Six Sigma & Software Process Improvement

ICSPI

International Conference on Software Process Improvement

University of Maryland Conference Center

Tuesday, November 19, 2002

Steve Janiszewski

Ellen George

PS&J Software Six Sigma

Six Sigma Starts with Business Results

- Six Sigma is a metrics driven approach to continuous improvement that starts with quantitative business goals providing direct value to the customer
- Data analysis is used to identify specific processes with the greatest leverage on these goals
- Critical inputs affecting process performance are identified
- Goals are related to changes in process outputs
- Improvements are implemented on a pilot basis
- If measurements indicate goals have been achieved, improvements are institutionalized
- Process performance is controlled to the new levels by controlling critical inputs

What are you going to tell your new boss when she asks you to quantify the return on your SPI activities?

DMAIC

The Six Sigma Continuous Improvement Cycle

- **Define the process**
- **Measure the process**
- **Analyze the process to identify causal variables**
- **Improve the process**
  - **Modify the process**
  - **Measure the modified process**
  - **Verify the improvement**
  - **Define control mechanism**
- **Control the process to new performance levels**
  - **Monitor performance metrics and take designated action when required**
  - **Perform continuous verification of the stability and capability of the process**

A Control System Viewpoint

- Process outputs, y, are a function, f, of a set of control variables, x, and process noise ε:
  - $y = f(x) + \varepsilon$
  - The y's are not directly controllable, but they can be controlled by controlling the x's.
  - Statistical measurements are necessary to avoid re-acting to the noise ε
- For a software project, y's include cost and schedule and x's include product quality and time on task.
- A software process should like a responsive, "closed loop" control system driving the x's to planned values and through their relationship to the y's, achieving overall product goals

Closed loop processes mean the difference between tracking an effort and managing an effort

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
- 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

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

“What’s in a name?”

- The phrase “Six Sigma” is frequently used with three different but related meanings
- In the narrowest sense, Six Sigma is used as a measurement of product quality
  - A Six Sigma quality level means that products have less than 3.4 defects per million opportunities, i.e. the product is 99.9997% error-free
- By extension, a process capable of producing products at Six Sigma quality levels is referred to as a Six Sigma Process
  - typical software processes operate at between 2.3 and 3.0 sigma
  - the best software processes operate at 4 - 5 sigma although they exhibit all the characteristics of a typical 6 sigma process
- In the broadest sense Six Sigma is
  - the application of DMAIC as a continuous improvement method,
  - in conjunction with a more or less standard toolkit of statistical analysis methods,
  - with the object of producing and managing Six Sigma processes

Driving CMM Based SPI With Six Sigma

- Six Sigma methodology can be used to drive CMM based SPI in a bottoms-up fashion
- Emphasis on direct coupling to business results and measurable improvements
  - allows easy quantification SPI ROI
  - moves organization away from level oriented goals – levels become a by-product of SPI, not the primary goal of SPI
  - sustains executive sponsorship
- More likely to result in measurable improvements than top down process deployment driven by level goals
  - Apply DMAIC to one or two processes at a time as part of an SPI action plan
  - Use process metrics to access success in achieving business goals in order to quantify process effectiveness
  - Track SPI ROI
- Objective measurements are required to successfully manage a process - a process that is not managed is unlikely to perform well

Six Sigma Software Process Characteristics

- Accurate project planning based on historical data and accurate project tracking that enables timely and effective corrective actions by management
- Application of statistical tools to trustworthy process and product metrics to support real time decision making using quantitative phase exit criteria
- Quantitative management of product quality
  - allowing delivery of very high quality product (very few latent defects)
  - reducing time spent in integration and test thereby cutting overall cost and cycle time
  - making the software development process more repeatable and predictable
- Closed loop process management and improvement
- Quantifiable SPI benefits

Sounds a lot like CMM Level 5

Some Common Misconceptions

- Many organizations put off getting involved with six sigma until they are CMM level 3
  - A Six Sigma software process is basically a level 5 process so they wait until they are ready to move to level 4
  - Don't realize that Six Sigma as a continuous improvement methodology is applicable to any process element at any CMM level
  - Miss opportunity to make their CMM effort more likely to succeed and to achieve measurable business results
- Other organizations have heard about good experiences with Six Sigma in operations or services, but know that software development is not like manufacturing. So they assume that Six Sigma is not applicable
  - While software development is indeed very different from manufacturing, and Six Sigma training aimed at a manufacturing environment is not likely to be successful in a software organization, the basic Six Sigma principles and methods apply to software quite well

Software is different!

- Software development is not like manufacturing, it is not even like providing a service
- 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
  - Concept of quadratic loss function has very limited applicability
- Measurement System Evaluation (MSE) has limited applicability
- Rolled Throughput Yield is not as useful a concept for software development as it is for manufacturing

But software is 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
- Statistical analysis techniques can be applied t software measurements provided:
  - Data is complete, consistent, and accurate
  - Data from individuals with widely varying skill levels is not mixed
- Software process performance can be managed using statistical process control

