Team Dynamics as a Predictor of Software Quality

Fazeel Gareeboo

fgareeboo@ea.com

Abstract

Software quality is frequently attributed to technical factors, yet this perspective overlooks the profound impact of team dynamics on software outcomes. Drawing on four decades of industry experience, this paper highlights the critical roles of collaboration, communication, and psychological safety within development teams. Through reflective analysis, I examine how team dysfunctions have historically influenced software quality, including the often-overlooked decisions regarding team structure and communication practices. The paper introduces Jim and Michele McCarthy's Core Protocols as a practical, evidence-based framework for measuring and enhancing team dynamics. By sharing candid, experience-based insights, this work offers practitioners actionable strategies for leveraging team dynamics as a central driver of consistently higher software quality, empowering them to foster more effective, resilient, and high-performing teams.

Biography

I am a software development director at Electronic Arts, managing a team that provides automated builds and smoke-tests for the game teams. I have worked in the computer graphics industry since 1985, working in CAD/CAM, graphics card drivers and a gaming library developer. I grew up on the island of Mauritius and worked in Europe before settling in the United States.

1 Introduction

Traditionally, software quality is defined by the product's ability to do what it's supposed to do. We recognize the following dimensions as critical to defining and assessing quality in software:

- Functionality,
- Reliability,
- Efficiency,
- Usability,
- Maintainability, and
- Portability

The software team's dynamics affect the quality of the software because they impact multiple areas including:

Factor	**Impact on Software Quality**
Communication	Better problem-solving -> Fewer errors -> faster correct delivery
Trust	Higher collaboration -> better software
Shared Values	Alignment of goals -> more efficient work -> better software
Coordination	Effective use of expertise -> reduced bottlenecks -> faster correct delivery
Leadership	Adapts to change -> correct delivery despite requirement changes

There are several software development frameworks that aim to optimize team dynamics. The objective of those frameworks is to deliver a successful software project (which includes factors other than software quality - e.g. on time and within budget). Frameworks like Agile, the 'Spotify model', feature-based and component-based teams.

In my experience, most software projects start with an allocated number of contributors with some effort made to structure the team from the start. The team then goes through several iterations of organizational and process adjustments to address team dynamic issues that come up. Eventually, the successful teams get to a point where they are operating smoothly and any team issues that come up are

resolved relatively quickly. Success means the team delivers a product that does what it is supposed to do, on time and within budget.

This generally corresponds to Tuckman's "Stages of Team Development":

- **Forming** - Team comes together and members get to know each other
- **Storming** - Conflicts and disagreements between team members
- **Norming** - Team starts to resolve differences and improve
- **Performing** - Team functions at a high level of efficiency. Norms and processes established.
- **Adjourning** - The work is completed and team disbands.

The challenges with this process are that:

- Not all teams successfully get to the performing stage. Many remain in the earlier stages and either miss their shipping dates or deliver something that is distinctly sub-par or ship at the expense of burning out team members. The outcomes for such teams can span the spectrum between success and failure, but usually at a high cost (product quality, team morale, …).
- The teams that successfully navigate through to the performing stage do so after a relatively significant amount of time. Not many teams "hit the ground running". Those that do are typically teams that were together on a previous project so the team dynamics have been optimized already.

The purpose of this paper is to highlight the criticality of addressing team dynamics and propose a way to do that right at the start of the project and in a relatively deterministic manner. This mitigates the risks that:

- The team never makes it to the 'performing' stage and
- Allocates a fixed time for the team to hit their stride

2 Team Dysfunctions and Their Impact

In this section, I describe two cases where team dysfunction affected software outcomes. The main issues outlined below were not technical, but more about team dynamics and behavior.

2.1 Software Quality team dysfunction

The first case was a software quality team (SQT) that was mandated to provide automated testing to other software teams. The SQT was divided into three sub-teams:

- A test automation analyst team that was responsible for translating the needs of the client software teams into automated testing requirements.
- A scripting team that would write scripts for automated testing, as specified by the analysts and running on an in-house automated testing infrastructure.

