The Cost of Quality (CoQ) for Agile vs. Traditional Project Management

by Dr. David F. Rico, PMP, ACP, CSM

Code inspections are viewed as one of the most effective means of quality control among IT developers. They usually involve some sort of manual or static review of software source code before testing. They are an efficient means of finding software defects and result in high-quality software-based products and services. Today, they are a best-practice among traditional software developers and explicitly appear as a formal phase or activity in the waterfall life cycle.

However, code inspections are an implicit, often unspoken best practice among agile project management teams. This silence has caused some people to question the quality control of the agile project management paradigm. Surprisingly, agile teams have not forgotten to mind the Ps and Qs of quality engineering, and not only continue to perform code inspections, but perform them more often. This results in even greater quality than traditional project management teams.

Code inspections are an effective means of quality control created by Michael E. Fagan of IBM in 1972. They are highly-structured meetings in which a small group of peers are tasked with identifying as many defects as possible by reviewing software source code. Computer program statements are analyzed one-at-a-time to identify as many defects before testing as possible. The use of dynamic debugging and testing alone is an inefficient means of finding and fixing defects.

Code inspections are structured meetings with a set of formal rules. They are facilitated by trained moderators to enforce meeting protocols. They include distinct time-boxed phases for planning, education, preparation, code review, defect repair, and follow-up. They have a singular purpose, i.e., identify as many defects as possible before testing. Technical peers review code created by programmers. Managers and other inessential personnel do not participate.

Today, code inspections are simply referred to as "*peer reviews*." They are not just for software source code any more. They have been expanded to include all life-cycles phases and products, i.e., project plans, requirements, architectures, designs, tests, documentation, etc. Fiercely resisted in the early days, code inspections have taken nearly 40 years to be commonly accepted as a best computer programming practice. Furthermore, they are based upon several theories

- *Computer programmers are blind to errors or defects in their own source code.*
- *An unbiased second set of eyes is needed to objectively identify code defects.*
- *A group or team of technical peers can identify more defects than individuals acting alone.*
- *Code inspections must have rules and facilitators to maintain efficiency and effectiveness.*
- *Code inspections can be used to quickly and efficiently identify defects before testing begins.*
- *Testing without code inspections is an inefficient and ineffective means of quality control.*
- *Zero defects results by combining code inspections with other quality engineering activities.*

Code inspection cost of quality (CoQ) is quite impressive. It takes an hour to repair defects by code inspections, 10 hours by testing, and 100 hours by debugging. Most processes are only 70% efficient on average and defects escape from one stage to the next. Even then, code inspections have an ROI of 300% during development and 900% over the total life cycle. Although it costs less to use them than to ignore them, they are considered essential for mission critical situations.

Activity	CoQ	Economics of Code Inspections	Hours	ROI
Code Inspections	1	100 Defects x 70% Efficiency x 1 Hours	70	**n/a**
Testing	10	30 Defects x 70% Efficiency x 10 Hours	210	**300%**
Debugging	100	9 Defects x 70% Efficiency x 100 Hours	630	**900%**

Code inspections do have their detractors. Some feel they are too expensive to perform. Others say customers are not willing to pay for them. Most believe quality is good-enough without them. Some feel defects are not a good measure of product quality, reliability, or customer satisfaction. Still others assert they are no more efficient than individual code reviews (i.e., desk checking). They are not easy to perform and few have mastered the art of doing them well.

Agile project management was created to develop high-quality products and services. It is primarily intended for the development of complex new products and services like software. It is intended to be a better approach than traditional paradigms created in the last 40 years. It is not only designed to be more efficient than its predecessors, but more effective as well. Some of the quality and reliability engineering theories underlying agile project management include:

- *High quality requirements are obtained from intense customer involvement.*
- *Smaller processes, documents, and timelines reduce costs and increase quality.*
- *Automated inspections, tests, and reporting decreases costs and increases quality.*
- *Discipline, quality, and reliability are attainable with lightweight, flexible processes.*
- *Faster, automated testing shortens customer feedback cycles leading to higher quality.*
- *High quality and reliability is achievable in a matter of hours, days, weeks, and months.*
- *Lightweight processes increase sustainability, morale, customer satisfaction, and quality.*

Customer satisfaction and product quality are at the heart of agile project management. These basic tenets are often contrary to the belief of traditional developers. Its practices combine to form a synergistic whole that results in higher quality than traditional methods. While not an exhaustive list, some of the basic practices include developing high-quality requirements, tests, designs, etc. Only validated deliveries with high business value are provided to customers.

