Six Sigma & Software Process Improvement

Washington DC SPIN

Wednesday, March 3, 2004

Steve Janiszewski

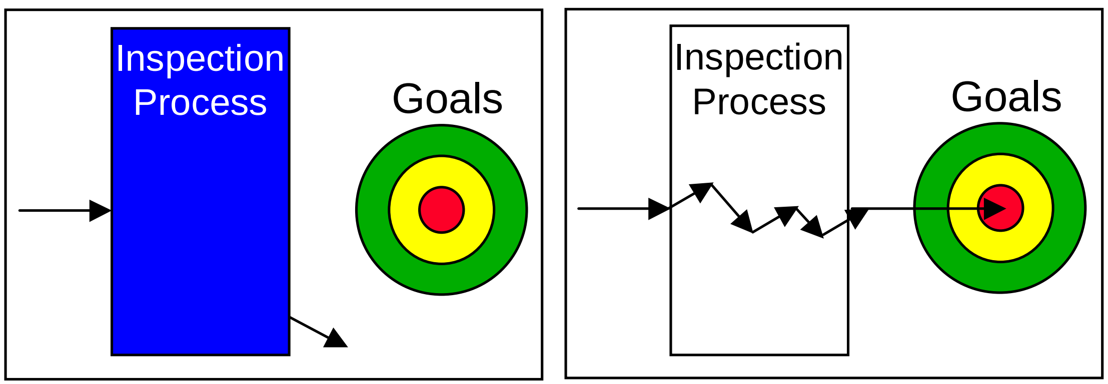
PS&J Software Six Sigma

CMM Shortcomings

- Many organizations take years to move from level to level because the CMM does not identify a specific process and because it does not provide much guidance on improvement methods
- Many more organizations fail to progress at all or drop back a level within six months of an assessment
- Many organizations don't understand how to define an efficient process and they implement processes that are CMM level 2 or level 3 compliant, but that actually add overhead without improving productivity, predictability, and product quality
- Because level 3 was the minimum level required for American military contracts, many organizations think that level 3 is "good enough". Level 3 requires most of the investment, but most of the benefits accrue at levels 4 and 5!
- Many organizations get no improvement at all because the developers don't actually use the new processes!

Measurements, Goals, & Management

- Measurements
  - If we can’t measure a process, we cannot manage it much less systematically improve it.
  - If we do not actively manage a process, it’s performance is a matter of chance
- Goals must be stated in terms of measurable quantities if we hope to achieve them



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 he asks you to quantify the return on your SPI activities?

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

"Six Sigma"

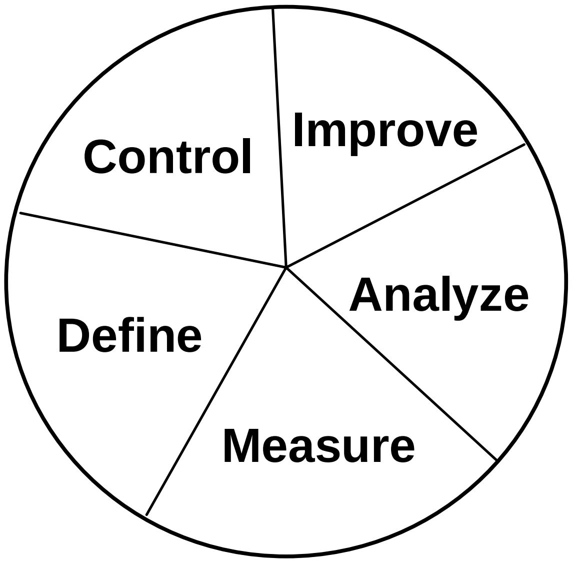
- The phrase "Six Sigma" is frequently used with three different but related meanings
- In the narrowest sense, Six Sigma is used as a measurement of product quality
  - A Six Sigma quality level means that products have less than 3.4 defects per million opportunities, i.e. the product is 99.9997% error-free
- 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

DMAIC

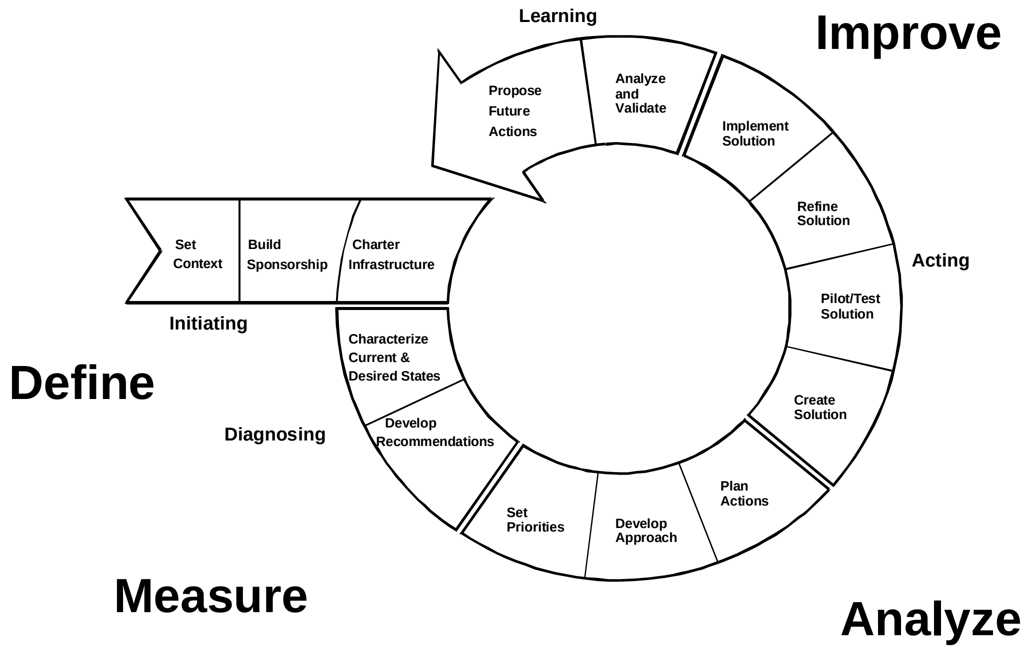
The Six Sigma Continuous Improvement Cycle