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

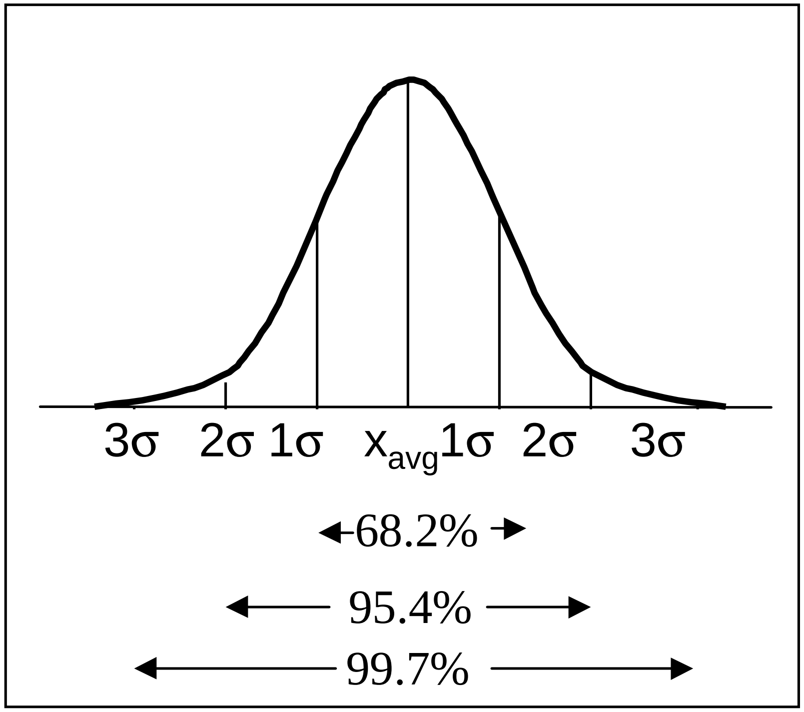
Applying the Six Sigma Toolkit to Software

- The majority of elements of the Six Sigma toolkit are directly applicable to every day software development data analysis
  - Quality Function Deployment (QFD) for prioritizing requirements
  - Process mapping for work flow optimization
  - Correlation Analysis
  - Analysis of Variance (ANOVA)
  - Failure Modes Effect Analysis (FMEA)
  - Statistical Process Control
  - Control Plans
- Design of Experiments (DOE), Measurement System Evaluation (MSE) and LEAN tend to have less applicability to every day software development situations than they do in manufacturing applications

Variation

- Most data tends to follow the normal distribution or bell curve.
- The standard deviation (σ) measures variation present in the data

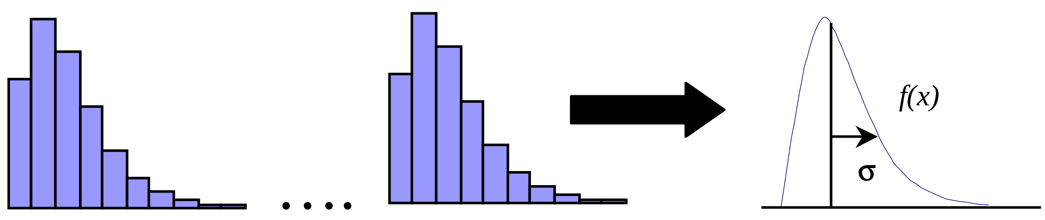
$$\sigma = \sqrt{\frac{1}{n-1}\sum (x - x_{avg})^2}$$



- For data that follows a normal distribution
  – 99.99999975% of the data is within ± 6σ
- ±3σ is natural limit of random data variation produced by a process
- The empirical rule allows us to treat non-normal data as if it were normal for the purposes of statistical process control
  – 60%-75% of the data is within 1σ of the mean
  – 90%-98% of the data is within 2σ of the mean
  – 99%-100% of the data is within 3σ of mean

Statistical Control

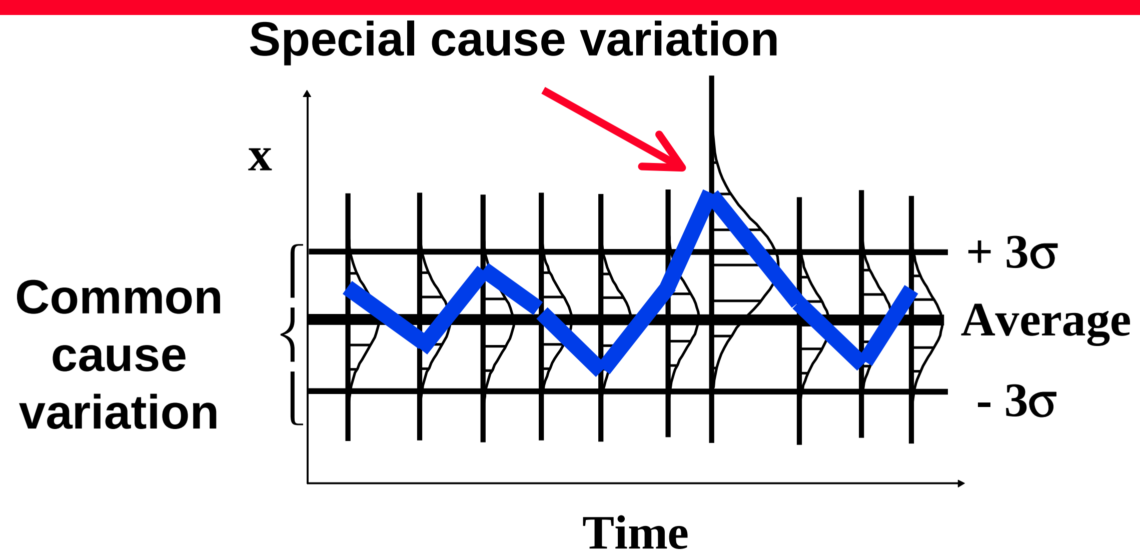
- A process exhibits statistical control when a sequence of measurements $x_1, x_2, x_3,\ldots x_n,\ldots$ has a consistent and predictable amount of variation
- It is possible to model this pattern of variation with a stationary probability density function f(x)



- Can make statistically valid predictions about processes that exhibits statistical control
- When the process does not exhibit statistical control, the distribution function changes over time, destroying the ability to make statistically valid predictions
- A stable well-defined process is a pre-requisite for statistical control

Control Charts

- Control charts are a graphical depiction of the normal range of variation of a stable process