- A tools team that was responsible for the development and operations of the core testing infrastructure.

When I joined the team as the manager of the tools team, there were several significant issues, including:

- The testing infrastructure was not stable, so the tools team spent a significant amount of time doing support.
- The automation analysts were frustrated because they were not getting traction on their requests. They had no information on estimates for their requests, much less expected delivery dates.
    - The automation analysts wanted to get positions on the client teams, and they wanted to do their best to please the clients.
- The scripting team felt squeezed between the two groups, because most of the client requests included new tool features that they could not estimate and they regularly interacted with the analysts, who were running the existing automated tests and were complaining to them when the infrastructure failed.

The solution was simple, but a 'hard sell'. Stop all new developments and focus on fixing the test infrastructure. This reduced the time that the tools and scripting teams spent on support and the analysts were happier with the existing test scripts running successfully. Once the infrastructure was stable, the team could then estimate and implement new requests from the clients.

Before those changes were implemented, the existing team members already knew what the problems were and the solution. But no one would voice it out and they were all expecting existing leadership to solve. One key issue was that the sub-teams were not operating as one team – problems were blamed on the other sub-team(s) instead of coming together and resolving the issues.

The result was that the existing automated testing solution was operating at a significantly lower success rate than expected, and client teams were getting frustrated because of that and because they were not getting any feedback from their new requests.

2.2 Junior engineer not getting direction

In this case, a junior engineer working on a feature had a few potential implementations. He tried to book time with his technical lead, but for reasons outside of his control, he could not. He tried several alternative ways to get time from the tech lead, but was unsuccessful. The feature was at risk and management assigned the feature to a more senior engineer, who just picked one of the implementations and proceeded. The feature was still delayed, but there was now a reliable ETA.

The primary issue here was the tech lead had too many things on his plate and did not resolve his issue with his leadership. The junior engineer could also have asked other team members (other than tech lead) for help (for example, from the senior engineer who eventually took on the task). The issue came

up during a retrospective, so this was a learning opportunity for both team members specifically, and for the team as a whole.

Being comfortable with asking for help is a key part of a healthy team. Problems get solved much sooner and team members can grow in their career.

3 Introducing the Core Protocols

The McCarthy Core Protocols are a set of structured interaction patterns designed to help teams achieve high performance, improve trust, streamline decision-making, and foster psychological safety. Developed by Jim and Michele McCarthy during years of research and teamwork experiments, these protocols encode the best practices and behaviors observed in effective teams (McCarthy 2007, Tarnowski 2018, Kasperowski 2025).

3.1 Purpose and Philosophy

The Core Protocols aim to:

- Create results-oriented, high-performing, collaborative teams.
- Boost psychological safety, trust, and shared vision.
- Provide explicit frameworks for conversations, decisions, conflict resolution, and self-reflection.
- They are built upon several foundational principles, including:
    - Positive bias: Focusing on constructive, forward-moving behavior.
    - Freedom and autonomy: Every team member has the right to fully participate, opt out, or request help as needed.
    - Transparency and self-awareness: Encouraging disclosure of thoughts, feelings, and needs.
    - Action over hesitation: Favoring active engagement and immediate action when necessary.

(Henderson 2016, Tarnowski 2018)

3.2 Structure

The use of the Core Protocols is preceded by agreeing to act according to the Core Commitments.

3.2.1 The Core Commitments

These are the behavioral standards expected of team members, including:

- Engage when present.
- Clearly state what you want, think, and feel.
- Always seek and offer effective help.
- Favor action; propose, support, or improve ideas proactively.
- Support the best idea, regardless of source.
- Prefer perceiving over being perceived.
- Use teams for challenging tasks.
- Speak to improve results.
- Practice rational, results-focused interactions.
- Disengage from unproductive situations.

- Do now what should be done soon.
- Use the Core Protocols (or better) when applicable.
- Never harm or tolerate harm for protocol adherence.
- Never do anything dumb on purpose.

