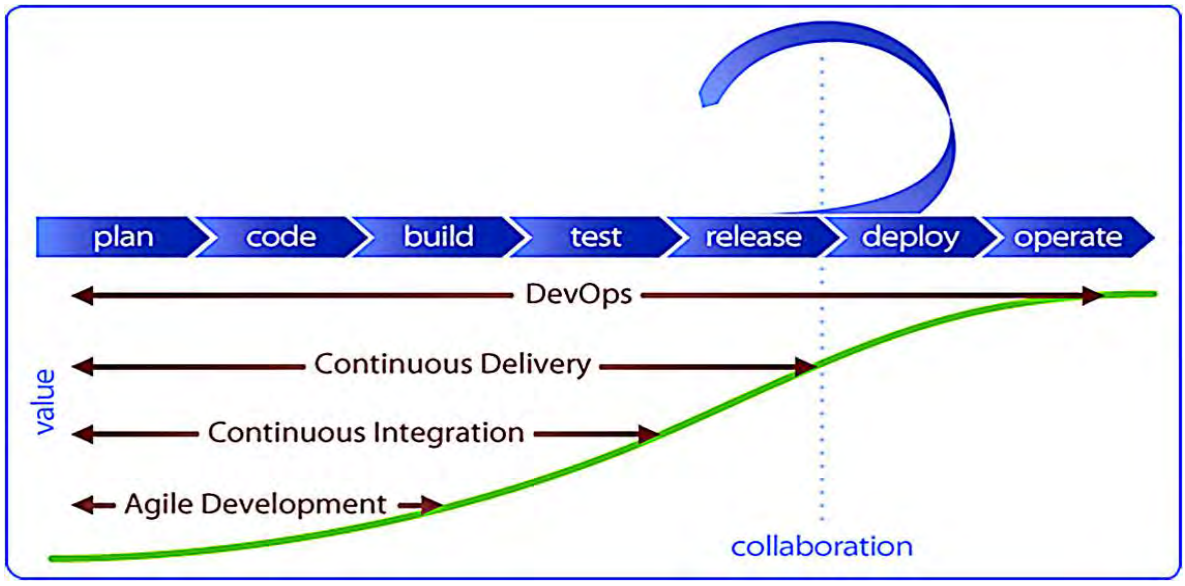
Business Value, ROI, and Cost of Quality (CoQ) for DevOps

Motivation. Nearly two-thirds or 4.5 billion of the world's 7.5 billion inhabitants are now interconnected by the World Wide Web (WWW) or Internet of Things (IOT) in the form of personal computers, laptops, tablets, smartphones, mobile devices, and other household appliances. Many of these devices are based upon a relatively small or common suite of hardware and software components such as operating systems, middleware, and applications. Long gone is the era when suppliers of hardware and software designed, built, and operated these components for a few dozens, hundred, or thousand people at the most. Today's products must be built and maintained in real-time for hundreds of millions or billions of global end-users simultaneously, 24 hours a day, 365 days a year.

Problem. Basically, the size and scale of the global landscape, industry, and marketplace for building and maintaining information technology (IT) products and services has dramatically increased by multiple orders-of-magnitude in just a few short years or decades. A typical IT company in the 1980s or 1990s may have 10,000 to 30,000 IT-related workers to handle the workload using traditional manually-intensive labor, paperwork, processes, and brick-n-mortar (physical) facilities. Using the same crude 20th century instruments for conducting IT business in the 21st century would require millions of workers for each company to handle the same workload along with the associated overhead costs (i.e., billions, perhaps trillions of dollars in operating expenses). However, it's a zero-sum game, because there are only about 25 to 75 million qualified IT-related workers in the entire world today who can service the entire population of IT-related product and service end-users. In other words, yesterday's means of conducting IT business doesn't scale and no longer applies to the modern era and today's IT firms are now faced with the challenge of delivering vastly more products and services with infinitesimally less resources, quite literally. Those firms who can cost-effectively provide high-quality products and services to the global marketplace on Internet time (speed of light) can survive, compete, and flourish (i.e., Google, Yahoo, Amazon, EBay, Microsoft, etc.). Conversely, IT firms entrenched in crude labor and manually-intensive practices of 20th century manufacturing era operating principles simply cannot keep up with the pace of global technological change, will go out of business and fail, and rapidly exit the market without a trace (likewise at the speed-of-light). The new currency is speed and successful 21st century firms must be able to provide IT products and services to billions of worldwide customers in fractions of a second using modern, contemporary, and often technologically-enhanced operating principles and technologies that are a significant departure anything anyone could ever dream or imagine in the 20th century (sort of like the bionically-augmented version of 20th century firms).