- **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**

DMAIC vs. Ideal



DMAIC MODEL

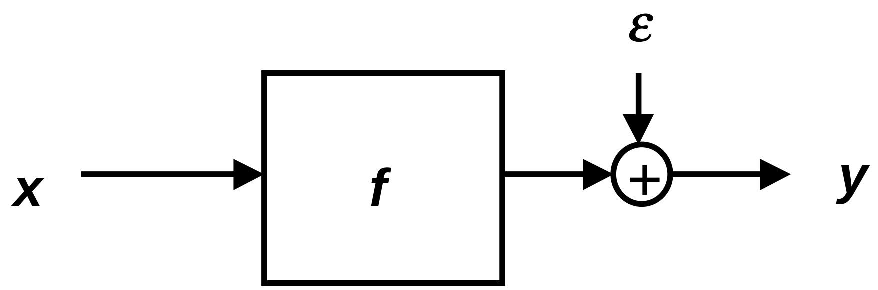


IDEAL^{SM} MODEL

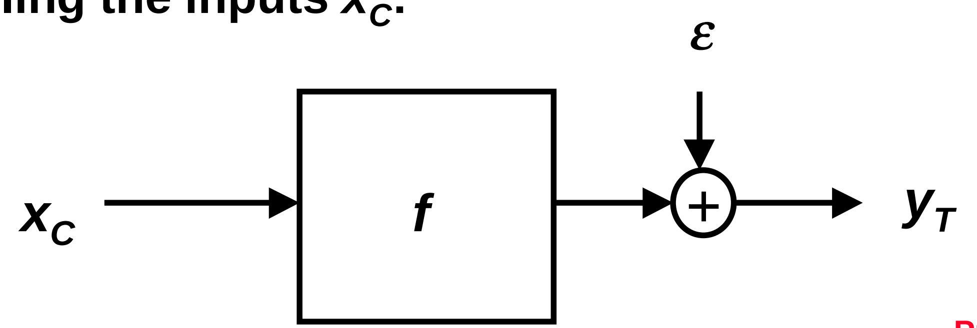
- **IDEAL cycles are long - sometimes up to two years**
- **IDEAL explicitly includes establishing sponsorship - DMAIC occurs in the context of a top down Six Sigma initiative with strong executive sponsorship**
- **DMAIC places much more emphasis on measurements and includes a control phase that is largely absent from IDEAL**
- **Either model can be used for CMM based improvement but DMAIC is more likely to yield measurable economic benefits**

A Control System Viewpoint

- Process outputs, *y*, are a function, *f*, of a set of controllable input 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

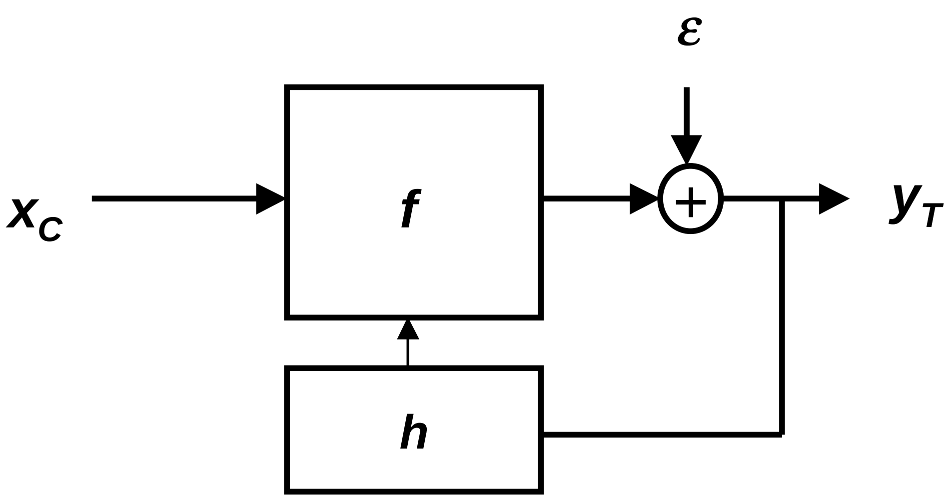


- Six Sigma techniques are used to develop a process model, identify the control variables x_C, and drive performance to targeted values y_T, by actively controlling the inputs x_C:



A Control System Viewpoint - Closing the Loop

- **For a software project, y's include cost and schedule and x's include product quality and time on task.**
- **A software process should a responsive, "closed loop" system that is controlled to achieve project goals**



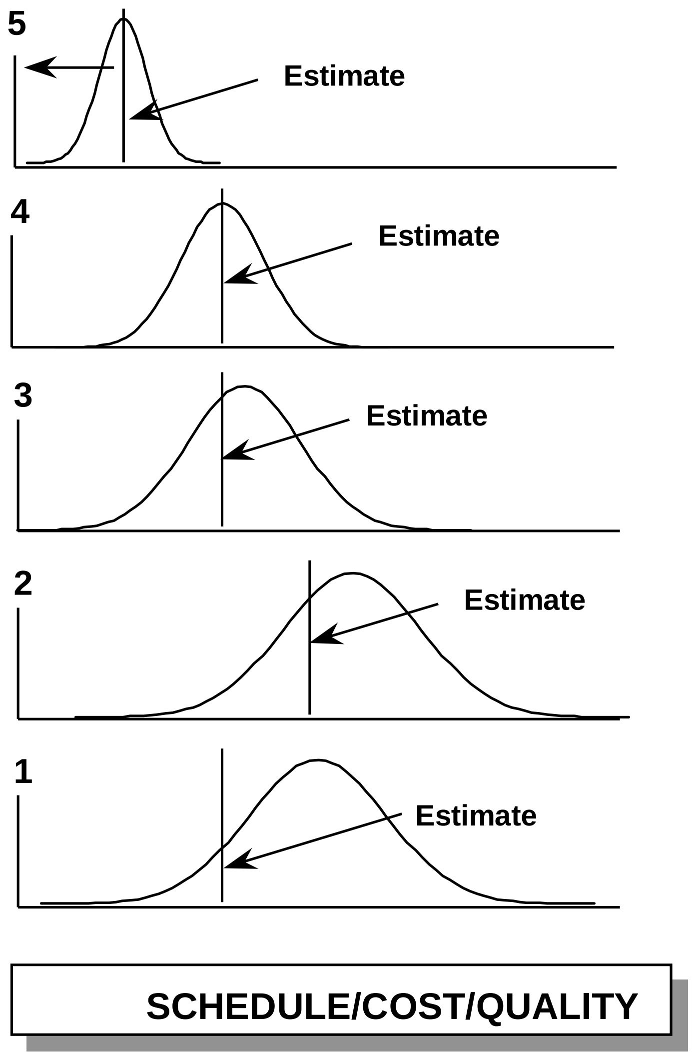
Closed loop processes are the difference between tracking and managing

