PS&J SOFTWARE SIX SIGMA

Measured Managed and Controlled Project Performance

Contact us at: 201-947-0150; 201-358-8828; Quality@SoftwareSixSigma.com

Software Six Sigma

Software Six Sigma is an overall strategy to accelerate and to sustain continuous improvements in the software development process and in software product quality. It is based on the application of statistical tools to the process and product metrics that characterize a stable software development process. Software Six Sigma enables quantitative management of product quality. This in turn:

- speeds up integration and test
- allows delivery of very high quality product (very few latent defects)
- improves the repeatability and predictability of the entire software development process.

Software Six Sigma involves the application of Statistical Process Control (SPC) and related techniques to the production of software requirements, architecture, modules, and test cases. These statistical techniques are known collectively as the six sigma tool kit.

Software Six Sigma's attributes are:

- trustworthy metric data
- accurate planning based on historical data
- use of statistical tools for real time analysis and decision support
- quantifiable Software Process Improvement cost and benefits
- predictably high product quality

Software Six Sigma enables more effective program management. Six Sigma techniques provide metrics to make timely "fact based" management decisions. The quality of the project tracking data is so high that problems can frequently be anticipated well in advance of their occurrence and can often be mitigated or avoided entirely.

Many organizations have had significant success deploying six sigma in manufacturing, but have had very little or no success with software engineering. Some barriers to successful application of six sigma to software are:

- Standard six sigma training does not directly relate to software development
- Software is different and software developers can't bridge gap on their own
- One size fits all generates resistance
- Too many competing initiatives - Total Quality Management, ISO 9000, Capability Maturity Model, Six Sigma
- Poor alignment of sponsorship - Divided responsibilities for process improvement and product development

- Business as usual after training - No time for continuous improvement
- Six Sigma Projects are not considered integral to software development

Software development is indeed different,

- Process variation can never be eliminated or even reduced below a moderate level No two modules are alike so process performance always includes an intrinsic degree of variability
- There are very large differences in skills and experience from one developer to another
- Specifications are not based around tolerances Systems don't fail because they are assembled from many loosely toleranced components. A single well-placed defect in a low level component can be catastrophic

But software development is also measurable and controllable. Software development processes can be fully characterized by three simple measurements

- Time: the time required to perform a task
- Size: the size of the work product produced
- Defects: the number and type of defects, removal time, point of injection and point of removal

Software measurements are amenable to statistical control provided:

- Data is complete, consistent, and accurate
- Data from individuals with widely varying skill levels is not mixed

One of the great strengths of the Six Sigma approach is its focus on business goals. Improvements are evaluated based on their impact on business goals, not maturity levels. Typical goals are to achieve lower development and maintenance costs, faster development cycles, and higher product quality.

Productivity and cycle time cannot be managed directly. In a sense they are outputs of the process used to develop the software. They are controlled by two other variables, product quality and time on task. The quality of the intermediate software products directly controls the amount of time spent in integration and test. This is typically 30% – 50% of development cost. Quality can be managed directly using statistical techniques yielding products that move through test very quickly and have very few latent defects remaining when deployed. Improving quality management practices can readily yield 15% - 25% improvements in productivity and cycle time.

Time on task is defined as the amount of time spent producing product. Data has repeatedly shown that most organizations only get 30% - 50% time on task during a work week. Time on task is another area that lends itself to rapid improvement when pro-actively managed with statistical techniques. We have frequently seen improvements of 50% within 6 months.

As product quality is improved and more defects are found earlier in the life cycle, the rate of defect removal becomes much more predictable. When review becomes the primary mechanism to find and remove defects, the removal rate is fairly constant and predictable. For more, see our presentation on optimizing inspections.

This has a significant impact on the ability to schedule tasks. It is even possible to predict the number of defects in a new module with a fair degree of accuracy, allowing the number of defects found already to serve as an indicator of the overall quality of a module.