Approach. A host of new business models, management paradigms, operating principles, and technologies have emerged to enable both public and private sector organizations, firms, and other enterprises to successfully compete in the hyper-competitive global landscape of the 21st century in fractions of second. These include Lean and Agile business models at the organizational and management levels and Continuous Integration, Continuous Delivery, and DevOps at the technical engineering levels. Lean and Agile methods enable business leaders and managers to think, plan, and act quickly, accurately, and effectively, while the latter three principles enable technical personnel to lift enormously-large and heavy 21st century globally-scaled workloads in fractions of a second where the level of attention to detail explodes and expands exponentially like an uncontrolled chain reaction. Their keys are to dramatically downsize and limit the number of humans, organizational layers, governance structures, decision-making gates, labor and manually-intensive processes and paperwork, and the brick-n-mortar infrastructure itself (physical buildings). Down at the lower technical levels where the cost is the greatest, the key is to eliminate, substitute, and automate the entire back-end of the supply chain as much as possible, if not entirely. Thus, the solution to successfully competing in today's marketplace begins to emerge. Continuous Integration enables IT personnel to automate the process of testing and integrating millions and billions of product and service components in fractions of a second and pennies on the dollar. Continuous Delivery further automates the process of assembling and packaging a final high-quality product and service that's ready for prime time.