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

CMM Levels – A Six Sigma Perspective



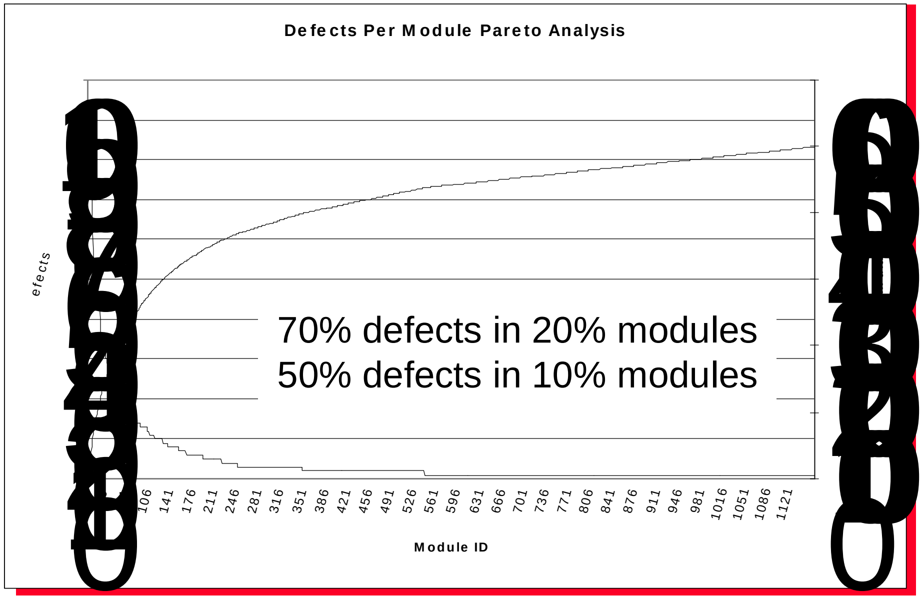
- From a business perspective, predictable process performance is a key aspect of process capability
  - Predictable performance requires a stable process
  - First step to a stable process is a "defined process"
- Moving up the CMM levels requires
  - first stabilizing the overall process,
  - centering 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

Driving Model Based SPI With Six Sigma

- Six Sigma methodology can be used to drive CMM or CMMI 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 - doesn't necessarily imply SPC
  - 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
  - Fits particularly well with CMMI's new measurement PA

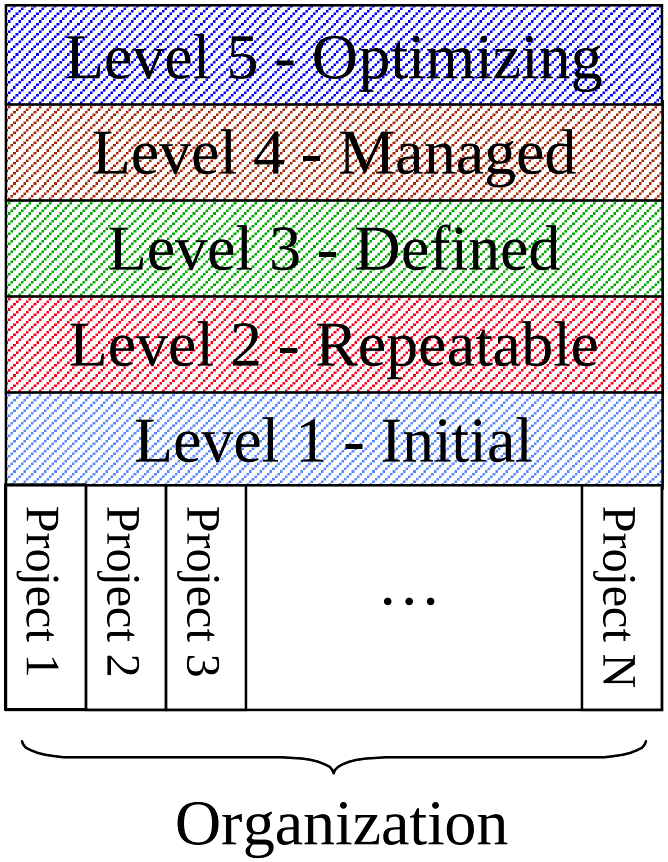
Example: DMAIC Without SPC

- Managing Integration & Test means managing defects
- Defect-prone modules can be identified by keeping a list of modules ranked by defect count
  - Consider pulling them from the link and performing a comprehensive inspection
  - If warranted, scrap and re-work



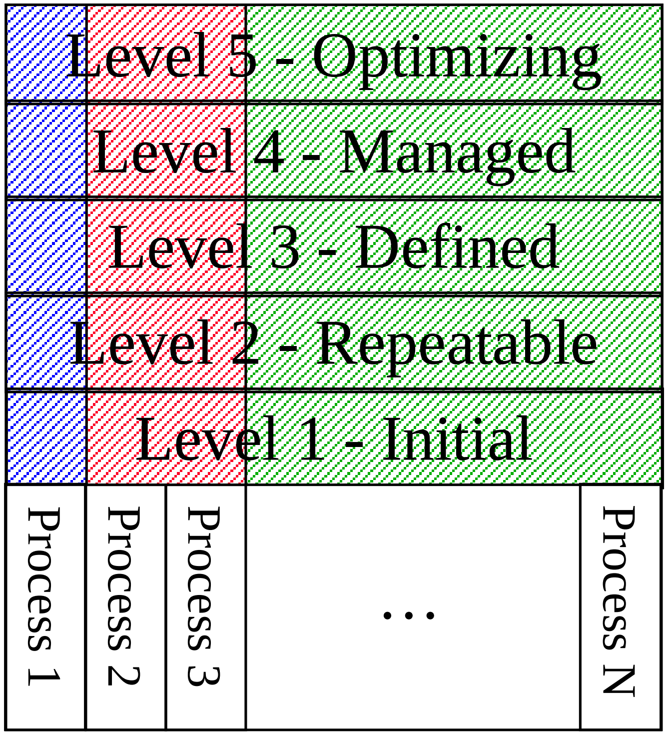
- Perform immediate root cause analysis on integration defects
  - opportunity for high value process improvements
  - escapes tend to occur in clusters, so find as many as possible with a single test failure
  - Root cause analysis starts with the author identifying
    - the mechanism by which the defect escaped earlier process steps
    - and the other modules that could have similar defects
- Defects in legacy code may warrant an automated search or an inspection targeting specific defect types

Staged Representation



- **Processes at the lower levels provide the foundation for processes at the higher levels**
- **Success at the lower levels prepares the organization to accept the changes required at the higher levels**
- **Most of the organization's projects move forward more or less in parallel one level at a time**
- **The main drawback is organizational inertia – it can literally take years to move a level**

