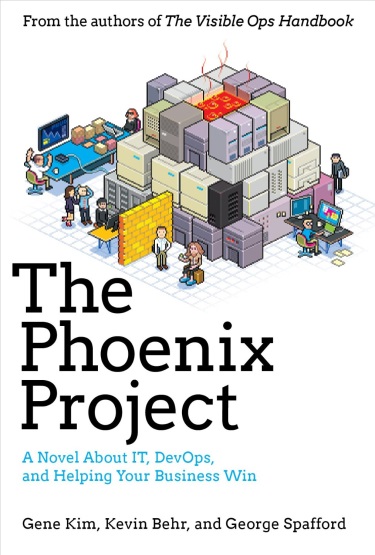
The Phoenix Project - Book Review

# Plot Summary [1]

Bill is an IT manager at Parts Unlimited. It's Tuesday morning and on his drive into the office, Bill gets a call from the CEO.

The company's new IT initiative, code named Phoenix Project, is critical to the future of Parts Unlimited, but the project is massively over budget and very late. The CEO wants Bill to report directly to him and fix the mess in ninety days or else Bill's entire department will be outsourced.

With the help of a prospective board member and his mysterious philosophy of The Three Ways, Bill starts to see that IT work has more in common with manufacturing plant work than he ever imagined. With the clock ticking, Bill must organize work flow streamline interdepartmental communications, and effectively serve the other business functions at Parts Unlimited.

# Key Concepts

The Three Ways [2]

**The First Way** emphasizes the performance of the entire system, as opposed to the performance of a specific silo of work or department. The outcomes of putting the First Way into practice include never passing a known defect to downstream work centers, never allowing local optimization to create global degradation, always seeking to increase flow, and always seeking to achieve profound understanding of the system (as per Deming).

**The Second Way** is about creating the right to left feedback loops. The goal of almost any process improvement initiative is to shorten and amplify feedback loops so necessary corrections can be continually made. The outcomes of the Second Way include understanding and responding to all customers, internal and external, shortening and amplifying all feedback loops, and embedding knowledge where we need it.

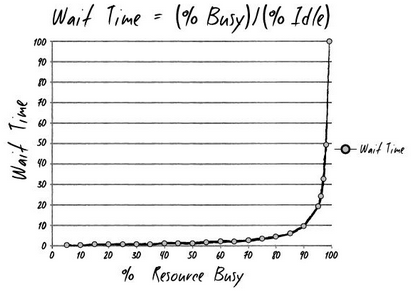
**The Third Way** is about creating a culture that fosters two things: continual experimentation, taking risks and learning from failure; and understanding that repetition and practice is the prerequisite to mastery. The outcomes of the Third Way include allocating time for the improvement of daily work, creating rituals that reward the team for taking risks, and introducing faults into the system to increase resilience.

Four Types of Work

|  |  |
| --- | --- |
| 1. Business projects 2. IT projects 3. Changes 4. Unplanned work | “What can displace planned work? Unplanned work. That’s why Erik called it the most destructive type of work. It’s not really work at all, like the others. The others are what you planned on doing, allegedly because you needed to do it.” |

Theory of Constraints

“Eliyahu M. Goldratt, who created the Theory of Constraints, showed us how any improvements made *anywhere besides the bottleneck* are an illusion. Astonishing, but true! Any improvement made after the bottleneck is useless, because it will always remain starved, waiting for work from the bottleneck. And any improvements made before the bottleneck merely results in more inventory piling up at the bottleneck.”

Resource Utilization

“The wait time is the ‘percentage of time busy’ divided by the ‘percentage of time idle.’ In other words, if a resource is fifty percent busy, then it’s fifty percent idle. The wait time is fifty percent divided by fifty percent, so one unit of time. Let’s call it one hour. So, on average, our task would wait in the queue for one hour before it gets worked. On the other hand, if a resource is ninety percent busy, the wait time is ‘ninety percent divided by ten percent’, or nine hours. In other words, our task would wait in queue nine times longer than if the resource were fifty percent idle.”

I conclude, “So, for the Phoenix task, assuming we have seven handoffs, and that each of those resources is busy ninety percent of the time, the tasks would spend in queue a total of nine hours time the seven steps…”

Change Control

* “We need to tighten up our change controls… what’s preventing us from getting there?”
* “That change management tool is impossible to use. There’s a million mandatory fields and most of the time, the drop down boxes for ‘applications affected’ don’t even have what I need. It’s why I’ve stopped even putting in change requests.”
* “I have to manually type in hundreds of server names in one of the text boxes. Most of the time, there’s not enough room in the field! A hundred server names are supposed to fit in a sixty-four-character text box? What idiot built that form?”
* “When my guys put in a change request, they have to wait a lifetime to get approvals, let alone get on the schedule. We have the business breathing down our neck to get crap done. We can’t wait for you to hem and haw, complaining that we didn’t fill out the form right.”

Lesson: Don’t let your change control process become a change prevention process. Don’t let your change control board become a bottleneck. The change control board is not the place to decide whether a change is a good idea, the role of change control board is to ensure changes have been properly coordinated with and agreed to by proper stakeholders.

# Attributions

[1] Amazon, http://www.amazon.com/Phoenix-Project-DevOps-Helping-Business/dp/0988262509/

[2] The Three Ways: The Principles Underpinning DevOps, Gene Kim, http://itrevolution.com/the-three-ways-principles-underpinning-devops/