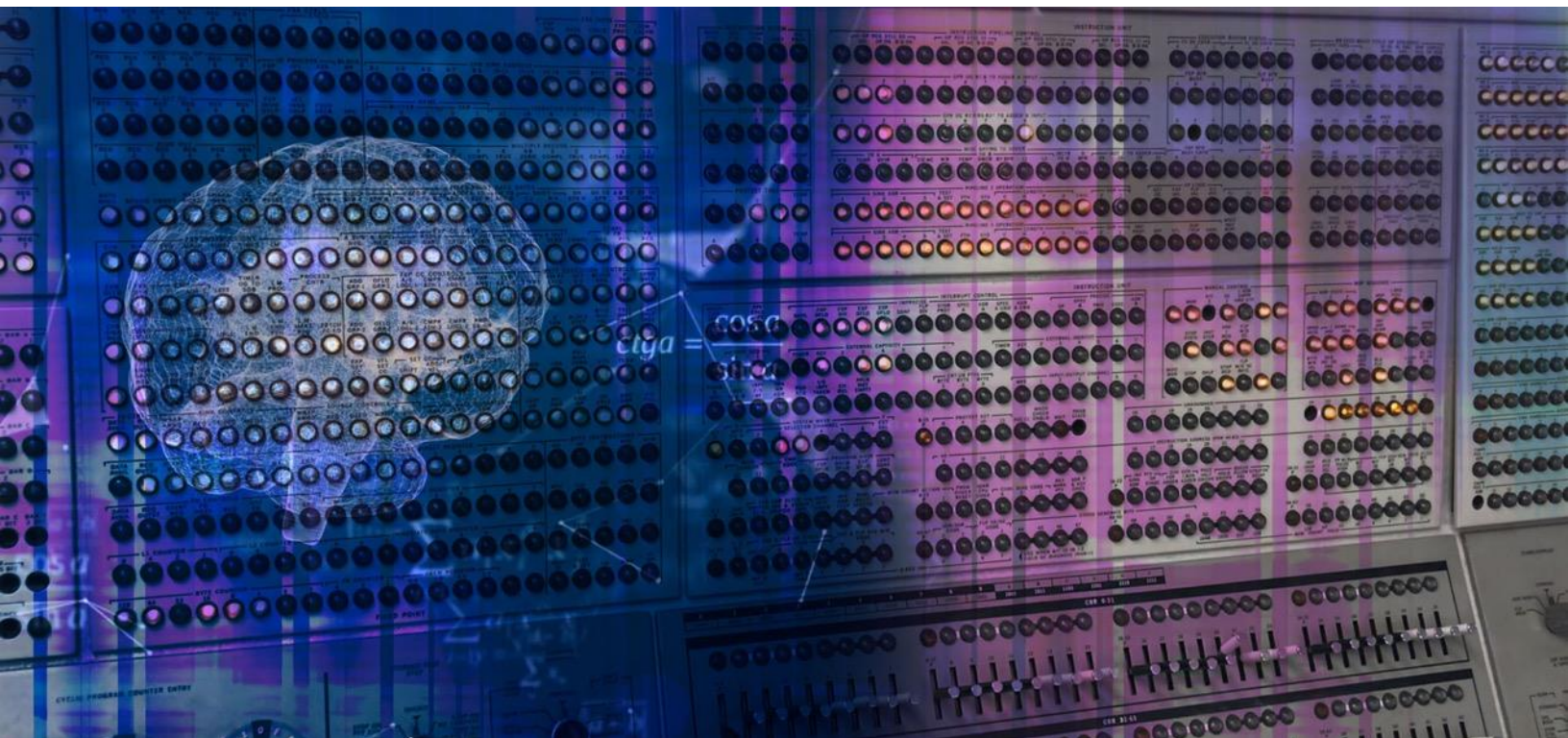




Intellyx™



White Paper

Double, Double, Toil is Trouble

Jason Bloomberg

President, Intellyx

March 2022



"I love all the busywork I have to do in order to reduce toil," said no developer ever.

Today, every developer (and architect, SRE, DevOps engineer, tester, etc.) knows that toil is the arch-nemesis of productive, efficient, innovative, and elegant work. All those roadblocks and activities that detract from the job at hand constitute toil. And the less toil, the better.

It doesn't take long for development teams seeking to improve flow and reduce toil to run into the toil Catch-22: all the tools on the market that promise to reduce toil add to the busywork – and thus counterproductively add to the toil.

Some tools like Slack have broad usability across great swaths of the organization. And while no one would argue that Slack isn't powerful and mature, its creators never intended for it to be a tool solely for software development teams. It's simply too general-purpose. Making it a good fit for development tasks takes a lot of, you guessed it, toil.

Some tools like Slack have broad usability across great swaths of the organization. And while no one would argue that Slack isn't powerful and mature, its creators never intended for it to be a tool solely for software development teams. It's simply too general-purpose. Making it a good fit for development tasks takes a lot of, you guessed it, toil.



Other collaboration tools like Trello are very good at what they do – in Trello's case, automating the use of Kanban boards. Most tools like Trello integrate with many other tools in hopes of facilitating the movement of data in order to make teams more productive when they use the tool.

Unfortunately, many such tools (not to single out Trello) all suffer from the same limitation: their specialization requires users to switch contexts as they move from one tool to another.



Such context switching disturbs the flow. Interrupts thought processes. Slows people and processes down. The end result: missed deadlines, performance issues, and customer churn. All because context switching adds to toil.