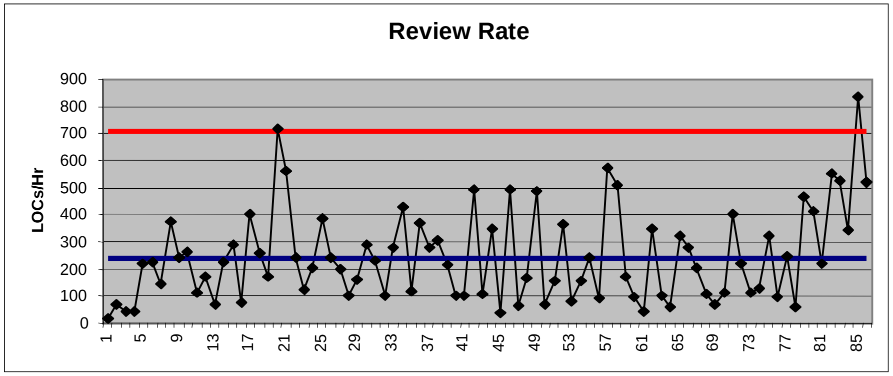
- Common cause variation is normal random variation in process performance
  - Don't over-react to common cause variation
  - Reduction requires a process change
- Special cause variation represents an exception to the process
  - Actions to correct special cause variation must eliminate a specific assignable cause
  - Special cause action eliminates a specific isolated event; does not necessarily involve a process change
- Don't take special cause action to deal with common cause problem

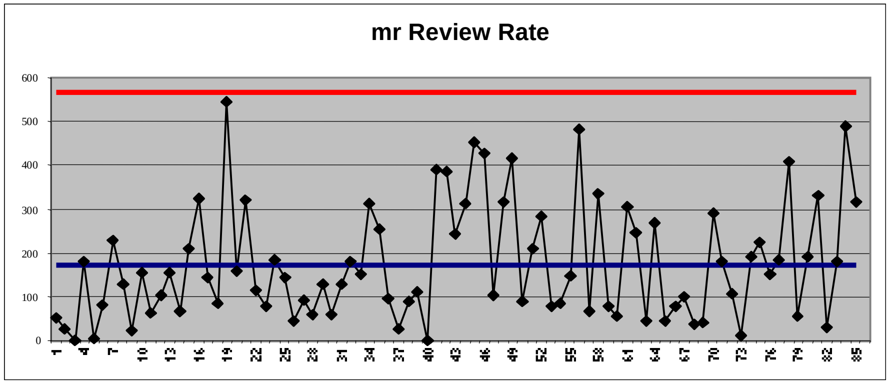
XmR Charts

- X is time series of variables x_i, $i = 1,2,...$
- R is time series for range of X from measurement to measurement, i.e. $r_i = |x_i - x_{i-1}|$

- x_{avg} is average value of X
- mR is average value of R

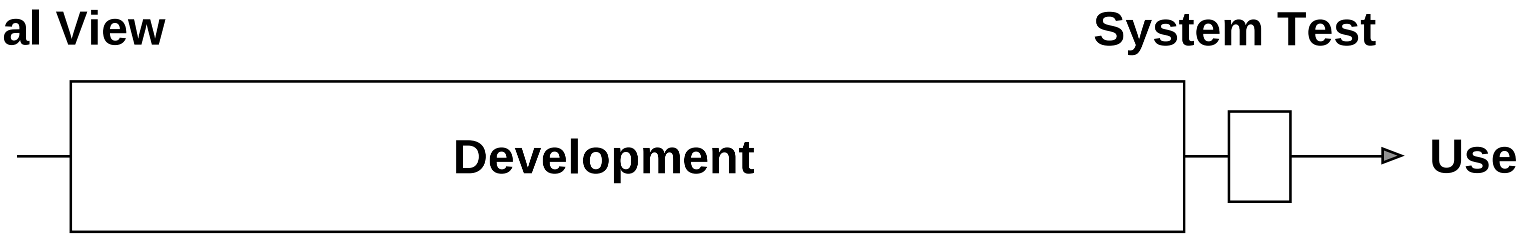
- X chart shows
  - x_i vs. time
  - X_{avg}
  - $CL_x = X_{avg} \pm 2.660$ mR
- R chart shows
  - r_i vs time
  - mR
  - $CL_r = 3.628$ mR





Hidden Factory and Yield

Traditional View



"The Hidden Factory"

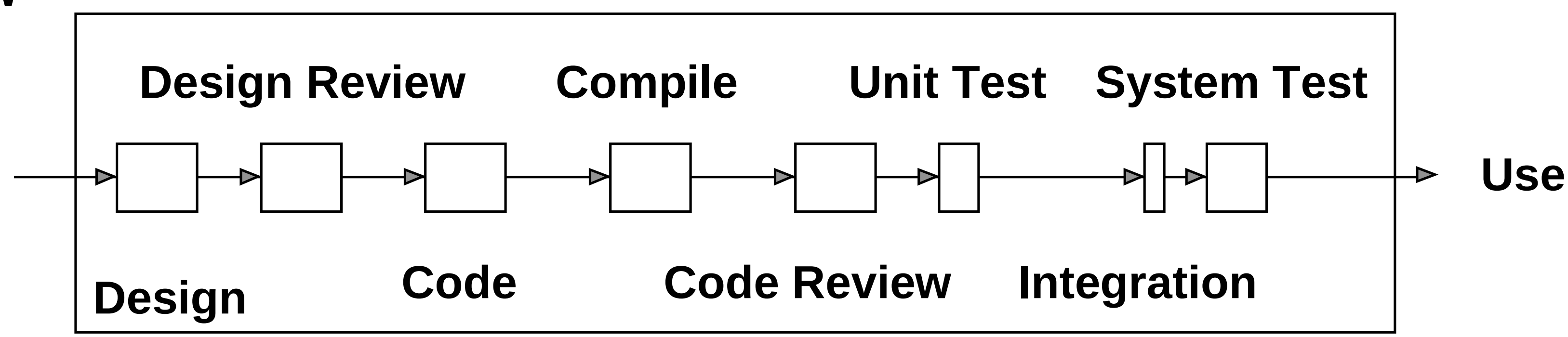
Defects are not recorded prior to system test

Yield = n_{system}/(n_{system} + $n_{escapes}$).

