Introducing Six Sigma to Software Development

PMI NYC Chapter

Wednesday, March 17, 2004

Steve Janiszewski

PS&J Software Six Sigma

Why Apply Six Sigma to SPI?

- In order to meet business needs, one cannot simply try harder. One must significantly change the developers' daily activities
  - involves a level of risk that many organizations are unwilling to accept
- With conventional SPI, it is easy to fall into the trap of laying a veneer of process over the same old activities
  - flows from a desire to hit CMM level goals while causing as little disruption to daily routine as possible
  - often adds overhead while resulting in no significant improvements
  - can destroying credibility of SPI initiative with the developers
- Exploit corporate Six Sigma sponsorship to boost your SPI initiative
- Accelerate progress to higher CMM levels

One definition of insanity: doing the same thing over and over and expecting a different result

Why Adopt Six Sigma in Software Development?

- Six Sigma increases the likelihood of success
  - providing visible linkage to business goals makes sustainable executive sponsorship more likely
  - emphasis on measurement makes significant changes in organization behavior more likely
- Achieve better process performance for the same investment
  - Initial estimates typically accurate to better than 20%
  - Estimates to go typically good to under 10%
  - Productivity up 30% - 50%
  - Product Quality better by at least a factor of 4

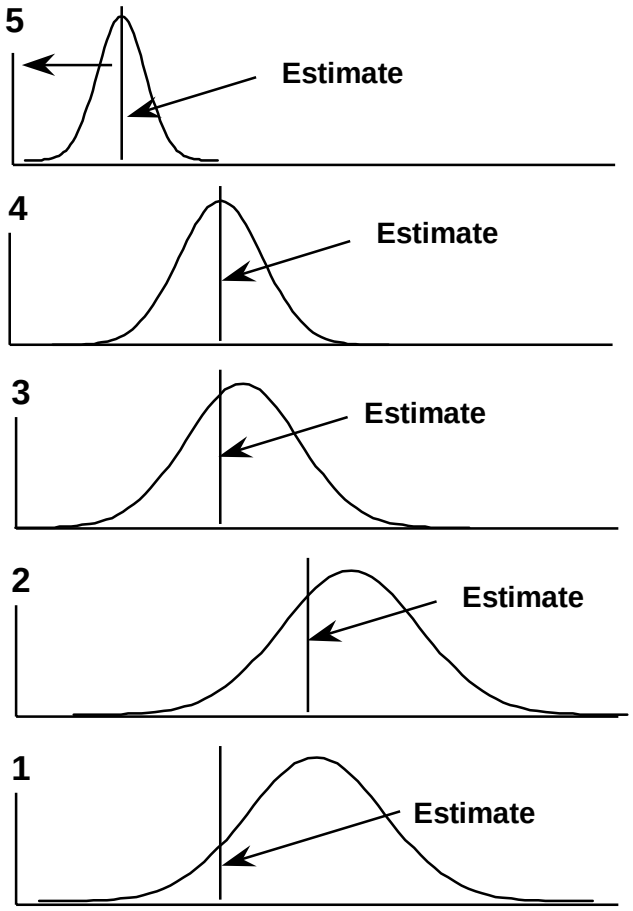
Common Misconceptions

- Many organizations put off getting involved with six sigma until they are CMM level 3
  - Don’t realize that Six Sigma as a continuous improvement methodology is applicable to any process element at any CMM level
- Some organizations have heard about good experiences with Six Sigma, but know that software development is not like manufacturing. So they assume that Six Sigma is not applicable because “software is different”
- Some organizations attempt to provide the same Six Sigma training to everyone resulting in sending software engineers to training courses appropriate for manufacturing or transaction processing
  - Most cited reason is need for common language on multi-disciplinary teams
  - Can generate resistance and stall adoption in engineering disciplines like software development

Acknowledge the Differences

- The first step to getting the software development community to accept six sigma is to acknowledge the things that make software different
- Build on existing SPI initiatives like CMM
  - Avoid the tendency to re-invent the wheel
  - Acknowledge the validity of the CMM approach
    - Seek to complement it, not supercede it
- Understand the technical differences between applying the six sigma tool kit to software development and applying it to manufacturing or transaction processing before you start training
  - Adapt your approach to six sigma to address the differences and modify training accordingly
  - Deal with motivational issues in training
- Remove key barriers before you start
  - Put a software measurement framework in place before starting six sigma training
  - Deal with data privacy issues at the management policy level and build credibility
- Adopt a bottoms up approach to continuous improvement where black belts frequently operate as consultants to individual developers attempting to understand their own data

CMM Levels – A Six Sigma Perspective



SCHEDULE/COST/QUALITY

- From a business perspective, predictable process performance is a key aspect of process capability
  - Predictable performance requires a stable process
  - First step to a stable process is a “defined process”
- Moving up the CMM levels requires
  - first stabilizing the overall process,
  - centering on estimated performance,
  - reducing variation,
  - continuously improving the process by improving centering & variation
- The same cycle can be applied to any low level process at any CMM level

Software is 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 & experience from one developer to another that cause dramatic differences in process performance**
- **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**
  - **Concept of quadratic loss function has less applicability because the true goal is no "serious" defects**

Software is different!

- Rolled Throughput Yield is not useful a concept for software development
- Measurement System Evaluation (MSE) has limited applicability when a good measurement framework is in place
- It is frequently very hard to perform designed experiments, so regression analysis of available data is relative more common than DOE
- Early defect removal is just as important as defect prevention
  - Certain classes of defects can be prevented
  - The application of Six Sigma to software development emphasizes defect containment & early removal as well as prevention

Software Processes Are Different Too!

- Individual low level software processes are generally much simpler than manufacturing processes
  - They have fewer steps - typically under 10
  - They are dependent on far fewer variables
    - There are no material factors
    - Environmental factors like temperature & humidity are generally not important
- Low level software processes are pretty much the same across the industry
  - They are not proprietary like manufacturing processes
  - They are frequently published with extensive supporting performance data
  - They are largely independent of the specific module under development

But software is measurable & controllable!

- Software development processes can be fully characterized by just three simple measurements time, size, and defects
- Statistical analysis techniques can be applied to software measurements provided:
  - Data is complete, consistent, and accurate
  - Data from individuals with widely varying skill levels is not mixed
- Metrics need to be put into a statistical context before being used to make decisions
- Software process performance can be managed using statistical process control

Six Sigma is applicable and has the potential for dramatic performance improvements

Contact Information

Ellen George **201- 358-8828**
EllenGeorge@SoftwareSixSigma.com

Steve Janiszewski **201- 947-0150**
SteveJaniszewski@SoftwareSixSigma.com

www.SoftwareSixSigma.com