



PRUGGERN, 2026

**Alexandra Schladebeck, Bart Knaack, Huib Schoots, João Proença,
José Díaz, Linda van de Vooren, Lisi Hocke, Maaïke Brinkhof,
Patrick van Enkhuijzen, Rahul Verma**

© 2026 Agile Testing Days. All rights reserved.

We, the people in the software engineering community, are facing challenging times within our industry. These challenges are related not only to new realities within tech and society, but also to engrained issues that have plagued us for decades. This impacts us, and the people who are affected by the systems we build.

What we see

When looking at software development and its bigger context through the lenses our experiences and backgrounds have given us, we see systemic patterns that lead us to software that is unfit for purpose and causing harm. Too often, we confuse writing code with building systems. AI/LLM didn't create these problems, it's amplifying the issues, making them harder to ignore.

What we acknowledge

We work within constraints that most of us cannot change on our own: budget pressure, time pressure, organizational politics, and market forces that keep demanding more for less. We are not going to pretend these constraints are easy to navigate or that they are fair. When the system pushes us into the compromises we disagree with, that is not a personal failure.

We also recognize that we are not powerless. Even if it is worth trying to change the system actively through direct and indirect influence we should not overestimate our chances of success, specifically not without teaming up with other communities. Moreover there are many unknowns about what the future might bring, so we have to stay prepared for change and adapt to it.

We realize that software is here to stay and will play an increasingly important role in our lives and society. We can therefore not afford to just look away. We take ownership of the consequences of our actions.

What we believe

We believe our community has the power and responsibility to affect its future, as well as the software systems we build. We acknowledge our responsibility to discover possible paths, share options with each other, and provide guidance for each other.

SOFTWARE AND ENGINEERING PRACTICES.

Sound engineering is the foundation. Architecture, observability, security, governance: these are not optional extras or bureaucratic overhead. They are disciplines that turn code into systems people can actually rely on.

SYSTEMS THINKING.

Software does not exist in isolation. It lives in ecosystems, organizations and interacts intricately into people's lives. We look above and beyond our immediate task and responsibility and understand the impact, feedback loops, and the consequences nobody planned for.

SOCIETY.

Our work affects people, not just systems. We think about who benefits, who gets left out, who might be harmed, and what happens when things go wrong.

CRITICAL THINKING.

We ask questions that matter, even when they are uncomfortable. This means we examine our own assumptions, we weigh evidence honestly and we're willing to challenge and change convenient answers, including the ones we came up with ourselves.

VALUES.

Ethics matter most when they are inconvenient. The current situation isn't certainly only about systems and constraints. It's also and most importantly about people. We have to stand for transparency, accessibility, accountability and ethics. This is especially true when we are under pressure to cut corners.

SKILLS AND CONTINUOUS LEARNING.

We strongly believe we will grow ourselves as humans when we improve our skills, and develop our knowledge to work with (any) new technology that comes our way to support us in creating better systems (for the people who use them).

We see our community, where individuals are understandingly concerned about their purpose, both from a personal perspective but also from the one from organizations. Enhancing their skills will be key to solidifying these two perceptions of purpose, by supporting their continuous learning and going back to core fundamentals that we firmly believe will help them face the state of the art.

PURPOSE AND WELL-BEING.

Purpose gives direction and meaning to our efforts and work. Working in an environment where there is a continuous pushed to go faster and work smarter, burnout is a serious threat. Burnout destroys judgment, courage and creativity. Taking care of ourselves, setting boundaries, reflecting and supporting (ourselves and others) is often framed as a weakness. We identify it as an integral part of being a professional.

THE NEXT GENERATION.

The future depends on what we pass on. We mentor, share and teach what we know. We make sure knowledge is preserved (in systems and in people).

This guidance will also be paramount to prepare the next generation, who will need directions for their careers and positive message framing. We do everything we can to make the path less intimidating for those coming after us.

What we commit to

Software is not going away. The world is going to need skilled, thoughtful software professionals through this period of change and well beyond it, even if the shape of those roles is still unclear or unknown.

We recognize that people hold values, either expressed through their own personal boundaries that are not always voiced, or ethics that organizations may not respect. We will listen, acknowledge, protect and make space for these values.

These values are key when reflecting and deciding on the impact systems have in society and if they are filling their needs. This reflection requires empathy not just for our users, but also for all the participants in the lifecycle of producing software. We commit to elevating our targets to beyond the (usually) easy to identify values like shareholder value.

We act through the platforms we build, the mentoring we do, the training we offer, the frameworks we create, and the honest conversations we are willing to have.

There is hope for all, and it starts with us.

Disclaimer: We're aware that this statement comes from a group in a position of privilege. We have the agency and autonomy to make a change or choice, that not everyone has. We cannot foresee the lived experience of all people who care about the outcomes of software systems. We can acknowledge and consider this in every action and choose curious inquiry first.