**Assignment 1.5 – The Principles of Underpinning DevOps**

Patrick Cuauro

Bellevue University

WEB 425-307O Angular with TypeScript

Professor Krasso

August 13, 2023

# Introduction

In economics terms you generate wealth when you create products or services valued for the consumers, also you must compete against other providers which offer the same products or services as you do, the market, the geography, even the culture is part of the environment where your efforts will bring value to what you do or create. The passion you put into the efforts will make you master your craft, and your results will improve and if you position your products in market correctly, following the same principles it would be a good idea to learn from the experts the philosophy, experience, and the knowledge, right?

In the Software development industry this is the case for what we know as DevOps, the cumulative knowledge, experience, philosophy, work practices, and principles that boost the productivity and efficiency in the marketplace represent the key to become a piece of the puzzle of the competitive industry. All the information derived from DevOps is solid gold in terms of value for the modern developer taking advantage of the effort of the experts that put together the system I am about to describe next.

# The Principles of Underpinning DevOps

If you’ve read either *The Phoenix Project* or *The DevOps Handbook*, you’ve been introduced to **The Three Ways**framework for DevOps:

* The First Way: Principles of Flow
* The Second Way: Principles of Feedback
* The Third Way: Principles of Continuous Learning

## The First Way: Principles of Flow

The First Way is mostly concerned with accelerating the “flow” of work throughout a process. Gene Kim also refers to the First Way as Systems Thinking in his article The Three Ways: Principles Underpinning DevOps. Whether you’re calling it Flow or Systems Thinking, the principles underpinning the First Way are working toward the same end: viewing the flow of work as one continuous system (unsiloed) that can be continually refined and optimized.

Some of the key principles of the First Way are:

* **Making work “visible”.** Unlike manufacturing processes, which are easily observable on a plant floor, the flow of software through its development lifecycle is not easily seen. Using methods such as Kanban boards can surface the activities going on behind the scenes, by showing the left-to-right movement of a user story through the development phases.
* **Limiting work-in-progress (WIP).** Keeping work-in-progress to a minimum has also been shown to accelerate work flow, because it minimizes multi-tasking and context-switching.
* **Reducing batch sizes.** “Chunking” work into smaller pieces like a two-week sprint can also help deliver features (albeit smaller ones) and bug fixes to the customer faster. Issues are often caught earlier when those updates and additions are released sooner.
* **Reducing hand-offs between teams.** The risk of “dropping the baton” increases as the hand-offs do. Although hand-offs can’t be completely minimized, the key is to keep the teams in tight communication with one another so that the hand-off itself is almost a non-event rather than a large ordeal with the potential for communication missteps along the way.
* **Identifying and removing constraints and waste.** Constraints might be bottlenecks in the process, such as environments, test setup, and overly tight architecture, while waste includes things like manual work, heroics, and context-switching.

The Second Way: Principles of Feedback

The Second Way works to enable fast and constant feedback cycles throughout all stages of a development cycle.

Some of the key principles of the Second Way are:

* **Swarming and solving problems to build new knowledge.** This principle fits into the “fail fast” mentality, so that teams can find issues with an implementation as soon as possible and address them early and often as iterations continue.
* **Pushing quality closer to source.** This principle is at the core of the DevSecOps movement, which is concerned with addressing security concerns during the development cycle, instead of at the end, when rework to remediate is more difficult and costly.
* **Optimizing for downstream work centers.** This principle works against the “throw it over the wall” mentality, by underscoring that development should be just as invested in their application being deployable, working with operations to bridge that gap (and vice versa).

The Third Way: Principles of Continuous Learning

The Third Way seeks to create a culture of continual learning and experimentation within the development organization.

Some of the key principles of the Third Way are:

* **Enabling organizational learning and a safety culture.** Leaders must help “set the tone” for the organization, making it okay to learn, make mistakes, and try again.
* **Institutionalizing the improvement of daily work.** Improving what you do and how you accomplish it should be part of everyone’s daily thinking and call to action.
* **Transforming local discoveries to global improvements.** Surfacing and sharing improvements at all levels will help enable a “bubble up” culture of continuous improvement.
* **Injecting resilience patterns into daily work.** Some examples might include rehearsing failures and working toward improving key metrics for deployment.
* **Leaders enforcing a learning culture.** Organization-wide learning is unlikely to take hold and become pervasive unless it is sanctioned and exemplified by its leaders. So being intentional about communicating the value of learning and problem-solving is crucial to building that culture.

# Conclusion

The purpose of the mentioned principles collected from the Phoenix Project and the DevOps Handbook is to improve efficiency, productivity, and work value, in any corporation. Although it does not guarantee the elimination of issues and obstacles in the modern-day enterprise, it provides tools to make these obstacles opportunities to improve and grow to finally meet the customer expectations and position the venture in a competitive position in the marketplace using top-notch principles improving the collaboration between different departments in the organization reflecting the relation between Development and Operations, which was, at the first stages of the IT industry, the principal obstacle in the develop of solutions in short periods of time.

Sources:

* The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win by Gene Kim, Kevin Behr, & George Spafford
* The Three Ways: The Principles Underpinning DevOps by Gene Kim