Continuous Representation



- **Six Sigma is used to deploy processes with managed and optimizing characteristics at each individual CMMI level**
- **Selection is guided by business value and CMMI level**
- **Measurable successes are used to pull through the next round of process improvements**
- **Number of processes operating with managed and optimizing characteristics grows as organization moves from level 1 to level 3**
- **Organization moves from level 3 to level 5 very quickly**

Some 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 and they wait till they are ready to go to level 4 before they even consider Six Sigma
  - Miss opportunity to make their CMM effort more likely to succeed and to achieve measurable business results
- Some organizations attempt to provide the same Six Sigma training to everyone resulting in sending software engineers to training courses appropriate for manufacturing
- 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 because “software is different”
  - The detailed application of Six Sigma to software is significantly different, but the principles and results are very similar

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
- 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

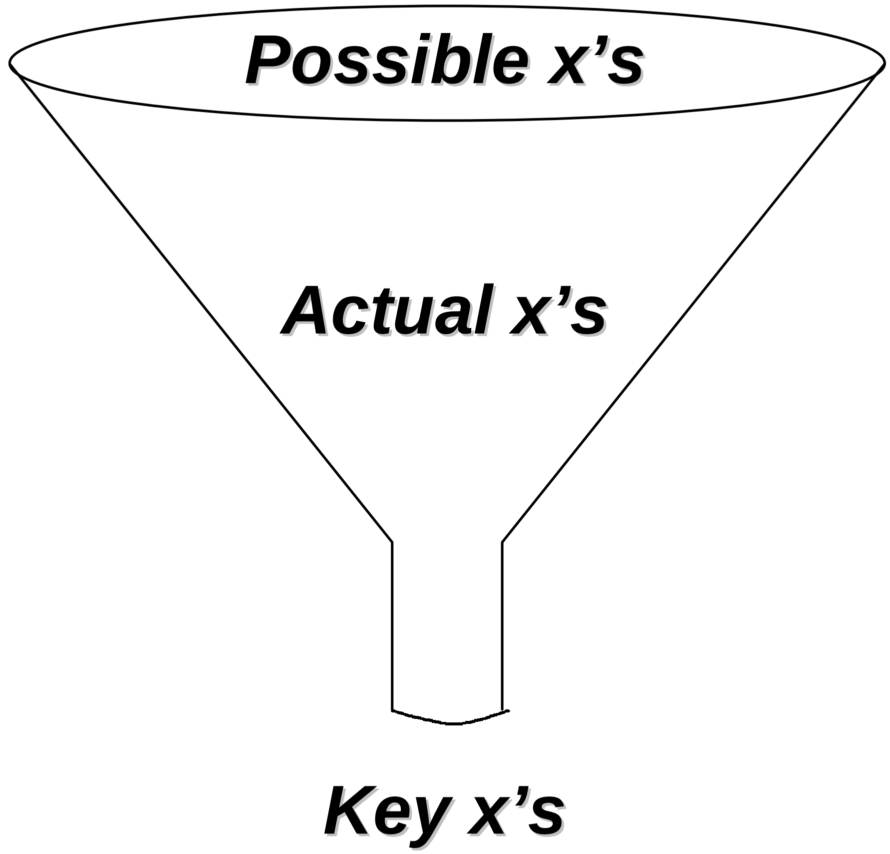
Six Sigma is applicable and has the potential for dramatic performance improvements

Making Effective Improvements

- Data driven continuous improvement requires the active participation of each software developer
  - managers and support staff cannot improve productivity & product quality because they don't develop the products
  - making an improvement requires changing current practice, something only developers can do
    - "improvements" that don't significantly change what the developers do at their desks are likely to be marginal or counterproductive
  - metrics are not likely to be collected for long or very consistent unless the people collecting them use the metrics and see their value
- To be successful at continuous improvement, developers must
  - be trained to understand the value of process, how to collect and use metrics, and how to make and measure improvements
    - Six Sigma uses "green belt" & "black belt" training
  - understand the mechanisms that make processes effective
  - be given quantitative improvement goals and motivated to meet them
  - be given the time to analyze the data and make improvements (post mortem phase)

Using the Six Sigma Toolkit

Define & Measure	• Process Maps • Descriptive Statistics • Statistical Process Control (SPC) • Measurement System Evaluation (MSE)
Analyze	• Failure Mode Effects Analysis (FMEA) • Root Cause Corrective Action (RCCA) • Statistical Process Control (SPC) • Regression Analysis • Analysis of Variance (ANOVA) • Design Of Experiments (DOE)
Improve & Control	• Statistical Process Control (SPC)



$$y = f(x)$$

Six Sigma Toolkit is a more or less standard set of statistical tool for data analysis and modeling

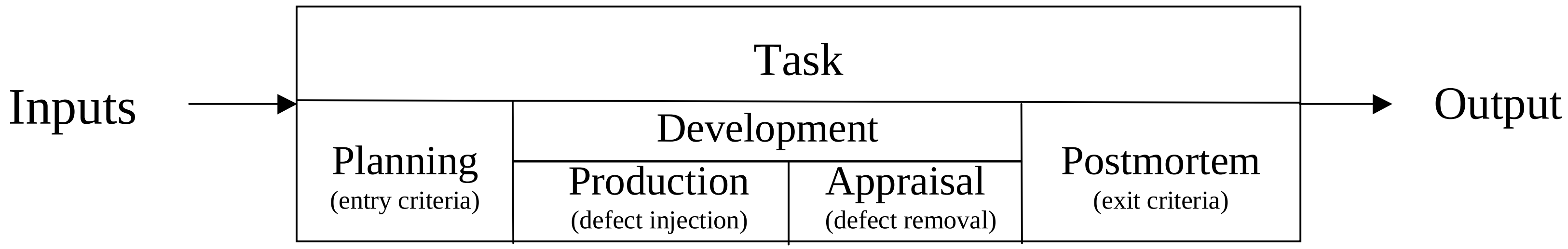
Software Six Sigma Measurement Framework

- In order to address the typical business goals: costs, cycle time, product quality, predictability, Software Six Sigma uses a simple measurement framework that fully characterizes software development processes with three simple measurements
  - Time: the time required to perform an activity
  - Defects: the number & type of defects, fix time, point of injection, point of removal, and a description adequate for root cause analysis or creation of a checklist question
  - Size: the size of the work product produced
- Time & defect metrics are collected <u>in-process</u>
- Size metrics are collected during task post mortem
- In general, there are far fewer variables that affect a software development process than a manufacturing process
  - No issues with variables like raw materials, temperature, humidity, curing time, measurement tolerances, etc.

