

A blurred background image showing several people working on laptops in an office environment. One person in the foreground is seen from behind, wearing headphones and looking at a laptop screen displaying a chart. Other people are visible in the background, also working on their computers.

The Trust Protocol

Human APIs for High-Performing Teams

Kamil Senecki

Have you ever worked in a TEAM?

That implies there was a moment you had to **join** the team.

New company, new people, maybe a new product...

Entering the room, the Slack, the call — everyone is **new**.



No idea what to do and say.

Who Am I?

- 🐦 **Kamil Senecki**
- 🎯 Scrum Master for GenAI teams at **Pega**
- 📱 Pega: Low-code platform for enterprise applications
- 🗑 Joined **Pega** one year ago

Everything was new:

- Industry
- Company
- Team
- Product



The Captain America Moment

You know that Marvel scene where Steve Rogers wakes up in the modern world?

Same skills, completely different context.

That was me.



The First Day

-  Office tour
-  Many new faces
-  Acronyms everywhere
-  Products to learn
-  Trainings

"Kamil, can you join the team next week and start putting everything in place?

We have high expectations."

What Makes a Team?

GROUP vs TEAM

A Group:

- Same Slack channel
- Same manager
- Same product

A Team:

- **Shared purpose** — not just location
- **Collective ownership** — over product AND code
- **Learning culture** — supportive, no blame

Google's Project Aristotle

They studied **hundreds of teams** to find what made some successful.

What was the **#1 factor**?

- ✗ Not technical skill
- ✗ Not experience
- ✗ Not having a star performer



Psychological Safety

The best teams were the ones where people felt **safe to take risks**, ask questions, and admit mistakes.

That's what we want to build with **TRUST**

Tuckman's Model

The Stages Every Team Goes Through

Tuckman's 5 Stages

1

Forming

"Nice to meet you"

Everyone's polite. No one's being real yet.

2

Storming

"Wait, what?!"

Conflict shows up.
Different approaches clash.

3

Norming

"Let's figure this out"

Standards emerge. Ways of working settle.

4

Performing

"Flow state"

Actually delivering. High effectiveness.

5

Adjourning

"Evolution"

Team evolves or ends.

The Truth Nobody Tells You

You will spend **most of your time** in Storming and Norming.

Performing is earned, not guaranteed.

The Reset Effect

Every time you add **ONE new person** to a team, you partially reset back to **Forming**.

Forming → Storming → Forming → Storming → Forming → Storming...

This is why onboarding matters so much.

It's not just about learning the tools. It's about re-establishing trust with every new configuration of the team.

The Trust Deficit

Lencioni's Model

We have just met.

Would you **trust me** with your team right now?

If the answer is "not yet" — that's **honest**. And that's the point.

Trust isn't guaranteed. **It's earned.**

Lencioni's Pyramid of Dysfunctions



Every level is caused by the one below it. **It all starts with trust.**

The Positive Flip

The Dysfunctions:

- ✗ Absence of Trust
- ✗ Fear of Conflict
- ✗ Lack of Commitment
- ✗ Avoidance of Accountability
- ✗ Inattention to Results

The Solutions:

- ✓ Build trust → through **vulnerability**
- ✓ Master conflict → through **honest debate**
- ✓ Achieve commitment → through **clarity**
- ✓ Embrace accountability → through **peer pressure**
(the good kind)
- ✓ Focus on results → through **collective ownership**

The Trust Equation

How to Build & Measure Trust

Charles Green's Trust Equation

$$\text{Trust} = (C + R + I) / S$$

C

Credibility

R

Reliability

I

Intimacy

S

Self-Orientation

Breaking Down The Equation

Credibility

"Do I believe what you say?"

Do you know your stuff? Does your code work? Is your documentation solid?

Intimacy

"Do I feel safe with you?"

Do people feel comfortable admitting mistakes? Do you create space for honesty?

Reliability

"Can I depend on you?"

Do you follow through? Do you show up? Can people count on you?

Self-Orientation

"Whose interests come first?"

Are you here for the team? Or for yourself, promotion, being in the spotlight?

The Denominator is Dangerous!

You can have **high credibility, high reliability, high intimacy...**

and it can still be hard to build trust with high **self-orientation**.

People can smell it.

Building Trust

Practical Tactics

1. Lead with Vulnerability

"Vulnerability is not weakness. It's the birthplace of innovation, creativity, and change."

— Brené Brown

In Practice:

- "I don't understand this system yet. Can you walk me through it?"
- "I made a mistake in that deployment. Here's what happened."
- Posting in incidents channel: "I broke prod. Here's the RCA."
- Sharing failures and lessons learned

This is hard. Especially when you're new and want to impress people.



2. Create Your Personal README

README files help understand services, right?

Why don't we have that for **people**?

I created 2-3 slides about myself and shared them during a meeting.

What to Include:

- My experience
- Funny facts about me
- What is important for me
- How I prefer to give/receive feedback
- My communication style
- What drives me nuts
- My blind spots and quirks

No need for others to guess. Just share what works for you.

3. Working Agreements

Make the **implicit** explicit.

Every team has unwritten rules. The problem is, **new people don't know them**.

Example Agreements:

- "Be on time to meetings"
- "Hold yourself and others accountable"
- "Be transparent — no surprises"
- "Assume good intentions"
- "Trust, but verify"

Key Points:

- Not rules from above
- Team creates them **together**
- Revisit them **regularly**
- Write them **down**

4. The Responsibility Flywheel



Start Small

- Maybe it's the flaky test suite no one wants to touch
- Maybe it's updating documentation
- Maybe it's running the retro
- Take ownership of **something**

It's a flywheel — but **you have to push it** to get it started.

Team Values

Not Just Scrum Values

The Serum Team Values



Commitment



Courage



Focus



Openness



Respect

When your team **commits** to goals and holds each other accountable...

When people have the **courage** to raise difficult problems...

When everyone **focuses** on what matters most...

When there's **openness** to feedback...

When people genuinely **respect** each other...

That's **trust in action** and it works whether you're using Scrum, Kanban, or something else.

Trust isn't magic. It's not luck.

Trust is a protocol.

It has inputs and outputs. It can be built, **systematically**, over time.

"Trust is built in drops and lost in buckets."

Every interaction is a deposit or a withdrawal.

Every meeting

Every code review

Every Slack message

Every time you show up

My Challenge to You



This Week

Have a coffee chat with someone on your team you don't know well.

No agenda. Just connection.



This Month

Write your Personal README and share it with your team.

Experience the difference yourself.



This Quarter

Facilitate a Working Agreements session.

Make the implicit explicit.

Keep trying

"Success consists of going from failure to failure without losing enthusiasm."

Trust is hard. Culture is hard. Building teams is hard.

But it's also the most important thing we do.

Because tooling scales systems and **TRUST** scales teams.

Let's Connect!



Website

zwinnapanda.pl



LinkedIn

linkedin.com/in/kamilsenecki



Slides

github.com/ksenecki/slides/tree/main/tr

Scan for Slides



Questions?



Let's discuss!