Project plans include highly granular task lists, that when used in conjunction with earned value techniques, enable very accurate project tracking. Accurate tracking in turn enables early recognition of and response to problems. This greatly enhances the probability of meeting project commitments. In practice it becomes possible to consistently estimate precedented activities to an accuracy of 20% or better and to produce estimates to go in the 5% range.

Introduction of Six Sigma techniques into a software development organization usually requires changes in both the software development process and in the project management process. Six Sigma works best with incremental and evolutionary product development lifecycles. A project model based around a waterfall process does not support a philosophy of rapid continuous improvement very well.

Unless the organization has a software process in place, it is necessary to introduce one. The process must reflect what the developers actually do, not the desires of management. Six Sigma techniques work best with lightweight, incremental processes that have been instrumented to produce high quality metric data. Processes like Team Software Process (TSP) and Extreme Programming can both make good platforms for Six Sigma. It can also work well with heavier weight processes like RUP provided care is taken to avoid a document-centric approach. Whatever the process platform is, it needs to include a basic software measurement framework.

There is no need for an organization to be at a high CMM level before it can embrace the style of results oriented, measurement driven continuous improvement exemplified by Six Sigma. It fact it works well at any CMM level, and be used to accelerate the progress of an immature organization to a higher maturity level.

Software Six Sigma with PS&J

PS&J Software Six Sigma specializes in the application of six sigma techniques to software development. We offer a unique perspective on software process improvement. No other organization offers the same levels of experience that we do. No other organization offers a comparable integrated approach to the application of six sigma techniques to software. Our principals have been working with the application of six sigma to software development since 1995 and were responsible for defining the approach and software specific training program used at AlliedSignal (now Honeywell).

Most six-sigma programs fail to address the differences between software engineering and manufacturing in their training programs. This makes it very difficult for software engineers to apply six sigma techniques in their everyday activities. Frequently courses draw all their examples from services or operations and they are taught by instructors with little or no background in software engineering.

This kind of training is not likely to succeed in moving Six Sigma concepts into software development. In fact the usual result is convincing the software developers that it doesn't apply.

While this is not true, there are many differences between applying Six Sigma to software and applying it in a service or operations environment. These differences have to be explicitly acknowledged in the training program. Moreover the elements of the six-sigma tool kit have different relative levels of importance for software applications than for more conventional applications. This requires changing the emphasis on and the amount of time spent with each tool kit element during training.

The model for continuous improvement in software six sigma also changes substantially the emphasis shift away from Six Sigma projects led by Black Belts to continuous improvement in personal practices by a development staff that has internalized six sigma techniques at a green belt level. This means the ratio of the number of black belts to green belts is substantially lower for a software application of six sigma than for a typical operations or services application.

PS&J Software Six Sigma has developed a training program and a continuous improvement model designed to address these issues with a specialized courses demonstrating the application of six sigma techniques to every aspect of software development.

If your organization already has a six sigma training program, our courses can be integrated into your curriculum to provide green belt training for software developers and electives for your software black belts. If not, we can help you put a software specific six sigma program in place with everything you need from training, to automated process data collection and analysis, to green belt and black belt certification.

Our Software Six Sigma Green Belt course provide a solid introduction to applying six sigma to software appropriate for software engineers who will employ the six sigma toolkit in everyday practice. As it stands the course is self-contained. If the participants have already had some six-sigma training, we can tailor it to avoid any duplication. It can also be tailored to provide a core course for a black belt that wants to specialize in software development.

Introduction to Software Six Sigma is suitable for people with a pre-existing six sigma background that need to understand how to apply six sigma concepts to software development at a high level. Along with Managing & Moderating Inspections, Software Defect Prevention, and Six Sigma Software Project Management, it can be used as a software elective in an existing Black Belt curriculum.

As always, PS&J Software Six Sigma follows up our training with mentoring and consulting to help you institutionalize new practices.