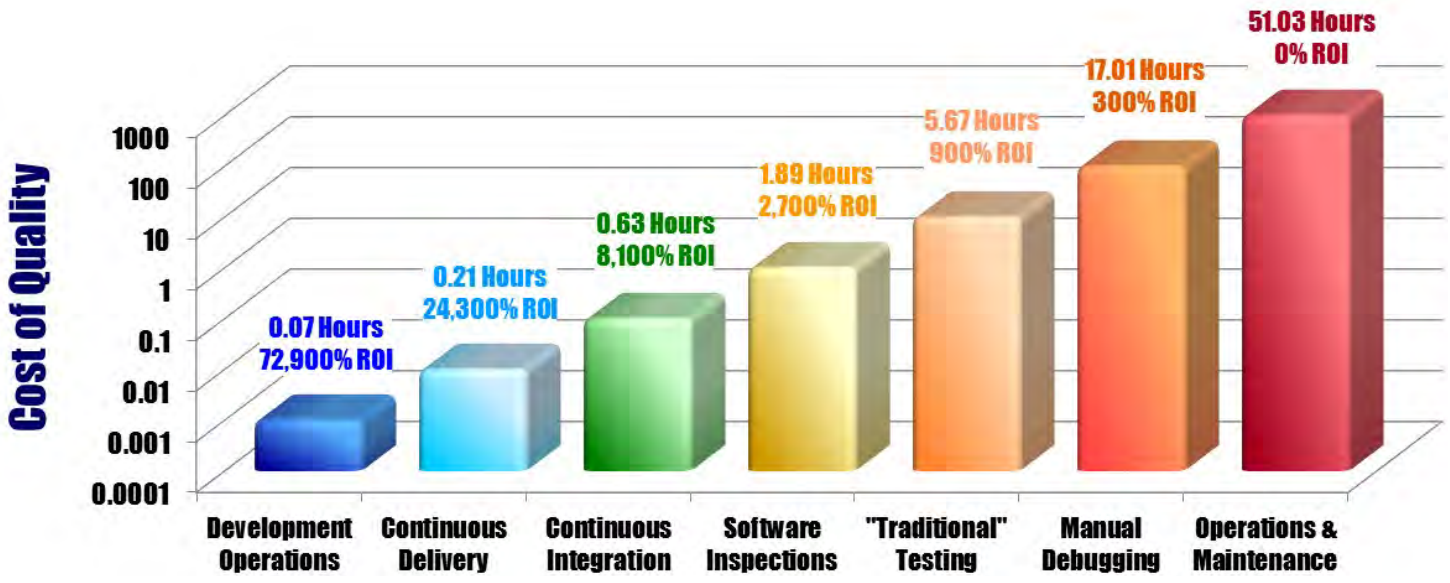
Finally, DevOps automates the final, cost and labor-intensive step of delivering these products and services to billions of global end-users in fractions of a second, transparently recalling them if necessary without penalty or cost, and ever so gradually refining, optimizing, and repairing them in real-time without interrupting the operation or daily lives of global end-users. That is, think of DevOps as a giant shipping and returns department all rolled into a single massive Star Trek-style transporter room zapping trillions of bits back and forth across the universe at the speed of light with ultra-high levels of reliability and safety for each organization that chooses to employ this 21st century paradigm. Likewise, imagine if you could change your tire while your automobile is screaming down the interstate or highway at 50 or 60 miles per hour without changing lanes, slowing down, stopping, or otherwise endangering your car's occupants. That's exactly what all of these business, management, and technical operating principles and technologies enable today's IT businesses to do. It's important to note that today's technologies themselves are key, essential, critical, and irreplaceable components that enable Lean, Agile, Continuous Integration, Continuous Delivery, and DevOps to succeed. These technologies include, but are not limited to public clouds, data centers, massively parallel-processing high-performance computer clusters, satellite communications, global networks, Linux operating systems, open source software-based middleware, modern programming paradigms such as Java, Perl, Ruby, etc., and the end-user devices themselves such as personal computers, laptops, tablets, smart phones, mobile devices, and a myriad of other Internet-enabled appliances. This fails to mention the semiconductors themselves, such as dirt-cheap microprocessors, memories, solid-state storage devices, and other so-called software-defined functions. The secret sauce for at least the last four or five decades has been the software, which is one of the most flexible and malleable substances ever created by humans. Software is the perfect medium for the Lean and Agile paradigms, and over 90% of complex systems functions are performed in software. Hardware technologies become obsolete before system prototypes can be constructed and the key to global success in the 21st century is to compete on software, while divesting capital-intensive hardware investments altogether. The ultimate goal is to seamlessly interconnect the entire supply chain from customer to supplier and then back to customer, automating away the traditional army of people constituting the 20th century organizational hierarchy resulting in the needless transaction costs identified by Ronald Coase in the 1930s.