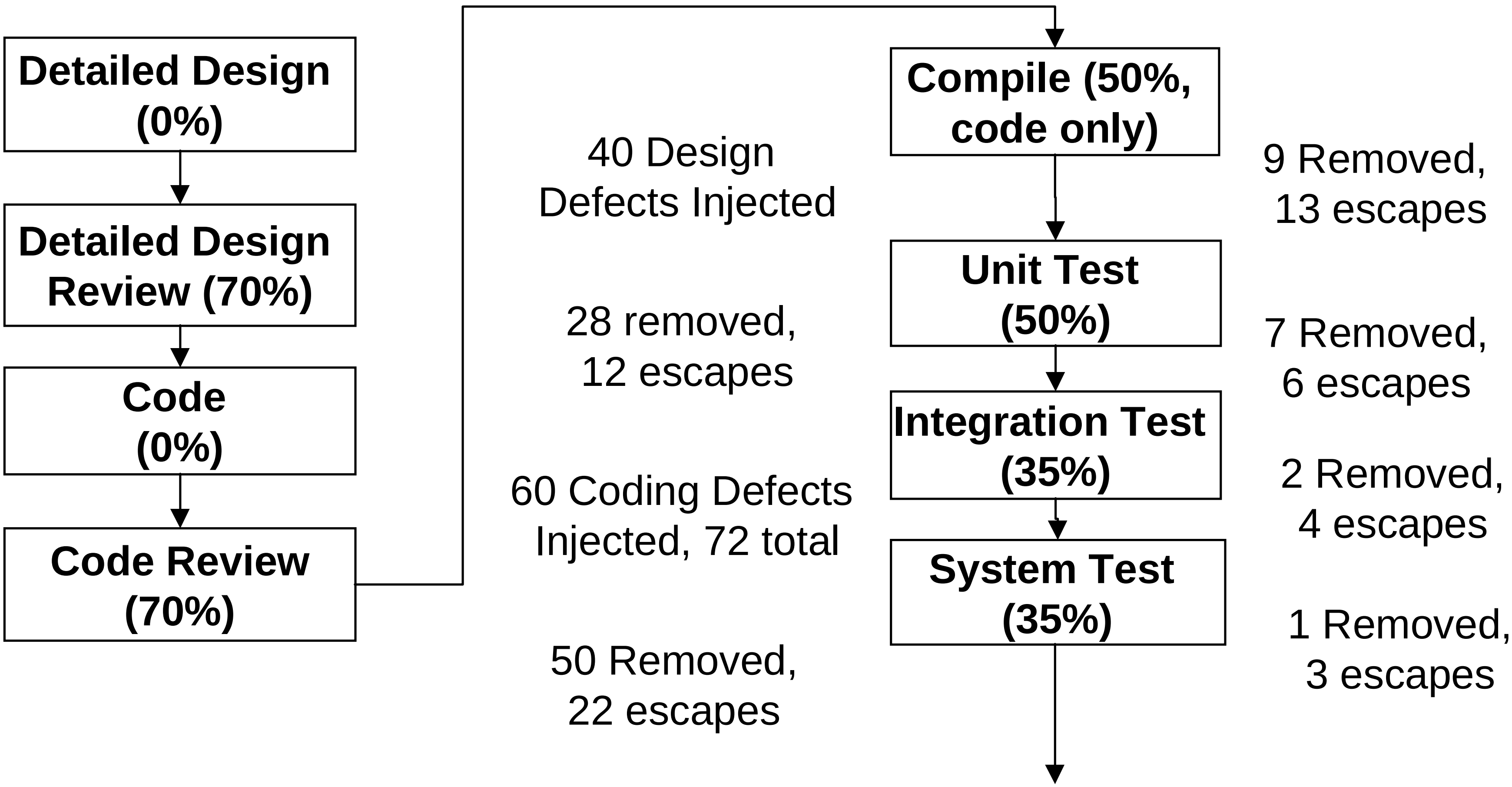
The <u>true</u> yield for the development process must include all defects injected during the development process

Yield = $n_{activity}$t/($n_{activity}$ + $n_{escapes}$).

