



DEVOPS FOR THE ENTERPRISE: THE EVOLUTION OF PROJECT MANAGEMENT AND DEVOPS

👤 Patrick McClory 🕒 June 23, 2016 📁 DevOps 👁 6,661 Views

When looking at the broad history of modern project management styles within the IT and a software development disciplines, much of what we see are variations on two basic concepts: upfront requirements gathering (Waterfall) and iterative or continual requirements gathering (Agile) methodologies. While the evolution of these processes over time has been beneficial, both Waterfall and Agile methodologies lack a critical perspective—understanding and leveraging the environment where applications actually live.

When Waterfall methodologies were formalized, they refined the process of developing software and added required nuances that differentiated the process of managing these projects from other non-technical work. Waterfall has the benefit of managing risk and ensuring that requirements are well composed, though teams working under this methodology are less able to be responsive to the changing needs of the business. Waterfall also adds a significant amount of time from the inception of a project to delivery of a working system due to the need to gather a full set of requirements before starting development work.

As an extension of Waterfall, Agile frameworks were developed for technical project management. Agile methodologies (like XP, pair programming, Scrum, etc.) aim to lower the barrier of entry for projects by focusing on iterative design and requirements development, along with iterative delivery of smaller, more focused pieces of software that, in the end, add up to the sum total of the intended system. This methodology also aims to be more responsive to the changing needs of the business and overtly recognizes that some requirements tend not to be articulated or understood until the system is already under construction. In practice, Agile has been well accepted within software development teams, but the process of releasing applications, changes, or updates to production environments still follow a gated, waterfall-like process in transitioning to a more operations-focused IT team who would then handle the release process. This misalignment in process and management style often leads to further organizational confusion and tension between teams operating under different project management styles.

DevOps, in turn, builds on Agile as a means of managing change in an iterative, feedback-focused, and responsive process. It does so by feature velocity (the ability to deliver functional software quickly) and a focus on delivering business value as highly prioritized functional requirements of the overall process and system. To achieve this, the most common practice is to automate the vast majority of the overall development and delivery process from the developer's desk to the production environment. A cloud-first strategy can assist in addressing objections to infrastructure availability issues (procurement, CAPEX expenditures, approvals, etc.) while also reinforcing the need to embrace automated configuration management over manual setup and management of systems. DevOps further aims to unify the team around both the development and the operational effort, often resulting in developers taking on-call time while operations staff works to more deeply understand the application stack. This commonly leads to insightful performance improvements and new non-functional requirements that improve the overall operational health of the application and drives a better end user and operational experience overall.

To simplify it, Agile is the process used to setup adaptive software development for fast feedback cycles and DevOps adds the cultural aspect of concentrating on business results and focuses on improving the collaboration between developer and long-term operational teams. Both are crucial to the effective management of project flow, but you can see the natural evolution that brought us to ever-increasing trend of DevOps. If we do not address the people, processes, and tools, as well as the technology needed during production and long-term management, we cannot expect to realize the benefits of faster delivery of functionality achieved through Agile. It is only when modern project management methodologies come together with DevOps that a technology organization can see optimized results.

☺