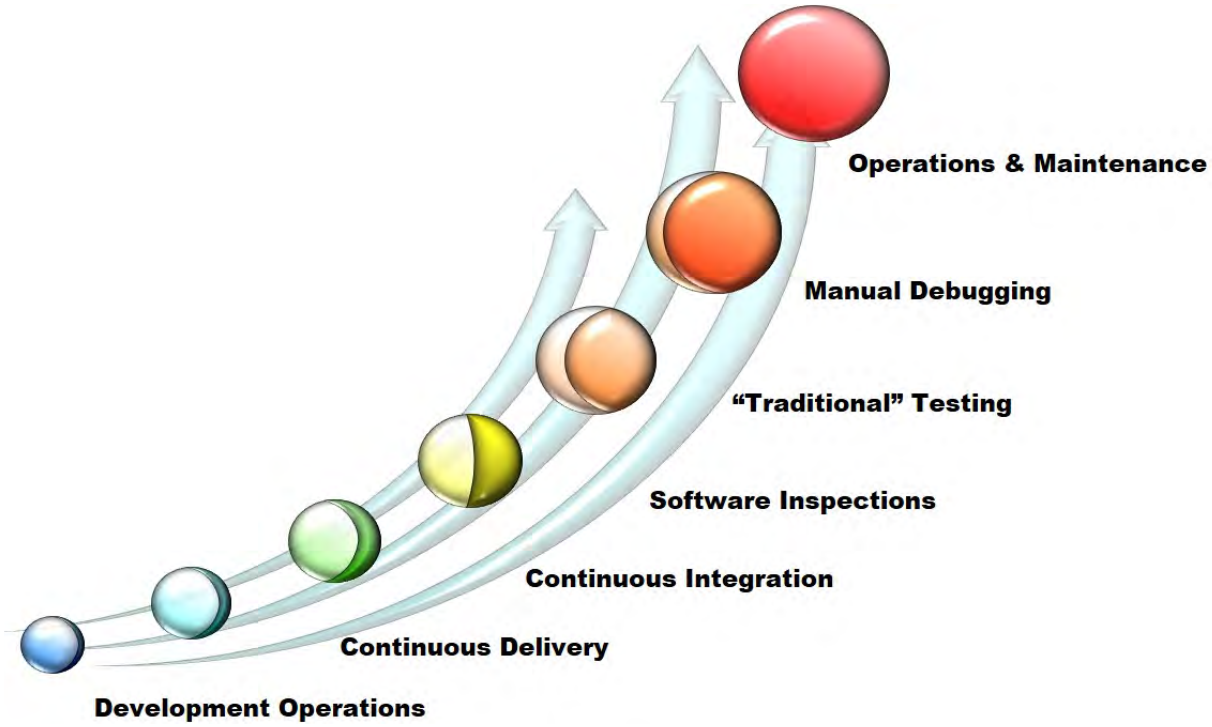
Paradigm	Practice	Description
Continuous Integration	**Building**	Frequently assembling products and services to ensure delivery readiness
	Database	Frequently generating/analyzing database schemas, queries, and forms
	Inspections	Frequently performing automated static analysis of product/service quality
	Testing	Frequently performing automated dynamic product and service evaluation
	Feedback	Frequently generating automated status reports/messages for all stakeholders
	Documentation	Frequently performing automated technical/customer document generation
	Deployment	Frequently performing automated delivery of products/services to pre-production
Continuous Delivery	**Packaging**	Frequently generating system images for pre-production testing and checkout
	Acceptance Test	Frequently performing automated system and user acceptance testing
	Load Test	Frequently performing automated system load, stress, and capacity testing
	Performance Test	Frequently performing automated system user and technical performance testing
	Pre-Production Test	Frequently performing automated pre-production tests prior to final deployment
	Certification	Frequently performing automated system certification and accreditation tests
	Deployment	Frequently generating product images for pre-deployment testing and checkout
Development Operations	**Systems Administration**	Frequently performing automated system administration tasks such as scripting
	Config. Management	Frequently performing automated infrastructure config. Management/version control
	Host Builds	Frequently performing automated system and server host builds and configuration
	Virtualization	Frequently performing automated system, server, and network virtualization services
	Containerization	Frequently performing automated software and Microservices containerization
	Deployment	Frequently generating final end-user system and software images for distribution
	Monitoring & Support	Frequently performing automated metrics collection and deployment monitoring

RESULTS. Using classical systems and software engineering life cycle-based quality management models and paradigms such as phase-based defect removal models, we demonstrate that Lean, Agile, Continuous Integration, Continuous Delivery, and DevOps yield previously unprecedented levels of ultra-high quality and reliability. Since the 1970s, systems and software engineering researchers used defect removal models, and even more sophisticated statistical models such as Weibull-based Rayleigh life cycle reliability equations, to model the effects of investments in upfront quality engineering activities. These activities often included causal analysis activities such as defect prevention, static analysis activities such as systems and software inspections, and dynamic analysis activities such as systems and software testing. However, the crux of these paradigms was their dependence upon humans to perform error-prone, manual and labor intensive activities. These so-called disciplined activities were often viewed as the necessary tools to build complex 20th century technology-intensive products, services, and systems. Elaborate models, paradigms, and operating principles were constructed such as project management, systems engineering, software engineering, quality and reliability engineering, and a myriad of process engineering-based maturity models. They were called disciplined, because humans needed the automaton like rigidity to repeatedly perform manual processes, construct detailed document suites, and execute a cacophony of hierarchical decision-making governance functions. When scaled to large and complex systems, tens of thousands of engineers were required to create thousands of documents over decades at a cost of billions and trillions of dollars for a single delivery of a technology-intensive product or service. Laborious processes and documents were viewed as the keys to scalability, when quite the opposite was true. The more complex the system, along with