(Henderson 2016, Tarnowski 2018)

3.2.2 The Core Protocols

These are specific, repeatable conversation structures for frequent team interactions. Key protocols include:

- Check In: Start meetings with a brief emotional statement to build connection and focus.
- Check Out: Opt out of participation when necessary, signifying disengagement respectfully.
- Pass: Decline to participate in any activity without negative consequences.
- Ask For Help: Explicitly request assistance, tapping into collective team expertise.
- Decider: Rapid, unanimous decisions on proposals.
- Perfection Game: Offer improvement-oriented feedback on ideas.
- Personal Alignment: Reflect on and articulate personal desires or obstacles.
- Investigate: Ask questions to deeply understand another's perspective or actions.

(Henderson 2016, Goldminz 2018, Tarnowski 2018)

3.3 Examples in Practice

Check In: “I am glad I feel energized and focused.” The group responds: “Welcome.”

The purpose of the Check In is to provide a safe avenue for team members to disclose their current emotional state. This is very helpful to the rest of the team because it could explain the team member’s unexpected behaviour. To keep it safe, the default is for team members not to inquire about the contents of the Check In unless the speaker explicitly states that she/he is open to talk about it.

Decider: “Does everyone agree we should adopt this approach?” Rapid thumbs-up or down; unresolved dissent handled by the Resolution protocol.

Ask For Help: “Lisa, will you help me troubleshoot this issue today?”

3.4 Impact and Uses

The Core Protocols are recognized for their positive impact on team dynamics, especially in software, product development, and agile teams. They help teams avoid time-wasting, build mutual respect, make decisions quickly, and continuously improve their collaboration. (Jocham 2016, Levison 2021, Tarnowski 2018).

They provide a practical, codified framework for teams to communicate, collaborate, and excel together by making the "rules of engagement" explicit, actionable, and safe for all participants. Adopting this proven framework saves the team having to create and evolve a set of their own.

3.5 Theoretical basis and evidence for their effectiveness

The theoretical foundations of the Core Protocols are laid out in the book:

"Software for Your Head: Core Protocols for Creating and Maintaining Shared Vision" by Jim and Michele McCarthy

The current version of the protocols is a result of refinement over the last 30 years.

While the objective of Google's Project Aristotle (New York Times Magazine 2016) was to identify what makes teams effective at Google, the findings that team dynamics are the foundation of high performing teams support the contention that they have an impact on software quality. A high performing team produces good quality software.

There have been several reviews of the Protocols, including one from Tom DeMarco (DeMarco 2003, 24). Here are some excerpts from the review:

> I write to call your attention to a unique new body of work that has the potential to change much of what we do in the software industry.
>
> …it is a worthwhile attempt to deal with the true essence of our work.
>
> Reading "*Software for Your Head"* will change you and change your practice.

Other reviews include:

- Blog on Boot Camp by Adam Feurer
- Core Protocols google site
- Core Protocols for Shared Vision site
- Blog on Team Tips by Paul Reeves
- An approach to build great (agile) teams by Hannes Horn
- Emotional Intelligence—the secret ingredient behind high performing teams by KD Singh Arneja
- Better meetings with the Core Protocols by Peter Antman
- Richard Kasperowski's site has lots of resources, including:
    - Positive Bias: the Foundation for High-performance Teams

4 Practical Implementation of the Core Protocols

The best way to implement the use of the Core Protocols is to have your team participate in a McCarthy TeamWork BootCamp. The resources page on LiveInGreatness has a list of certified coaches you can reach out to.

4.1 Key Features of the McCarthy TeamWork BootCamp

The TeamWork BootCamp is an immersive workshop designed to rapidly build high-performance teams by instilling the use of The Core Protocols. It functions as a "teamwork laboratory," simulating the full life cycle of a product development project in just five days.

Before BootCamp, participants agree to upload the Core Commitments.

Participants form a team, envision a product, agree on how to make it, and deliver the product by the end of the week. This compresses the learning and experiences of a long project into a few days, allowing for intense team bonding and accelerated learning. BootCamp centers around learning and practicing the Core Protocols.