Operational Definitions

- An operational definition is a precise description of how to measure a data item
  - It defines the data and provides a clear procedure for measuring the data
- The goal of an operational definition is to make sure all data collectors measure a characteristic in the same way
  - Removes ambiguity
  - Reduces variation in measurements
- A Six Sigma approach should use very precise operational definitions for size, time, and defects and provide an automated means to support complete consistent data collection
- Example: Time measurement
  - Record the time (M/D/Y H:M) when you begin or resume the activity as the start time
  - Record the activity phase. Note any comments
  - When you complete the phase or are interrupted record the stop time
  - Elapsed time is the difference between stop and start times in minutes

Structuring Tasks for Measurement



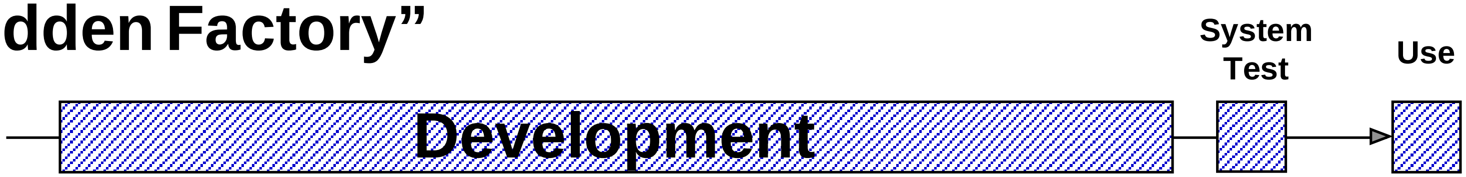
- **Tasks transform inputs to outputs**
- **In Six Sigma, tasks should always begin with a planning phase and end with a postmortem phase**
  - **Beginning planning requires meeting task entry criteria**
  - **Completing postmortem requires meeting task exit criteria**
  - **Planning sets up expected process performance and product quality targets that serve as exit criteria**
- **In between planning & postmortem is development**
  - **Consists of at least one production phase & one appraisal phase**
- **Since human beings are involved**
  - **production brings with it defect injection**
  - **appraisal carries with it the opportunity for defect removal**

Process Performance Models

- In order to understand the relationship between cost and quality, we can begin by modeling the economics of the software development process
- Starts with quantitative business goals providing direct value to the customer
  - Frequently they are on-time delivery, cost, cycle time, and product quality
- In order to understand where and how to improve, one needs to understand how the development process x's drive these process y's
  - Costs are driven by productivity, product quality, and time on task
  - Cycle time is driven by cost and time on task
  - Predictability is driven by product quality

Software's Hidden Factory

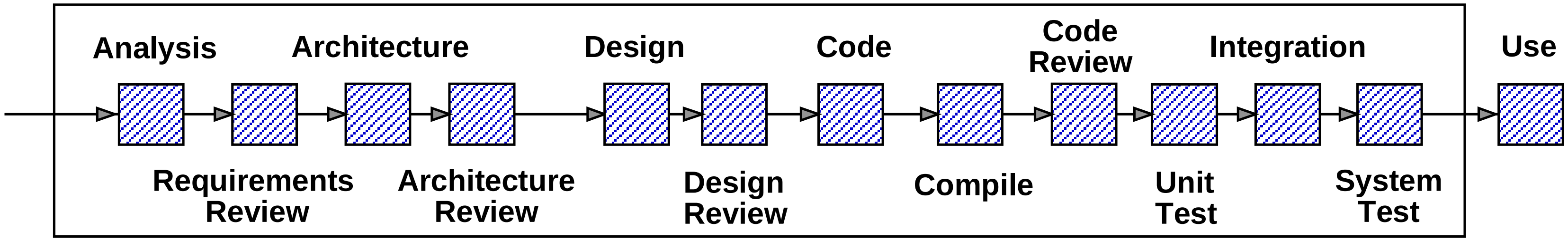
"Hidden Factory"



Defects not recorded prior to system test

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

Six Sigma



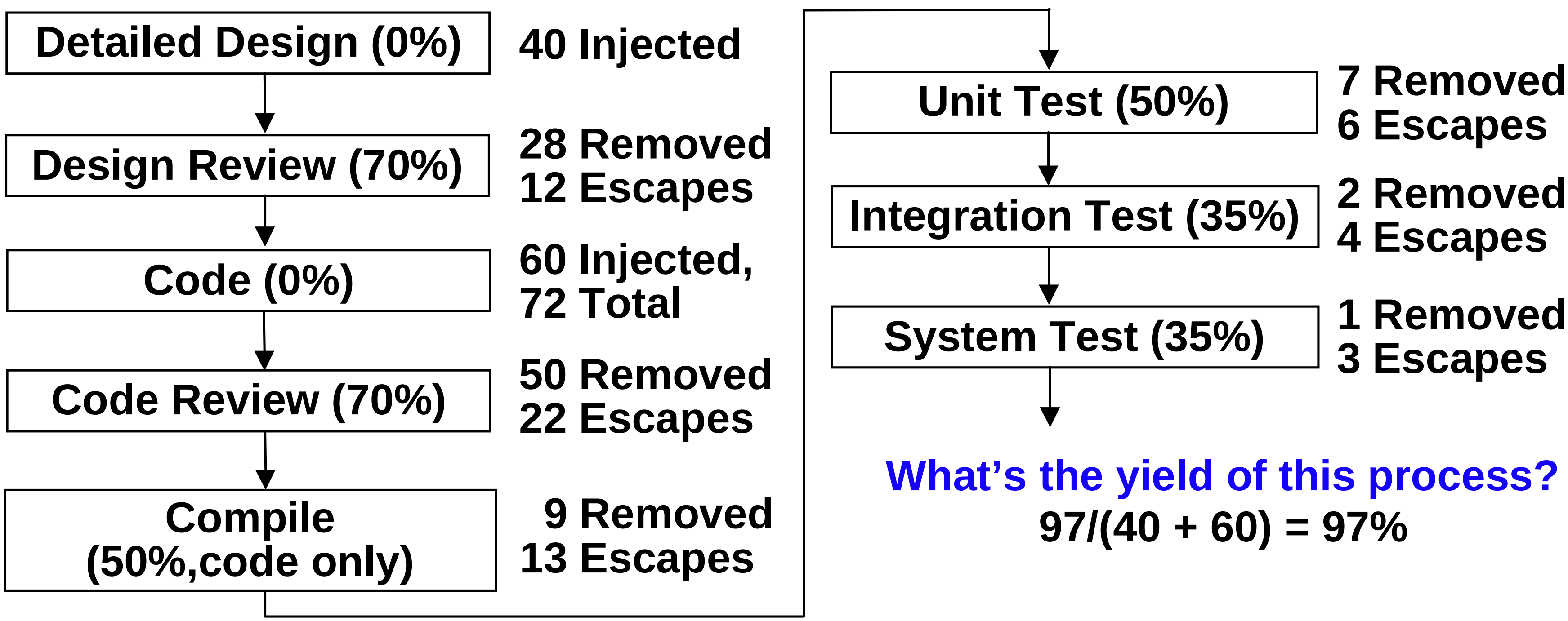
True yield includes all defects injected in development