6σ View

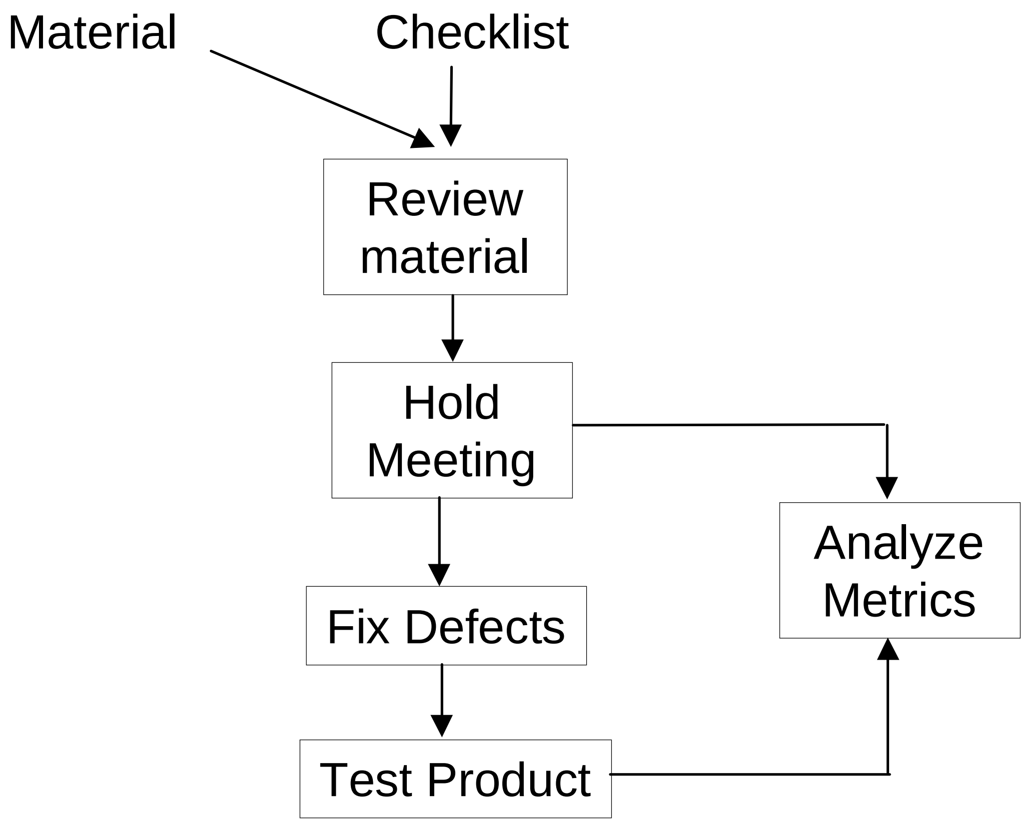


Process Yield



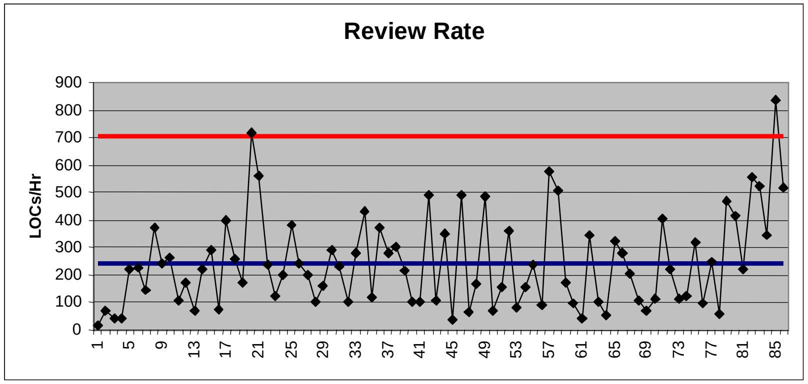
What's the yield of this process? 97/(40 + 60) = 97%

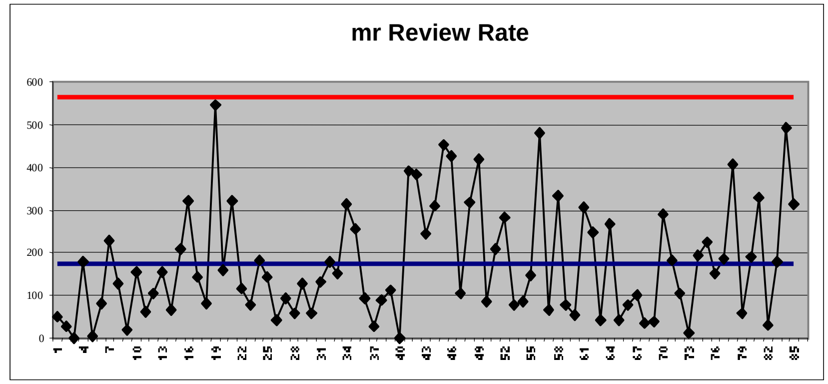
Example: Open Loop Inspection Process

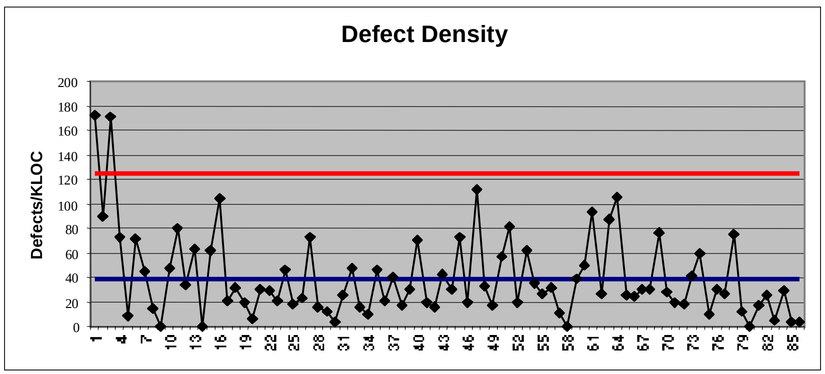


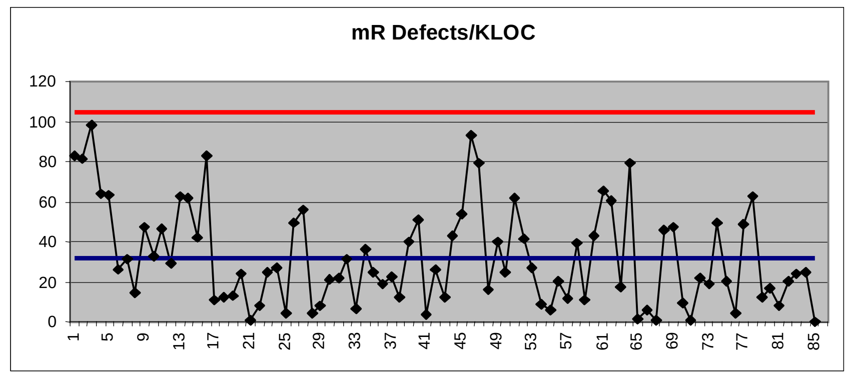
- **Optimizing the inspection process is a good place for an organization to try out a six sigma approach**