Personal Alignment: During the first part of the BootCamp, participants articulate personal virtues they want to develop (such as courage, trust, or presence). They practice the Core Protocols by 'Investigating' each other, Asking for Help, and play Perfection Game on each other's alignment.

Web of Commitments Ceremony: After individual alignments, the team shares all their commitments, signals, and desired outcomes, building a strong sense of support and shared accountability.

Manager Involvement: The facilitators sometimes take on the "manager" role, encouraging team members not just to participate, but to frequently ask for help, reflecting a common workplace challenge.

Project Delivery and Reflection: At week's end, teams present their finished product and reflect on what they've learned and built together, often facing the challenge of reaching unanimous agreement on what to showcase.

4.2 A framework for measuring and improving team dynamics

One of the artifacts of a BootCamp is the Team alignment record - typically a spreadsheet documenting what each team member's alignment is, and how the rest of the team rates it (Perfection score).

An example Team alignment sheet:

(While this may seem very personal, one of the key features of boot-camp is to accelerate team bonding. One of the most effective ways to do that is for individuals to become open and vulnerable to each other, hence the individual alignment that is shared with the team).

Name	Alignment	Signal	Response	Short term	Medium term	Long-term evidence	Perfection score
James	Self-Awareness	I do the Vulcan salute	You respond and say "Live long and prosper"	I will keep a self-awareness diary and record my feelings every day	After 1 month, I will share my diary with my partner and ask them to rate the accuracy	After 3 months, the accuracy of my feelings will be 9/10 or more.	7
Maria	Peace	Nod	Deep breath	I will keep a peace diary and record my 'level of peace' every day	After 1 week, I will calculate a cumulative average of my level of peace and post it in a visible place	After 3 months, my level of peace will increase by at least 50%	9
Mary	Passion	When you see me	You say passionately "It's Mary !"	I will find at least 3 ways that I can show my passion in the world	I will start measuring my passion in those 3 areas on a daily basis	After 3 months, my passion will increase by at least 50%	5
Miguel	Self-care	Both palms out and facing up	You respond with Namaste sign	I will find at least 3 areas in my life where I am lacking self-care	I will start measuring my self-care in those 3 areas on a daily basis	After 3 months, my self-care will increase by at least 50%	4
Muhammad	Wisdom	Right hand on my heart	You respond with Jnana Mudra	I will find at least 3 people who can rate my wisdom and ask them to send me a score	I will ask my 3 contact for a score every month	After 3 months, my wisdom score will increase by at least 30%	6
Ram	Courage	Hands up for high-five	You high-five me	I will find at least 3 areas in my life where I am lacking courage	I will start measuring my courage in those 3 areas on a weekly basis	After 3 months, my courage will increase by at least 30%	4
Wei	Health	I jump up in the air	You also jump up in the air	I will find at least 3 measures of heath for me (e.g. blood pressure, cholesterol, resting heart rate, ...) and get a baseline measure	I will measuring and record those 3 measures on a weekly basis	After 3 months, my health scores will increase by at least 30%	9

The perfection score is the number (1 to 10) that is given by other team members when playing the perfection game on the alignment. For example, Wei gets a score of 9 on her alignment and the improvement (to get a 10) would be to specify by when she will pick the 3 health measures.

The greater the average perfection score for the team, the more likely that each member of the team knows what they want out of life, and other team members also know what they are looking for (since they played perfection game on it). This can be a good measure for the quality of the Team dynamics.

5 Actionable Strategies for Practitioners

Getting the best team dynamics you can from the get-go is a worthwhile return on time invested. It ensures that:

- You get the best out of the individuals on your team and
- Team members work together to achieve the common goal

I propose the **Core Protocols** as a set of best practices for **team dynamics**; the **McCarthy TeamWork BootCamp** as a **training program** for using the Core Protocols and the **Alignment score** as a way of measuring the **quality** of the team. The product reflects the team.