$$\text{Yield} = n_{development} / (n_{development} + n_{escapes})$$

Usually, the later a defect is removed, the higher its removal costs

Working with Yields

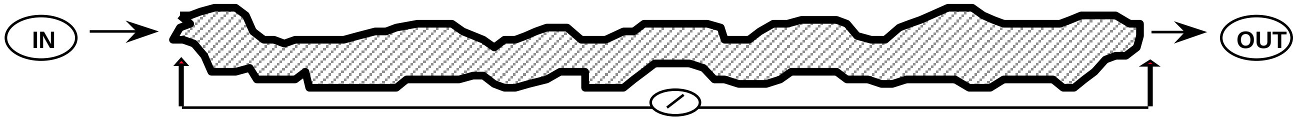
- Historical injection numbers and yields can be used to estimate the number of defects that will be removed during each phase
  - typical inspection yields are 50% - 80%
  - typical testing yields are less than 50%
  - typical injection rates are 100/KLOC



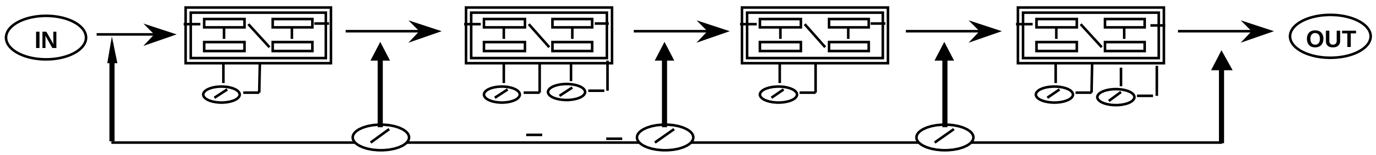
What's the yield of this process?

97/(40 + 60) = 97%

Process Modeling & Design



Are the business goals achievable?



Code & Test

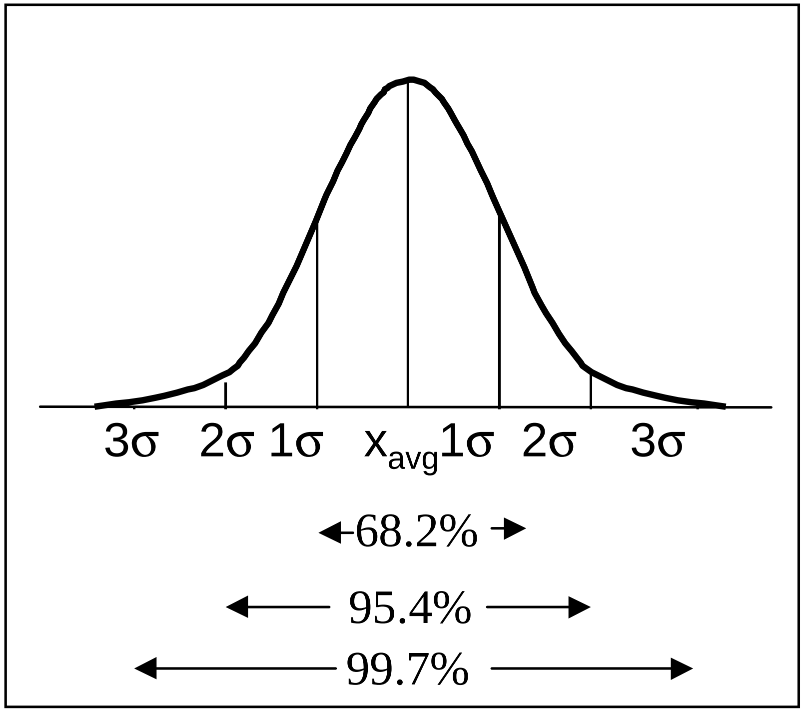
Phase	Remaining	Injected	Present	Yield	Removed	Cost/Defect	Activity Cost	Failure Cost
Design	0.0	40	40.0	0%	0.0	5	0	0
Design Review	40.0		40.0	0%	0.0	5		0
Design Inspection	40.0		40.0	0%	0.0	10		0
Code	40.0	60	100.0	0%	0.0	1	2000	0
Code Review	100.0		100.0	0%	0.0	3		0
Compile	100.0		100.0	50%	50.0	1		50
Code Inspection	50.0		50.0	0%	0.0	5		0
Unit Test	50.0		50.0	50%	25.0	12	180	300
Integration	25.0		25.0	35%	8.8	300	180	2625
System Test	16.3		16.3	35%	5.7	600	180	3413
Customer Test	10.6		10.6	35%	3.7	1200	180	4436
Total (minutes)							**2720**	**10824**

Grand Total (hrs) **226**

Variation

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

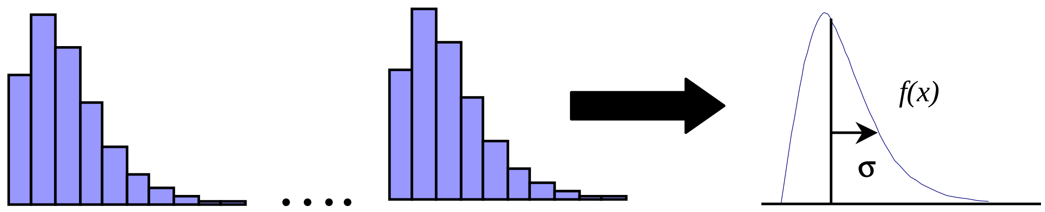
$$\sigma = \sqrt{\frac{1}{n-1}\sum (x - x_{avg})^2}$$



