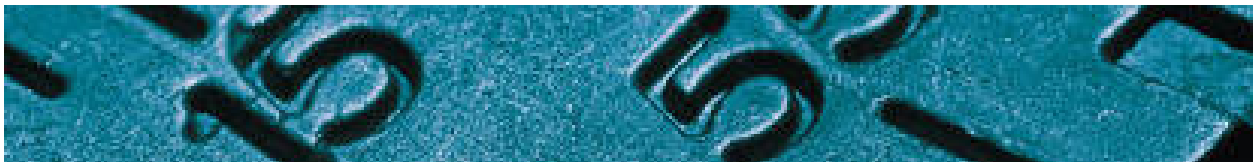

Quality Focus – what can be learned from the Toyota recall?

During their February 2010 earnings call, Toyota estimated it will have to spend $1.12 billion on warranty expenses and will lose as much as $895 million in lost sales over the recall. On top of that, it is bracing for a U.S. Congressional hearing about massive vehicle recalls [i]

This news should be used as a wakeup call to remind all of us how important it is to consider quality at all phases of the lifecycle. It highlights what quality issues can cost a company and what it can do to a firm's reputation; just look how quickly a reputation built on quality over decades can be diminished.

In the software development business you are only as good as your last release. In addition to executing against a strong quality assurance process, managing perception is critical to ensuring customer satisfaction and maintaining and generating new business. You can have ten solid production releases and people will only remember that your last release needed an emergency release or hot-fix to deal with a software defect and that cost the firm N number of dollars in profit and or lost revenue.

Quality is what differentiates good software delivery teams from great delivery teams.

Traditionally, in project management the triple constraint is listed as:

1. scope,
2. schedule and
3. cost.

However one of the major changes in the *PMBOK® Guide* – Fourth Edition is it no longer mentions the triple constraint of scope, schedule and cost. Instead it discusses how project managers must balance the constraints of scope, quality, schedule, budget, resources and risk[ii]. Quality is the key item.

Whether you are the Project Manager, Product Manager, or the Quality Assurance Lead there are five key dimensions to consider with regards to quality:

1. communications,
2. metrics,
3. automation,
4. process and
5. project management.

1. Communications

Build a strong communications bridge between all teams, but specifically the product development and quality control and/or assurance team. Have the quality control and/or

assurance team spend time understanding the development team's processes, for example their source code management processes. This can be an area of high risk and exposure if not done well and monitored closely[iii].

A real life example is any multi-release situation where you have parallel software branches going at once (e.g., delivering multi-state quote and bind functionality in the insurance industry). By understanding why, when and how the team handles software merges and branches, the quality team can better plan and control their regression tests and look for patterns in defect management.

Another key consideration is to ensure that all team have a sound communication management plan in place. Consider this formula for communications channels: N (N-1)/2, where N equals the number of stakeholders. Hence a project with 10 stakeholders could have 45 potential communications channels[iv].

2. **Metrics**

Mathematic assets, methods and capabilities help to develop predictive analytics and business optimization to create solutions for the challenges we face everyday.

Gather the information through consistent and rigorous processes. Software projects have a lot of great data that can be used to greatly enhance the quality of the deliverables.

Take advantage of the data you capture and use it to benefit your success. Predictions are not perfect but they allow you to set goals and provide meaningful metrics.

Metrics also allow for you to provide full transparency and take perception, emotions, etcetera out of the equation and focus on the facts.

3. **Automation**

Effective use of automation tools can reduce cost, increase availability of resources, maintain and in most cases improve quality.

For example[v] at a US P&C insurer, working on a B2C (Business to Consumer) development project, they typically build and deploy software to the test environment three times, each week.

The development team is required to quickly perform build acceptance tests (BAT) within a 2 to 3 hour window - from build to deployment. It requires testing the application for all the geographic states included in the release, in two environments.

When the team is in the new development phase a bundle can consist of 7 to 8 states. This requires a group of 4 to 5 testers being available to execute the task.
4 testers x 2 hours of test execution x 3 times a week = 24 person hours. By using an automated testing tool, the team performed the same task with 1 tester.