We currently do not have a lot of data on the use of the core protocols in software teams. It would be very helpful if you could record data on your implementation, so this could be collated and shared with the rest of the industry. Data such as what measures you have taken on improving team dynamics and the results in terms of software quality (e.g. number of bugs per line of code, severity index, other metrics that are relevant for your product). This could then be shared at a later PNSQC.

5.1 Recommendations for integrating team-focused practices into existing workflows

Real life is messy. You don't always start off with your full team on day one and you don't always have the time and budget to send them off to a weeklong BootCamp.

If attending a boot-camp is not possible, then introducing the use of the protocols during existing meetings can be an alternative.

- Start with the core commitments, then the check-ins, and then use the others when relevant and appropriate.
- Once you have started using some of the protocols as a team, explore the possibility of running a "Personal Alignment express" for your team.
- If you cannot get an alignment score for the team because of lack of time, set up a short team activity where they create something as a team. A team painting, drawing, a story; anything that:
    - Can be done in a short amount of time (say less than 15 minutes)
    - Is 'visible' - there is an artifact
    - Your/others score for the quality of the artifact can become your team quality score.

6 Conclusion

Team dynamics have an impact on the quality of software. I argue that team dynamics are a key driver of software quality - not directly, but through the actions of the team members in making sure that the software does what it is supposed to do.

I share two case studies where improvement in team dynamics could have significantly improved the performance of the team and resulted in better products and services.

I present a proven framework for measuring and improving team dynamics. I share strategies to implement the Core Protocols if time/funding is not available to send the team to a TeamWork BootCamp.

I encourage all individuals and teams to invest some time in implementing the Core Protocols. My experience has been that you get a significant return on your time investment, and it can help you to create teams that are self-managing and successfully deliver high quality software.

7 References

DeMarco, Tom, June 2003, Communications of the ACM, Vol. 46 No. 6, p24-25.

Goldminz, Itamar, July 3 2018, “Live in Greatness protocols [McCarthy]”, https://orghacking.com/live-in-greatness-protocols-mccarthy-f2afd10c3130, (accessed August 3, 2025).

Henderson, Joel Parker, “The Core Protocols - by McCarthy”, October 14 2016, https://github.com/joelparkerhenderson/ways-of-working/blob/main/doc/the-core-protocols-by-mccarthy/index.md, (accessed August 3, 2025).

Jocham, Ralph, “The Core Protocols”, April 21 2016, https://www.scrum.org/resources/core-protocols, (accessed August 3, 2025).

Kasperowski, R, “Jim and Michele McCarthy: The Origin of the Core Protocols”, posted February 17, 2025, https://kasperowski.com/podcast-6-jim-and-michele-mccarthy/, (accessed August 3, 2025).

Levison, Mark, “Core Protocols”, posted December 2021, https://agilepainrelief.com/glossary/core-protocols/, (accessed August 3, 2025).

McCarthy, Jim and Michele, “Software for Your Head: Core Protocols for Creating and Maintaining Shared Vision”, January 1 2002, Addison-Wesley Professional

McCarthy, Jim and Michele, “Origin of the Core Protocols”, posted 2007, https://liveingreatness.com/origin-of-the-core-protocols/ (accessed August 3, 2025).

New York Times Magazine, “What Google Learned From Its Quest to Build the perfect team”, February 25 2016, https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html (accessed August 17 2025)

Tarnowski, Micheal, “The Core Protocols – Make Yourself And Your Team Great”, posted January 7 2018, https://www.plays-in-business.com/the-core-protocols-make-yourself-and-your-team-great/ (accessed August 3, 2025).

Weimar, Emily; Nugroho, Ariadi; Visser, Joost; Plaat, Aske; Goudbeek, Martijn; Schouten, Alexander P; “The Influence of Teamwork Quality on Software Team Performance”, January 22, 2017, Tilburg University; Radboud University, Nijmegen; Software Improvement Group, Amsterdam, https://arxiv.org/pdf/1701.06146 (accessed August 16, 2025)