- 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

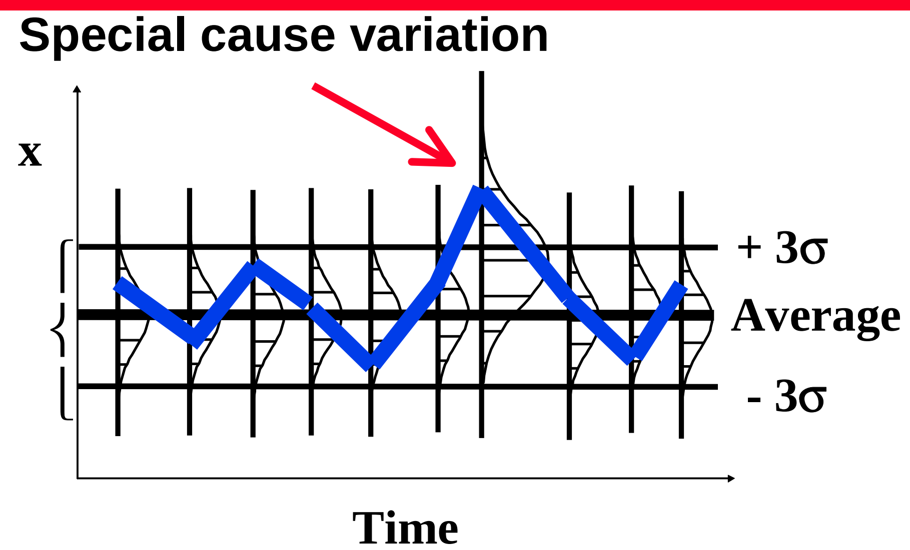
- A process exhibits statistical control when a sequence of measurements $x_1, x_2, x_3,\ldots x_n,\ldots$ has a consistent and predictable amount of variation
- It is possible to model this pattern of variation with a stationary probability density function f(x)



- 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

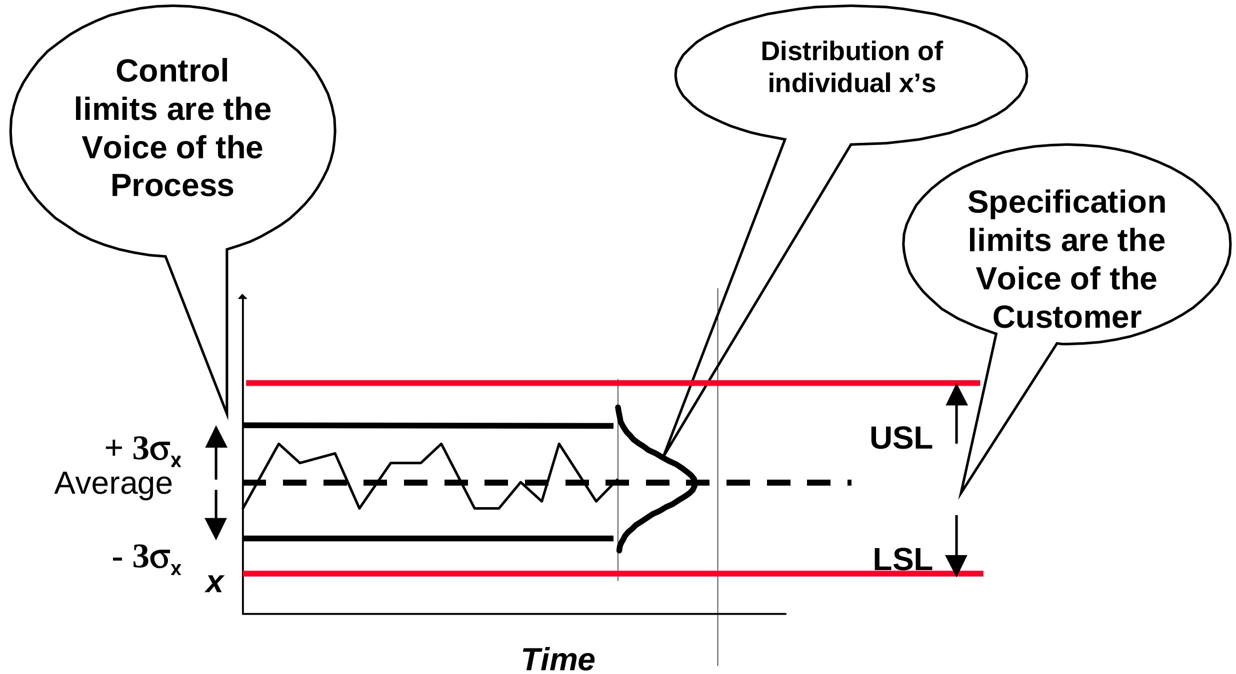
Control Charts

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



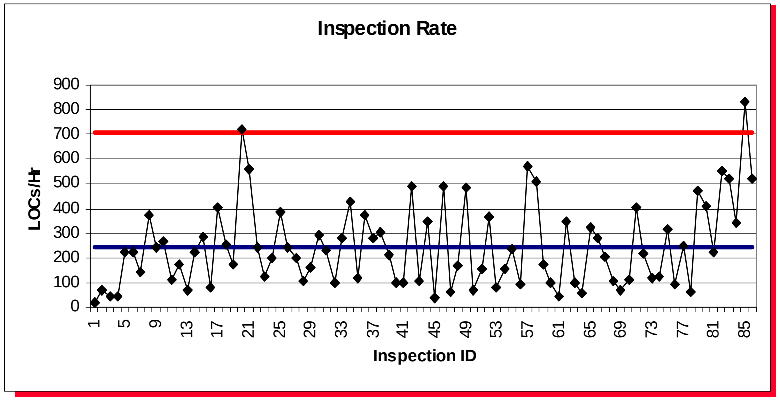
- Common cause variation is normal random variation in process performance
  - Don't over-react to common cause variation
  - Reduction requires a process change
- Special cause variation represents an exception to the process
  - Actions to correct special cause variation must eliminate a specific assignable 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