Open Loop Process Run Charts



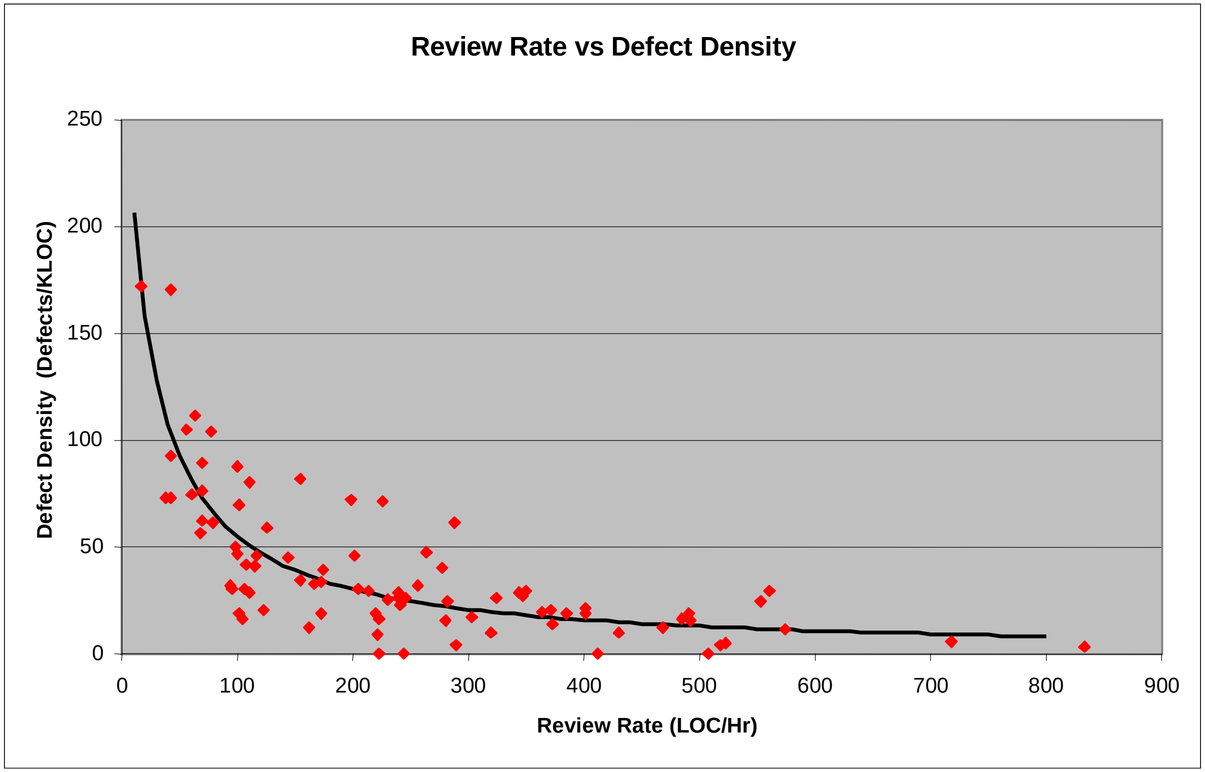






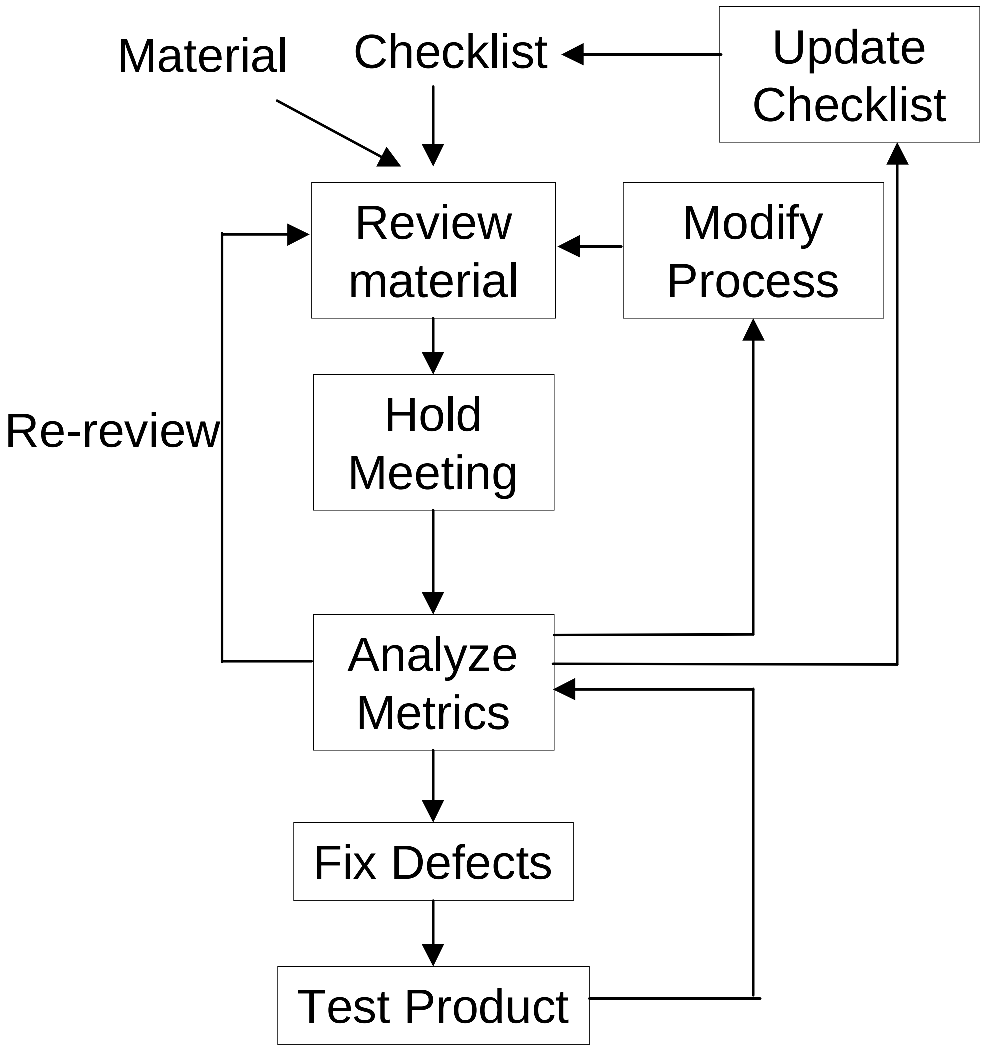
- **Average review rate 244 LOCs/Hr**
- **Average defect density 39 Defects/KLOC**
- **Average removal rate 6/Hr**

