



Intellyx™



Special Report: *Using Playbooks to Boost Productivity by Balancing the Need for Free-Flowing Collaboration and Structured Workflows*

Charles Araujo

Principal Analyst, Intellyx

March 2022

It's an almost impossible task. You need to help your organization execute its strategic transformative initiatives, continually boosting your development team's productivity *while* ensuring ever higher levels of reliability.

Sure. That sounds easy.

It seems an impossible task mostly because it requires you to balance two seemingly counter forces: the need for developers to embrace free-flowing creativity and collaboration and the need for structured controls of the process.

Add in increasing levels of volume, complexity, and velocity and you arrive at impossible, right?

Not if you change the way you approach the challenge. That's what [Mattermost Playbooks](#) is intended to help you do.



Finding Balance to Deliver Outcomes

Ironically, we've arrived at this juncture because the pendulum has swung too far in an opposite direction.

In the dark ages of a decade or two ago, development was highly controlled and structured. And moved at a snail's pace as a result.

Waterfall methodologies ensured that everyone followed a highly prescribed development lifecycle, with checks and controls every step of the way. But while controlled, the process was too slow to keep up with an increasing rate of change and demand for new development.

Organizations rightfully pivoted.

They mostly got out of the way of developers and (among other things) embraced collaboration and communication tools to allow developers to move at pace.

This free-flowing development approach led to an explosion of production, but also created both inefficiencies and risks. Developers were often left with unstructured and uncontrolled development processes that left teams open to security vulnerabilities, missed bugs, and overall lower quality code. It also created inefficiencies as teams often unnecessarily recreated already proven processes and procedures.

The reality is that development organizations need a balance between free-flowing and structured to make things work most effectively.

A little structure can help reduce toil, ensure compliance, and sometimes just keep teams from continually reinventing important, but low differentiation actions like onboarding, incident response, and roadmap updates.

While too much structure can slow things down (and no one wants to go backward), too little can introduce error, rework, and unnecessary overhead later in the development process.



The good news is that this isn't an either-or proposition. There is a way to introduce higher levels of structure without inhibiting the free flow of developer collaboration.

Checklists and Workflows to the Rescue

In 2010, [surgeon Atul Gawande](#) published *The Checklist Manifesto*. Reflecting on his years dealing with life-and-death procedures, he realized that the answer to these increasingly complex challenges was simple: *the checklist*.

As the New York Times said in its review of the book, "No matter how expert you may be, well-designed checklists can improve outcomes." It's a point that everyone from surgeons to U.S. Air Force Top Gun pilots have understood and adopted.

"No matter how expert you may be, well-designed checklists can improve outcomes."



Shockingly, however, most development teams have ignored this simple, yet potent and effective tool.

In his book, Gawande explains that as volume and complexity increase (exactly what's happened in enterprise development), it overwhelms our "individual ability



to deliver...correctly, safely, or reliably.” Checklists, however, provide a layer of protection by identifying “minimum necessary steps and mak[ing] them explicit.”

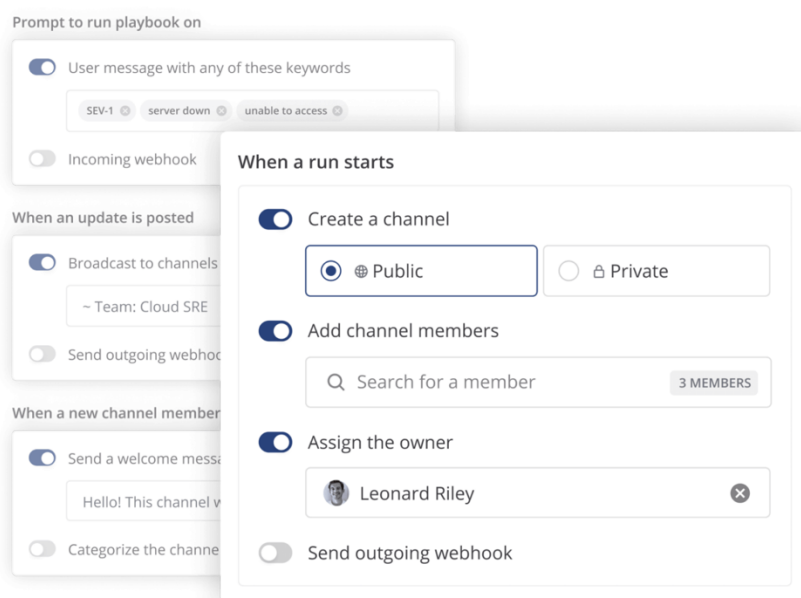
Most interestingly, he points out that checklists are not intended to be how-to guides or comprehensive directions, but rather simple tools that support and enable the already robust skills of qualified experts by freeing them to focus on more important activities.

