coordinating mechanisms with other teams; and teams making safety-critical software would need a special emphasis on peer feedback to reduce errors.

3. This model is applicable even if you are on an Agile team not doing software development. If your work is knowledge-intensive and similar to software development, or involves many dependencies between work tasks, and if your task priorities change frequently with changes in customer needs or available technology, then ATEM should fit. However, if the context is more traditional, we recommend the original Big Five model.

Using ATEM will direct attention to key team effectiveness factors and offer the *right thoughts* that enable the *right action* for Agile teams.

## Acknowledgments

We are grateful to the band Franz Ferdinand. Our title comes from their album *Right Words, Right Thoughts, Right Action*.<sup>8</sup> Our forthcoming article in *Empirical Software Engineering* includes further acknowledgment regarding research funding.<sup>9</sup> We are also grateful to the following people who provided comments on earlier versions of the manuscript: Anastasiia Tkalic at SINTEF Digital, Marit Larsen at TechnipFMC Norway, and Kristin Wulff at Kantega and the Norwegian University of Science and Technology.

## References

- <sup>1</sup>"Principles Behind the Agile Manifesto," accessed February 2022.
- <sup>2</sup>Overeem, Barry. "Characteristics of a Great Scrum Team." InfoQ, 15 April 2016.

<sup>3</sup>Duhigg, Charles. "What Google Learned from Its Quest to Build the Perfect Team." *The New York Times Magazine*, 25 February 2016.

<sup>4</sup>Strode, Diane, Torgeir Dingsøy, and Yngve Lindsjørn. "A Teamwork Effectiveness Model for Agile Software Development." *Empirical Software Engineering*, forthcoming, 2022.

<sup>5</sup>Salas, Eduardo, Dana E. Sims, and C. Shawn Burke. "Is There a 'Big Five' in Teamwork?" *Small Group Research*, Vol. 36, No. 5, 1 October 2005.

<sup>6</sup>Salas et al. (see 5).

<sup>7</sup>Edmondson, Amy C. "Psychological Safety, Trust, and Learning in Organizations: A Group-Level Lens." In *Trust and Distrust in Organizations: Dilemmas and Approaches*, edited by Roderick M. Kramer and Karen S. Cook. Russell Sage Foundation, April 2007.

<sup>8</sup>Franz Ferdinand. "Right Action (Official Video)." YouTube, 7 July 2013.

<sup>9</sup>Strode et al. (see 4).

*Torgeir Dingsøy is a Professor in the Department of Computer Science at Norwegian University of Science and Technology (NTNU), Norway, and an adjunct Chief Research Scientist at SimulaMet research laboratory. Dr. Dingsøy's research focuses on teamwork and learning in software development and Agile methods for large software projects and programs. He has been published in the areas of software engineering, information systems, and project management. Dr. Dingsøy earned a PhD from NTNU. He can be reached at [torgeir.dingsoyr@ntnu.no](mailto:torgeir.dingsoyr@ntnu.no).*

*Diane Strode is a Senior Lecturer at the School of Information Technology, Whitireia Polytechnic, New Zealand, and a research fellow at Open University, UK. She also has experience as a software developer for Mobil Oil Australia. Dr. Strode's research centers on Agile software development and coordination. She has been published in the areas of information systems and software engineering. Dr. Strode earned a PhD from Victoria University of Wellington, New Zealand. She can be reached at [Diane.Strode@whitireia.ac.nz](mailto:Diane.Strode@whitireia.ac.nz).*

*Yngve Lindsjørn is an Associate Professor in the Department of Informatics at the University of Oslo, Norway. He spent 10 years as a researcher at the Norwegian Computing Center and has 13 years' industry experience as a project manager and CEO of a software company. Previously, Mr. Lindsjørn was project manager for a research project investigating teamwork in software development. His research includes software development methods and teamwork factors that influence software project success. He can be reached at [ynglin@ifi.uio.no](mailto:ynglin@ifi.uio.no).*