the requisite processes, documents, and armies of project participants actually slowed the project down, dramatically decreased the system quality rather than increasing it, and reduced the likelihood of project success to nearly zero. These were often called zero-defect paradigms, but should have been called zero-success or certain failure paradigms. The net effects of these 20th century manufacturing practices were prophetically characterized in 1975 by Frederick Brooks who said, “adding more people to a late project makes it later.” It was also commonly known in the 1970s that adding more people to projects increased communication paths exponentially, slowing progress to a dead halt. In spite of these early predictions, no one listened, and project managers, systems engineers, and software engineers continued to create, promote, and ruthlessly apply even more process and document-intensive paradigms well into the early 21st century. Today, we understand the economics of these failed initiatives in terms of queuing theory that emerged in the early 1900s. That is, adding more items (complexity) to a queue increases wait times (duration), while decreasing quality and reliability. Not only that, but technology and information decay exponentially and become obsolete faster than they can be documented in print or even in graphical models such as blue prints, schematics, directed graphs, and other so-called system and software models (due to entropy). Therefore, despite our most visionary mathematicians, systems and software engineers have been killing their complex projects, products, and services dead with ever increasing systems and software engineering discipline by stuffing their queues full of managers, engineers, processes, documents, governance boards, meetings, reviews, check points, decision points, stage gates, audits, inspections, certifications, and a myriad of other process minutia. Even the most sophisticated defect removal models relied upon humans to perform thousands of inspections every year with machine like accuracy and without error to yield exactly one defect per investment hour for complex systems over long periods of time, which was not even humanly possible. Conversely, by automating the phases (with technology) using Continuous Integration, Continuous Delivery, and DevOps, then the defect removal model becomes an accurate lens to portray the true cost of quality (CoQ), defect levels, and system reliability. That is, if each phase is automated using inexpensive and often free technology, then millions and billions of simultaneous static and dynamic analysis tests, integrations, builds, and deliveries can be made in fractions of a second at the speed of light without the certainty of human failure. Even when modeled as a simple seven-stage defect removal model, Continuous Integration, Continuous Delivery, and DevOps render a three or four order of magnitude reduction in staffing size, CoQ, defects, system failures, duration (wait times), and project failure. Thus, these latter three paradigms can be combined to serve as a modern operating paradigm to enable 21st century firms to conduct trillions of transactions for billions of global end-users in fractions of a second at pennies on the dollar. Key principles also include the application of Lean and Agile principles, public clouds and data centers to eliminate the debilitating effects of private investments in capital infrastructure, and a host of interrelated concepts such as a Minimal Marketable Feature (MMF), Minimal Viable Product (MVP), Story Map, and even Microservices. These latter four paradigms are used to dramatically shrink the scope of the project, product, service, solution, and ultimately queue size to a small, manageable, and inconsequential size. Thus, all of these paradigms converge to flip the CoQ upside down by decreasing the queue wait times from infinity to nearly zero. Continuous Integration, Continuous Delivery, DevOps, and Microservices can truly be referred to as zero CoQ paradigms.