1 tester X 3 hours of test execution X 3 times a week = 9 person hours. Note the team initially invested 120 hours into the creation of the base, but that investment will pay for itself.

4. **Process**

Enforcing consistent and rigorous quality processes during requirements definition can also greatly increase the quality of a project's deliverables. As auto manufacturers are experiencing now, defects found early in the project lifecycle are much less expensive than finding the same defect in a later phase of the project[vi].

For example, a defect found in requirements costs $2 to fix versus the same defect costing $200 to fix in the operation phase.

Project Managers need to introduce rigorous frameworks to ensure deliverables are being created at the highest level of quality possible within schedule and budget.

One example of a framework is the "Inspection" process, developed by former IBM'er Michael Fagan. This framework, used by companies from Intuit to BAE Systems[vii], is an excellent and proven example of how introducing rigor into requirements definition can maximize quality while significantly reducing long term project costs.

Similarly to how the Japanese originally viewed quality, the Inspection process sees every defect as a treasure! Using defined roles and responsibilities during the inspection process, objective checklists are used to thoroughly examine deliverables looking for defects.

Defect logs are used to gather and track data for further analysis, which further stresses the importance of strong metrics to improve quality. In short, the inspection process allows software projects to create their own "Andon[1]" cord to pull and discover defects earlier rather than later when they become much more expensive.

5. **Project Management**

Strong Project Management fundaments never go out of style and will keep you on track. The world we live in is getting more and more complex by the second and challenging. According to IBM, 49% of software-related projects suffer budget overruns and 50% of outsourced projects are expected to under-perform. Having good Project Management fundaments will help to set you up for success.

One key area that I see over looked often is the lack of a strong requirements management and traceability system, even for non-functional items like security. You can have a great product that does everything exactly to specification, but then your CEO is interviewed about how the average cost for a comprised personal record in US companies is $204[viii], because testing the security items were overlooked.

The take away for delivery teams is whether it is a gas pedal on a car, misspelled or mistyped content or a faulty insurance premium calculation causing a customer to be overcharged quality is not only a step in the process or something you should feel like is a necessary evil; it should really be part of a team's DNA and they should always challenges themselves to not have a Toyota moment, **because it can happen to any of us**.

[1] The andon cord empowered workers to stop the entire production line if any complications arose, thus adding a human check to jidoka. Referenced from http://www2.toyota.co.jp/en/vision/traditions/jul_aug_04.html

About the Authors

Luke Kelleher is a Project Executive and Senior Managing Consultant with IBM's Global Business Services Insurance practice. His specialties include complex systems integration, project management, data conversion and object-based project methodologies. He can be reached at luke.kelleher@us.ibm.com.

David Lipien is a Senior Managing Consultant with IBM's Global Business Services Insurance practice. His specialties include complex systems integration, release management, internet-based technologies, wireless technologies and object-based project methodologies. David is a PMI Certified Project Management Professional and Microsoft Certified Professional (MCP), and holds a Bachelor of Science and a Master of Business Administration. He has also completed an Executive Management Program at the Harvard Business School. He can be reached at lipien@us.ibm.com.

Seth Newell is a Senior Consultant with IBM's Global Business Services Insurance practice. His specialties include software package implementations and process improvement. He can be reached at srnewell@us.ibm.com.

Justin Ryan is a Managing Consultant with IBM's Global Business Services CRM practice. His specialties include software package implementations, project management, and custom application software development. He can be reached at justryan@us.ibm.com.

[i] Business Week. (2010). Businessweek.com. Retrieved February 10, 2010, from http://www.businessweek.com

[ii] PM Forum. (2010). pmforum.org. Retrieved February 11, 2010, from http://www.pmforum.org

[iv] A Guide to the Project Management Body of Knowledge – Third Edition (2004).

[v] Automated Testing for Insurance Applications Using Rational Functional Tester, S. Purakan and D. Lipien, IBM Presentation (2009)

[vi] Boehm. B (1981). Software Engineering Economics. New York: Prentice Hall.

[vii] Michael Fagan and Associates. (2010). Mfagan.com. Retrieved February 20, 2010. http://www.mfagan.com/clientsay_frame.html

[viii] Source: Ponemon Institute, Traverse City, MI - (2010)