Or to paraphrase the great archeologist Professor Henry Jones, *development teams should write these things down so they don't have to remember them.* (From, of course, Indiana Jones and the Last Crusade. Am I dating myself?)

The premise behind Mattermost's Playbook feature is to help teams bridge and balance the need for free-flowing collaboration with the necessary structure and guidance that checklists and prescribed workflows can offer.

As Gawande's book makes clear, Professor Jones reminds us, and Mattermost's use cases prove, appropriately applying checklists and workflows help the very best performers perform even better by relieving their minds of having to keep track of the routine and mundane.

In so doing, this almost counterintuitive approach that adds structure actually increases productivity and improves the business outcomes that development efforts are seeking to deliver.



¹Source: Mattermost



The Intellyx Take: The Playbook Path to Improved Productivity & Better Outcomes

With its Playbook feature, Mattermost brings the adoption of checklists and prescribed workflows to the development process. But importantly, it does so in the context of a development team's on-going communication and collaboration efforts.

The approach centralizes process documentation in one place and enables development teams to create and execute checklists and workflows alongside their project communication.

It is this co-location and contextual coordination that allows you to strike the balance you need between structure and free-flowing work.

Most importantly, this natural interconnection between communication and collaboration on one side and checklists and workflows on the other removes the friction that often comes with structured development approaches. Essentially, this approach gives development teams the best of both worlds offering guidance without constraint.

While merely adopting this concept of Playbooks is not a panacea and doesn't eliminate the need for project management, scrum meetings, or compliance controls, it does go a long way towards helping development team leaders strike what might otherwise be an impossible balance.

As it does, it can simultaneously help them increase productivity and improve outcomes. See? Easy.



Special Report: *Using Playbooks to Boost Productivity by Balancing the Need for Free-Flowing Collaboration and Structured Workflows*

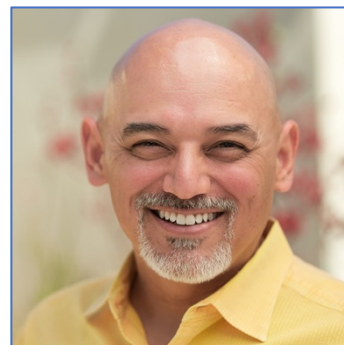
About the Author: Charles Araujo

Charles Araujo is an industry analyst, internationally recognized authority on the Digital Enterprise and author of [The Quantum Age of IT: Why Everything You Know About IT is About to Change](#).

As Principal Analyst with Intellyx, he writes, speaks and advises organizations on how to navigate through this time of disruption.

He is also the founder of The Institute for Digital Transformation and a sought after keynote speaker.

He is a regular contributor to CIO.com and has been seen in Time, InformationWeek, NetworkWorld, Computerworld, USA Today, and Forbes.



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 team communication, task and project management, and workflow orchestration together into a unified platform for agile software development.

Copyright © [Intellyx LLC](#). As of the time of writing, Mattermost is an Intellyx customer. Intellyx retains final editorial control of this paper.