Correlation Analysis



- **To evaluate review rate for suitability as a control variable use correlation analysis**
- **r^2 = 0.67 – moderately good fit by hyperbola**
- **Chart suggests targeting review rate in the 100 – 200 LOCs hour range**

Closed Loop Inspection Process



Update Checklist

- **Remove questions that are not catching defects.**
- **Add questions to catch defects that are leaking out to test.**

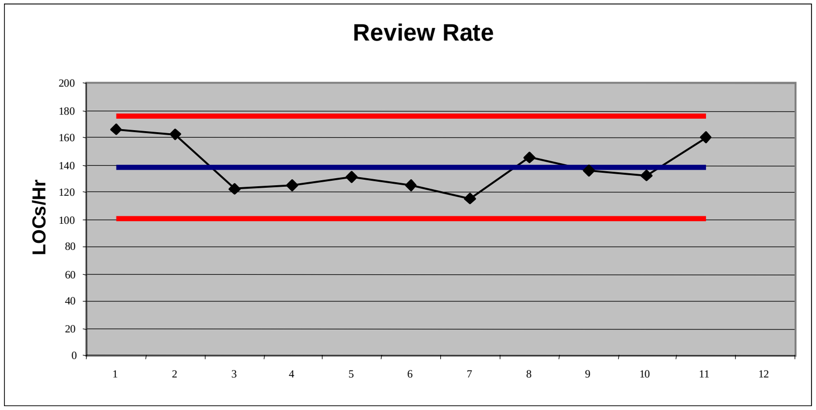
Modify Process

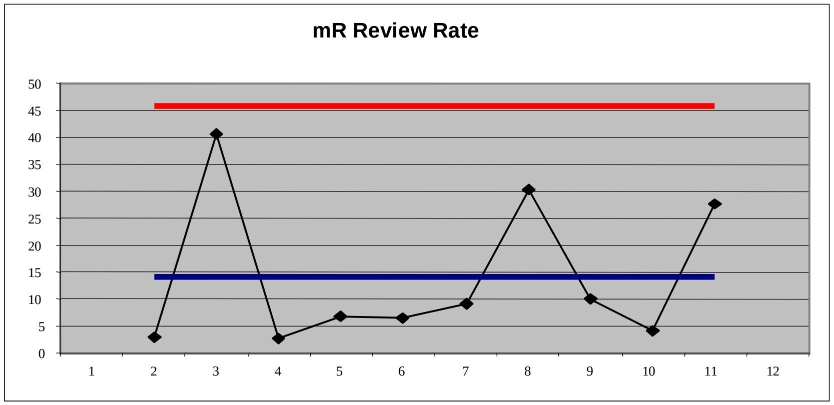
- **Modify review rate**
- **Vary size of material reviewed**
- **Include test cases**

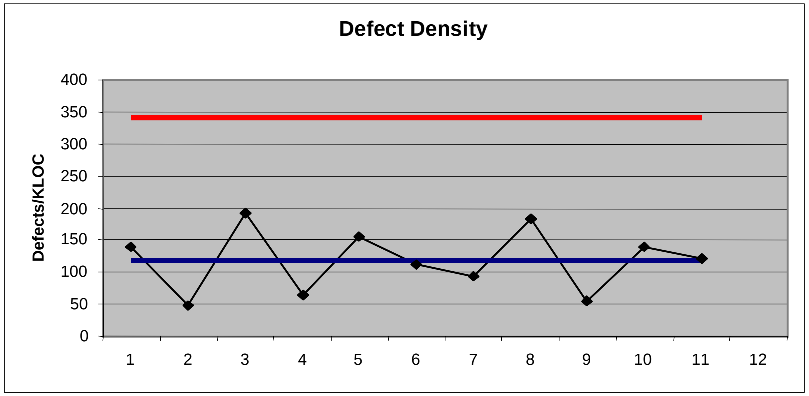
Analyze Metrics

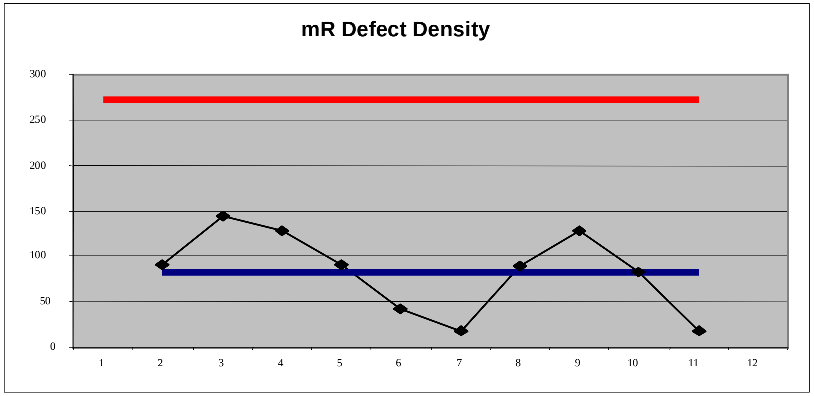
- **Process metrics:**
  - **Rate vs Yield**
- **Product metrics:**
  - **Compare yields to quality plan**
  - **Re-review of products that fall outside quality thresholds**
  - **Buggiest products list**

