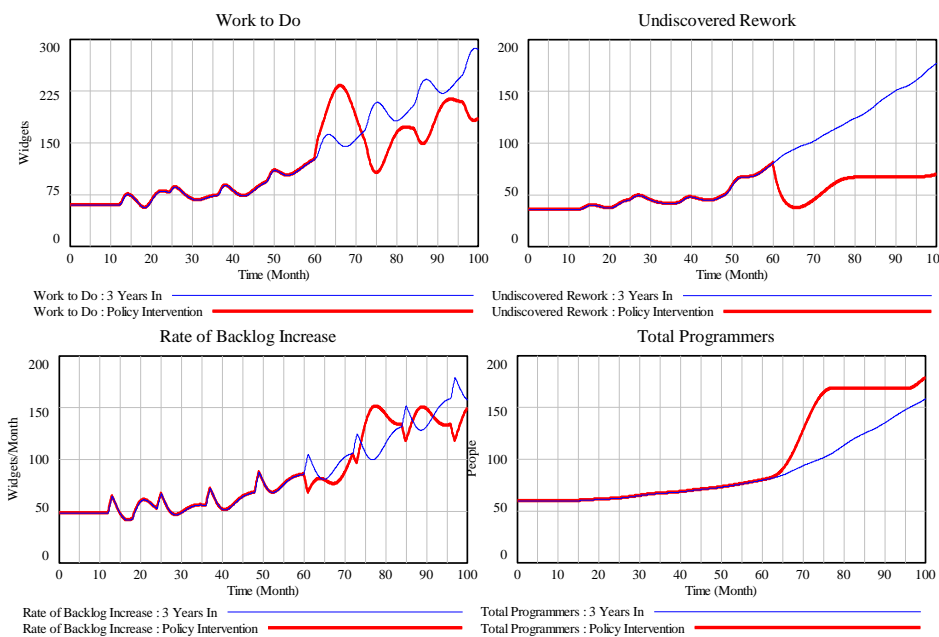




Stabilizing DevOps

Looking at the simulation charts on her monitor Amy shook her head. *Not good enough.* The combined team of her own workers and simulation scientists had forwarded initial results of recommended changes to help stabilize the environment. According to her project manager, David’s email “Three years in” was a recreation of the last five year’s her company’s ongoing transition from waterfall to Agile. Now, at Month 60, the simulation could also project what the future might hold on the current path.

But this wasn’t an idle exercise in navel gazing. The goal was to use the simulation to identify policy changes, process improvement efforts, cultural adjustments - anything that could help stabilize the environment. This first pass labeled “Policy Intervention”, didn’t look promising.



“Work to Do decreases a little bit,” Amy read aloud. “But the Rate of Backlog is hardly stable. And even though the estimate of the errors in the code is much lower... it looks like that’s because we’ve hired a lot more programmers.” *That can’t be right.*

Amy looked again over the recommendations the team had come up with. None of them were to hire more staff, which meant that the simulation itself had anticipated pressures within the system would cause management to hire more staff based on

Figure 1: Team Initial Findings

company policies and the logic of management decision-making identified during the project.

Table 1: Team Recommendations

- Slow the “pulsing” of refactoring projects to every 12 months
- Reduce backlog pressure on staff with Lean kanban
- Start Lean effort to reduce code error discovery time
- Slow time between project starts slightly to allow more completion before the next project start.
- Start a Lean process to reduce Milestone/Tollgate program effort by 50% over 18months
- Reduce interventions by Portfolio Management

Later that evening Amy texted the consulting team’s lead and her own project manager on the effort: “Pls do better.” They still had a month left on the contract to finalize the recommendations and present to senior management. *There has to be a better strategy than this!*

A week and a half later the next pass came in. Amy had passed David on the way to a meeting and received a brief update. “Turns out by starting all the interventions at the same time some overlapped and caused churn,” he said. “Oscillations I think they called it. Anyways, the simulation allowed us to run a few thousand scenarios

on the computer to isolate which ones were causing it and how far apart they had to be to avoid the effect. If we sequence the kanban deployment and project rollout changes into a second implementation window 12 months after the first the results are much better. Check your email!”