Practice	Description
Release planning	Develop a small set of high-quality customer requirements
Test-driven development	Develop test cases for every product or service unit
Architecture spikes	Create rapid prototypes to reduce risks and explore valid solutions
Pair programming	Create every product or service unit using two or more developers
Continuous integration	Automatically regression test every unit against the system frequently
Iterative development	Deliver validated units to customers every few days or weeks
Retrospectives	Frequently conduct causal analysis to continuously improve quality

Agile project management practices improve productivity, teamwork, and customer satisfaction. However, continuous integration simultaneously increases quality and reduces costs by an order of magnitude. Some believe it’s an inefficient process of debugging rather than inspection or test. Instead, it is a complex combination of automated quality engineering practices. It includes static and dynamic quality analysis, along with automated reporting, documentation, and delivery.

Practice	Description
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 end users

Continuous integration is a key practice within agile project management. Setup time is less than a day's worth of effort and the cost of a personal computer. Most automated inspection and test tools are free and open source software. It requires a fraction of project effort, it costs 10 times less than code inspections, and it yields 10 times more defects. Therefore, its ROI is at least 3 times more than code inspections, 9 times more than testing, and 27 times more than debugging.

Activity	CoQ	Economics of Continuous Integration	Hours	ROI
Continuous Integration	0.1	100 Defects x 70% Efficiency x 0.1 Hours	7	**n/a**
Code Inspections	1	30 Defects x 70% Efficiency x 1 Hours	21	**300%**
Testing	10	9 Defects x 70% Efficiency x 10 Hours	63	**900%**
Debugging	100	2.7 Defects x 70% Efficiency x 100 Hours	189	**2,700%**

Agile teams perform quality practices more often than traditional teams. They perform numerous automated static and dynamic tests. Dozens of static, unit, component, integration, system, and other special tests are automatically initiated every 10 minutes. A traditional team only performs about 30 to 80 code inspections per year. However, agile teams perform 12,480 to 149,760 or more automated static and dynamic tests per year, based on their level of process proficiency.

Tests	1 Hour	1 Day	1 Week	1 Month	3 Months	6 Months	1 Year
One	6	48	240	1,040	3,120	6,240	**12,480**
Three	18	144	720	3,120	9,360	18,720	**37,440**
Six	36	288	1,440	6,240	18,720	37,440	**74,880**
Twelve	72	576	2,880	12,480	37,440	74,880	**149,760**

It now becomes a little clearer how agile teams produce high quality products and services. They perform far more tests than traditional teams at a fraction of the cost of manual, labor-intensive techniques. They are also more efficient, effective, and consistent. Organization change is easier, because it is less human-intensive and involves less resistance-to-change. However, some teams are remiss to let go of traditional testing, which is late, manual, less frequent, and less effective.

Traditional project management rests upon firm requirements, rigid processes, long schedules, and manual testing. Conversely, agile project management consists of flexible requirements, lightweight processes, fast delivery, and robust automation. It represents a fundamental paradigm shift that causes traditional teams to question its discipline, quality, and reliability. However, its use has proven to increase product and service quality by 10 times for a fraction of the cost.

Agile project management is used by over 80% of world-wide IT projects. This includes U.S. DoD, Fortune 500, financial, global telecommunications, and Silicon Valley firms such as Google, Facebook, Yahoo, Amazon, Microsoft, etc. It is a highly-disciplined, flexible, and lightweight alternative to traditional scope-driven project management. Finally, agile project management is clearly a "*10X solution*" for today's enterprise and mission critical needs.

Dr. Rico has been a leader in support of major U.S. gov't agencies for 25 years. He's led many Cloud, Lean, Agile, SOA, Web Services, Six Sigma, FOSS, ISO 9001, CMMI, and SW-CMM projects. He specializes in IT investment analysis, portfolio valuation, and organizational change. He's been an international keynote speaker, presented at leading conferences, written six textbooks, and published numerous articles. He's also a frequent PMI, APLN, INCOSE, SPIN, and conference speaker (http://davidfrico.com).