Closed Loop Run Charts



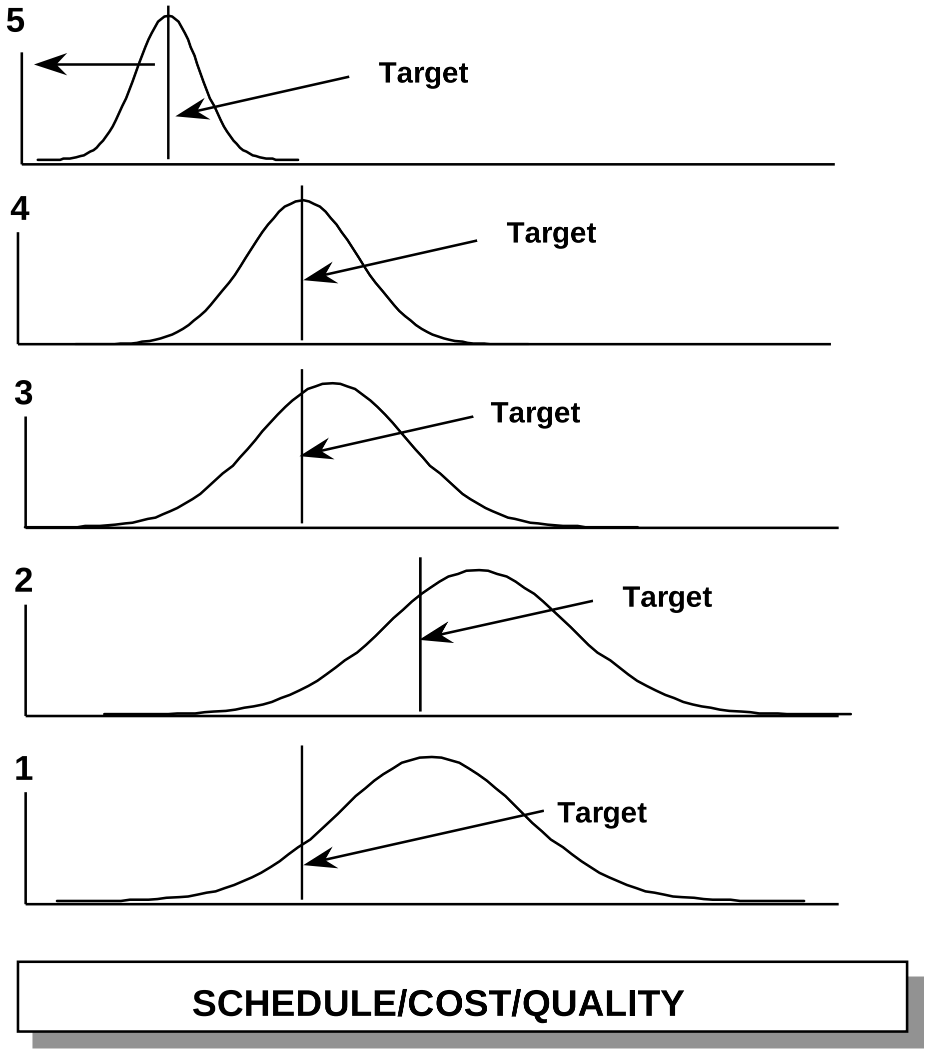






- **Average Review Rate 138 LOCs/hr**
- **Average Defect Density 118 Defects/KLOC**
- **Average Defect Removal Rate 15/hr**

CMM – A Six Sigma Perspective



- **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 corresponds to first stabilizing the process, then reducing variation, centering the process on target performance, and finally continuously improving the process by improving average performance and lowering variation**

Six Sigma and the CMM - A Summary

- CMM Levels do not equate to business results
- Closed loop processes outperform open loop ones
- Consider combining a Six Sigma approach to SPI at the tactical level with a CMM approach at the strategic level
- You only need to measure three things: size, time, and defects, but you need to measure them well
- Metrics can and should be taken at every CMM level and should be used to manage and evaluate process effectiveness
- Metrics need to be put into a statistical context before being used to make decisions
- Once you know how, you'll find most elements of the Six Sigma tool kit have broad applicability to software development and to software process improvements

Glossary of Terms

CMM®	**Capability Maturity Model**
COQ	**Cost Of Quality**
EV	**Earned Value**
KLOC	**Thousand Lines Of Code**
LOC	**Lines Of Code**
ROI	**Return On Analysis**
SEI	**Software Engineering Institute**
SPC	**Statistical Process Control**
SPI	**Software Process Improvement**

CMM® is registered in the U.S. Patent and Trademark Office.

References

- A more detailed discussion on using Six Sigma techniques to optimize inspections is being presented later at this conference on ***Wed, Nov 20 at 1:00 PM*** **in:** ***Optimizing Inspections Through the Application of Statistical Management Techniques***
- An explanation of how to use Six Sigma techniques in conjunction with Personal Software Process and Team Software Process is being presented later at this conference on ***Wed, Nov 20 at 3:45*** **in:** ***Integrating PSP, TSP and Six Sigma***
- For additional information see our web site or to answer any questions contact

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

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

www.SoftwareSixSigma.com