“I’ll believe it when I see it,” Amy retorted before ducking into the meeting. Now she opened and was relieved. Comparing all three runs, this time adding in “Sequenced Intervention” did show the kind of results she had hoped to see.

Even though none of the recommendations had changed, by adjusting the timing windows the results were much better. The amount of work and rate of backlog increases were both significantly lower, and more stable. The undiscovered errors in code actually stayed higher for longer, but then dropped further. Best of all, this stability wasn’t just purchased with extra staff!

“Develop an implementation plan,” she responded in email.

The next morning Amy briefed the senior steering committee. Greg who as VP had championed Tollgate program and Sarah the Director running it were not pleased to hear their program had contributed to destabilization. But they couldn’t argue the validity after agreeing throughout the effort on the internal logic used to build the model.

“It looks like the first year is worse than the baseline,” Greg retorted at her holding the charts in his hand.

“Yes.” Amy admitted. “But that’s going to be true for almost any turn around effort.”

“At least we know it’s coming,” Eric said. As CIO he had championed the transformation to DevOps and was as much on the hot-seat as Amy. “This is better than piloting for six months and then having to guess if it’s working. I’d rather test 500 pilots on a computer, learn from them and build the least-painful one. Do you have another strategy to consider Greg?” Greg shook his head, resigned to his program being cut in half.

“That settles it. Amy, you’ve begun planning for implementation correct? Great. Ask your team if the simulation can help us cut down the pain in the first year a little bit without risking later results. Sarah, you’re going to have some free resources. I want them to shift and help with communication and change. And maybe the simulation team can help us with better understanding how to be effective at managing change?”

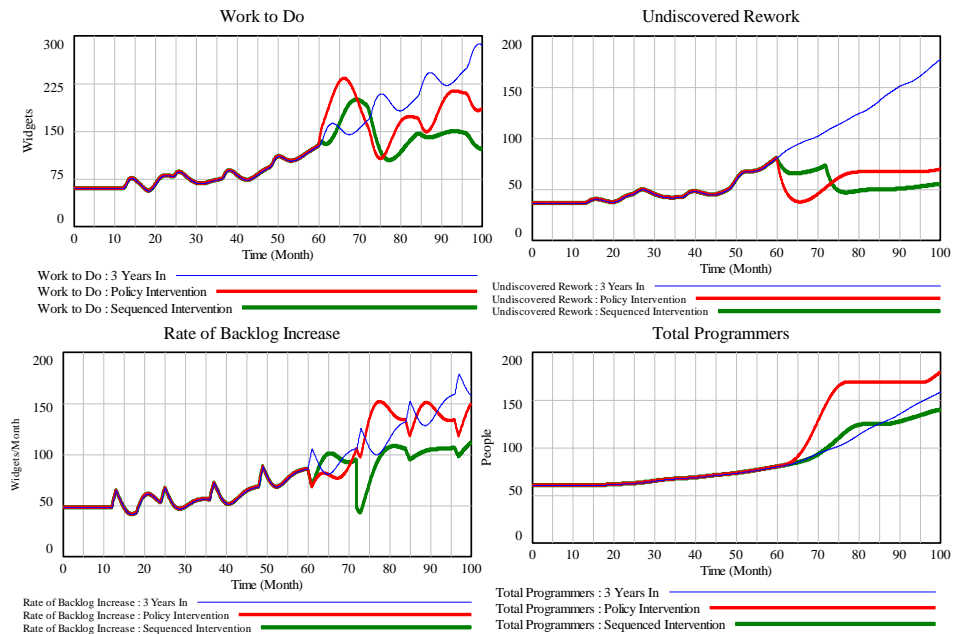


Figure 2: Team Updated Findings

*To learn more about system dynamics simulations please visit www.dialecticsims.com
The entire DevOps Stabilization story will be coming soon to www.dialecticsims.com*