Integration vs. Collaboration

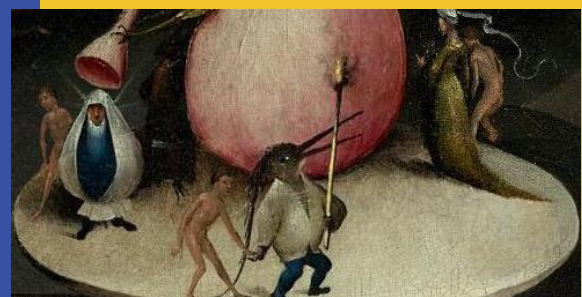
Today there are numerous tools for every aspect of software development, from DevOps tools to Value Stream Management platforms to ops-focused tools for SREs, operations personnel, the security team, and anybody else responsible for their corner of the digital world.

What most if not all of these tools have in common is their ability to integrate with each other. Every tool sports a RESTful API or two, and they all follow increasingly standardized data formats for the information they exchange with each other.

This willingness to integrate gives certain vendors a Sauron-like vision of power as they attempt to build the one platform to rule them all. "Connect everything to our platform," they like to say, "and we'll control everything for you."

Such platforms can certainly provide value – especially when compared to a mishmash of disconnected tools – but they constrain the way the development team wants to work. Instead of supporting collaboration, they dictate it, or worse: they offer superficial collaboration capabilities that don't integrate well with existing tools.

Every tool sports a RESTful API or two, and they all follow increasingly standardized data formats for the information they exchange with each other. This willingness to integrate gives certain vendors a Sauron-like vision of power as they attempt to build the one platform to rule them all. "Connect everything to our platform," they like to say, "and we'll control everything for you."



As a result, development teams must come to some sort of compromise between different visions of how to get work done – and hammering out such compromises amounts to nothing but more toil.



Putting Collaboration at the Center

Mattermost takes the opposite approach. Its platform focuses on supporting collaboration however team members want to collaborate and makes collaboration the foundation of the platform and the workflows it supports.

This platform integrates with all the other tools as well, adding to the standard approaches with options that give developers more control, even to the point of contributing to Mattermost source code to make changes to the core.

The Mattermost platform centers on three capabilities: Channels, Boards, and Playbooks.

- Channels offer team communication, all in one place, at every stage of the software development lifecycle. Think Slack, only Mattermost built Channels from the ground up to support development teams.
- Boards help with project management and team alignment. If Trello and Slack had a baby focused exclusively on software development, it would be Mattermost Boards.
- Playbooks are interactive checklists for prescribed workflows. Playbooks are like runbooks, except they can handle any workflow that might be useful to anyone across the lifecycle, from coding to testing to release to managing incidents.

Playbooks combine collaborative checklists with integrated, channel-based communication, automations, and reporting. Like runbooks, only without the sole emphasis on 'run.'

Mattermost is ideal for any development team's day-to-day collaboration, but it really shines when collaboration is top of mind, like for software lifecycle processes or incident response ('war room') activities.

By Developers, For Developers

The Mattermost team [built Mattermost with Mattermost](#) – making their own company their first beta tester.



The Mattermost platform is open source and actively nurtures and supports community contributors and upstream projects.

It features both a modern GUI as well as CLI-oriented shortcuts that form the crux of its support for collaboration. As a result, the Mattermost desktop app allows team members to navigate among tools and servers without even touching the mouse.

Mattermost also helps development teams manage notification overload with custom collapsible sidebar categories and collapsed reply threads that allow team members to selectively follow or unfollow specific threads.

Mattermost reduces toil by automating processes, centralizing functionality, and offering documentation and predictability across all processes and teams.

Finally, Mattermost also offers Controls that provide data protection, information governance, and security – without adding to toil.

The Intellyx Take

If you break down a development team into its basic elements, you have the planners (architects), creators (developers) and fixers (SREs). There are other roles to be sure, but even in this simplified view, every role partly consists of software development.

It makes sense, therefore, to build a collaborative tool like Mattermost around the central software development activity: shipping quality software.

If a tool can remove the toil from around coding – and from every role that involves coding, including planning, creating, and fixing – then it has a fair shot at supporting collaboration across all aspects of the software development process.

The Mattermost platform is maturing quickly, in part due to the contribution of customers and other community members. If you haven't taken a look recently, it's worth checking out the latest collaboration story from Mattermost.



About the Author: Jason Bloomberg



Jason Bloomberg is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

He is founder and president of Digital Transformation analyst firm Intellyx. He is ranked #5 on [Thinkers360's Top 50 Global Thought Leaders and Influencers on Cloud Computing](#) for 2020, among the top low-code analysts on the [Influencer50 Low-Code50 Study](#) for 2019, #5 on Analytica's [list of top Digital Transformation influencers](#) for 2018, and #15 on Jax's [list of top DevOps influencers](#) for 2017.

Mr. Bloomberg is the author or coauthor of five books, including [Low-Code for Dummies](#), published in October 2019.

About Mattermost

[Mattermost](#) is an open source platform for secure collaboration across the entire software development lifecycle. Hundreds of thousands of developers around the globe trust Mattermost to increase their productivity by bringing together team communication, task and project management, and workflow orchestration into a unified platform for agile software development. To learn more, visit www.mattermost.com.

Copyright © Intellyx LLC. Mattermost is an Intellyx customer. None of the other organizations mentioned in this paper is an Intellyx customer. Intellyx retains final editorial control of this paper. Image credit: Hieronymus Bosch (public domain).