Activity	Def	CoQ	DevOps Economics	Hours	ROI
Development Operations	100	0.001	100 Defects x 70% Efficiency x 0.001 Hours	0.070	72,900%
Continuous Delivery	30	0.01	30 Defects x 70% Efficiency x 0.01 Hours	0.210	24,300%
Continuous Integration	9	0.1	9 Defects x 70% Efficiency x 0.1 Hours	0.630	8,100%
Software Inspections	3	1	2.7 Defects x 70% Efficiency x 1 Hours	1.890	2,700%
"Traditional" Testing	0.81	10	0.81 Defects x 70% Efficiency x 10 Hours	5.670	900%
Manual Debugging	0.243	100	0.243 Defects x 70% Efficiency x 100 Hours	17.010	300%
Operations & Maintenance	0.073	1,000	0.0729 Defects x 70% Efficiency x 1,000 Hours	51.030	n/a





CONCLUSION. The global landscape constituting the 21st century has changed dramatically. There have been numerous fundamental shifts in the marketplace that changed the operating rules and principles for organizations and businesses over the decades and centuries. These are too numerous to mention here. Some of the notable ones were the advent of the electronic computer in 1945, emergence of commercial semiconductors and the Internet in the 1960s, legalization of the commercial software industry in 1969, beginning of the PC era in the 1970s, explosion of the WWW in the 1990s, etc. Each of these technologies were extinction-level dinosaur-killing events, signaling the end of global manufacturing for post-industrial nations, along with manufacturing era operating principles like traditional project management, systems and software engineering, and quality management practices. More importantly, they signaled the end of traditional capital-intensive brick-n-mortar organizations and the armies of humans it took to operate them, along with labor and manual-intensive governance boards, processes, documents, and other trappings of the physical world. Software became the new medium of exchange in the form of network protocols, operating systems, middleware, databases, and application software. None were more profound than the dominance of Open Source Software, Microservices, key enablers to outsourcing capital infrastructure like public clouds and data centers, and software-defined networks, communications, radios, etc., that were once dominated by hard-wired logic and embedded systems. Public clouds, social media, and mobile computing became the tools of nearly all of the world's 7.5 billion inhabitants. A small handful of organizations emerged from the nuclear winter of global organizational change that were well-suited to develop, deliver, operate, and maintain products and services for the software-intensive electronic landscape. These included Google, Amazon, Yahoo, Facebook, Twitter, Snapchat, Ebay, and even Microsoft, Apple, Verizon, AT&T, Sprint, T-Mobile, and a few other global telecoms. These 21st century Internet giants were faced with serving billions of customers on a 24 x 7 basis. They had to design, develop, deliver, operate, and maintain billions and sometimes trillions of Microservices at the speed of light. The paradigms for doing so included Lean and Agile methods, Continuous Integration, Continuous Delivery, and Development Operations or DevOps. Ultra-high levels of quality, reliability, and availability became center stage, along with security, privacy, usability, and user experience design. Although skeptical at first, Lean, Agile, Continuous Integration, Continuous Delivery, and DevOps easily rose to the challenge of satisfying each of these non-functional requirements for billions of global end-users at a near zero CoQ.

FURTHER READING.

- Rico, D. F. (2014). *18 reasons why agile cost of quality (CoQ) is a fraction of traditional methods*. Retrieved June 25, 2014, from http://davidfrico.com/agile-vs-trad-coq.pdf
- Rico, D. F. (2014). *Best practices*: *Kickstarting agile methods in a traditional organization*. Retrieved June 16, 2014, from http://davidfrico.com/kickstarting-agility.pdf
- Rico, D. F. (2012). *The cost of quality (CoQ) for agile vs. traditional project management*. Fairfax, VA: Gantthead.Com
- Rico, D. F. (2011). *The business value of agile project management for new products and services*. Fairfax, VA: Gantthead.Com.
- Rico, D. F., Sayani, H. H., & Sone, S. (2009). *The business value of agile software methods*: *Maximizing ROI with right-sized processes and documentation*. Ft. Lauderdale, FL: J. Ross Publishing.
- Rico, D. F. (2008). What is the ROI of agile vs. traditional methods? An analysis of extreme programming, test-driven development, pair programming, and scrum (using real options). *TickIT International, 10*(4), 9-18.
- Rico, D. F. (2004). *ROI of software process improvement*: *Metrics for project managers and software engineers*. Boca Raton, FL: J. Ross Publishing.