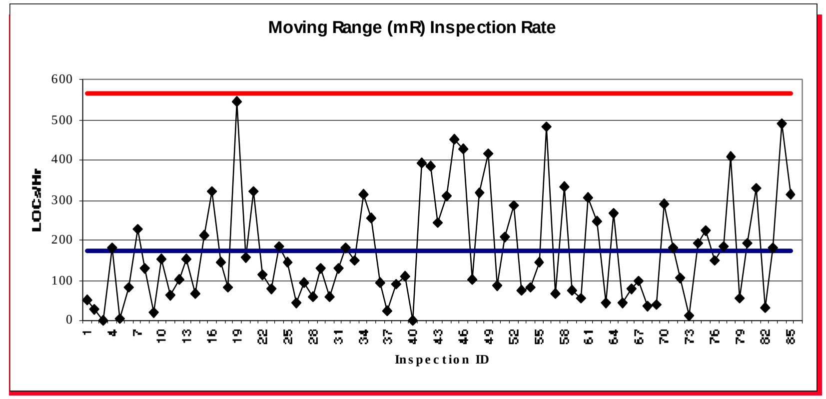
Reading XmR Charts

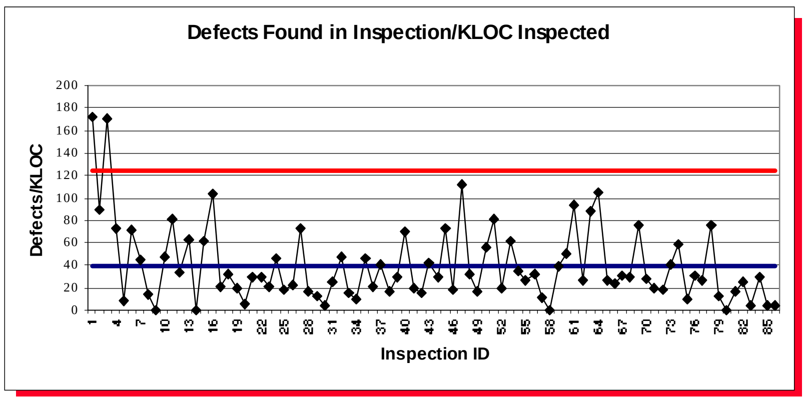


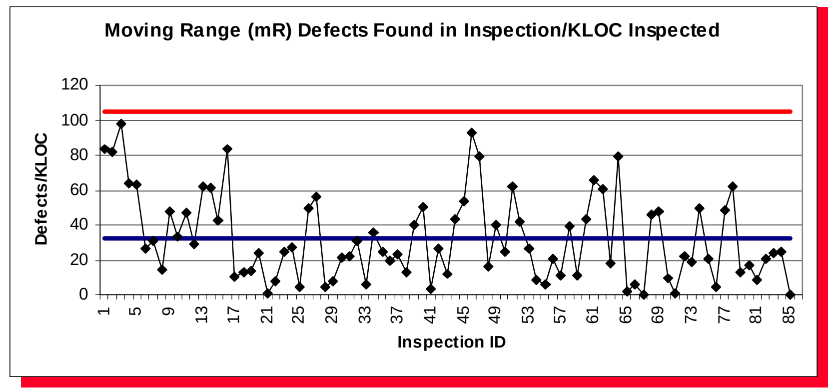
- There are many types of control charts in the Six Sigma Toolkit
- The XmR chart is most useful for software development
- Consists of two charts: X & mR (moving Range of X)
  - mR chart must be in control before you can interpret X chart
  - Sigma estimated from average moving range
- Special causes are indicated by points outside the control limits, runs of points to one side of the central line, and trends

Open Loop Process Run Charts



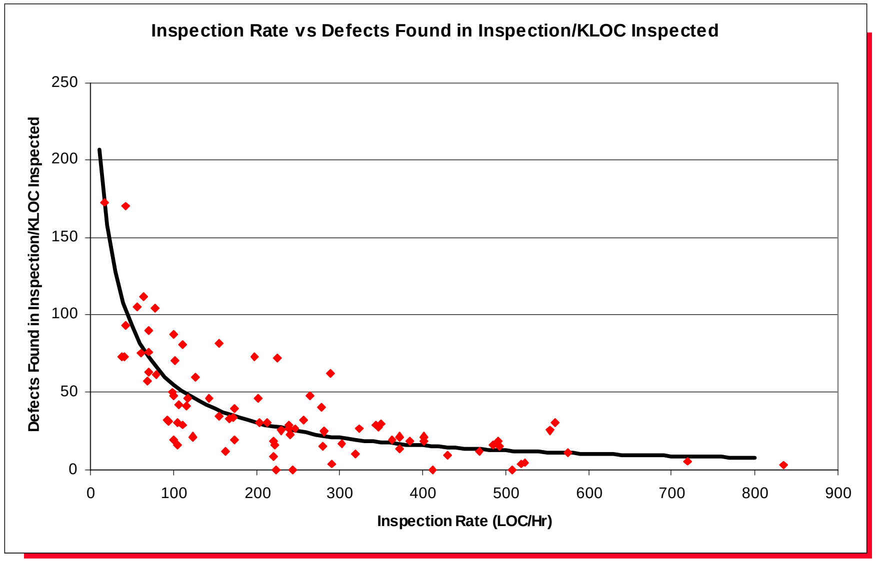






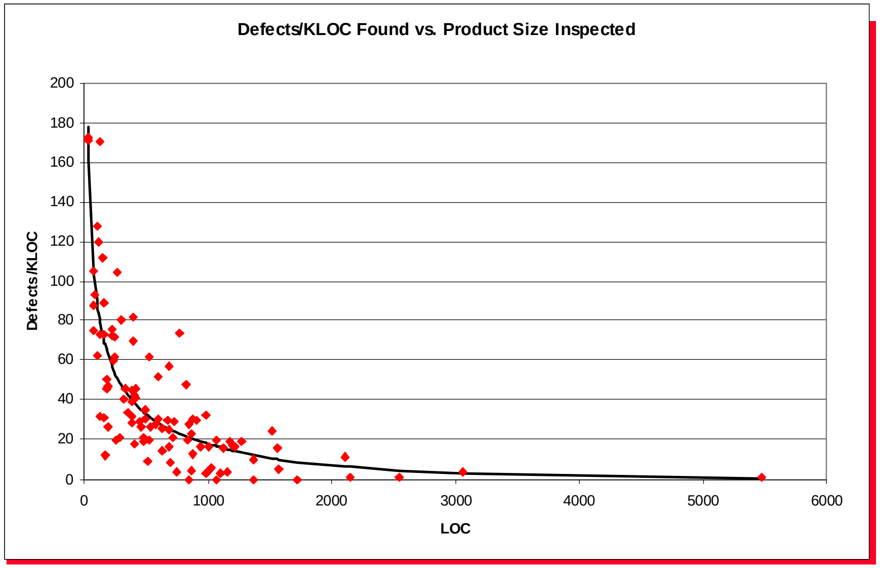
- **Average review rate 244 LOCs/Hr**
- **Average defect density 39 Defects/KLOC**
- **Average removal rate 6/Hr**

Correlation Analysis



- **To evaluate review rate for suitability as a control variable use correlation analysis**
- **$r^2 = 0.67$ – moderately good fit by hyperbola: $y = 1000/(0.1x + 3)$**
- **Chart suggests targeting review rate in the 100 – 200 LOCs hour range**

- **Similar analysis show dependency on size of product under review**
- **$r^2 = 0.68$ – moderately good fit by hyperbola: $y = 1000\exp(-x/2000)/ (x)^{1/2}$**
- **Charts suggests very little value in inspection review of large products**
- **Target product size < 500 LOCs**



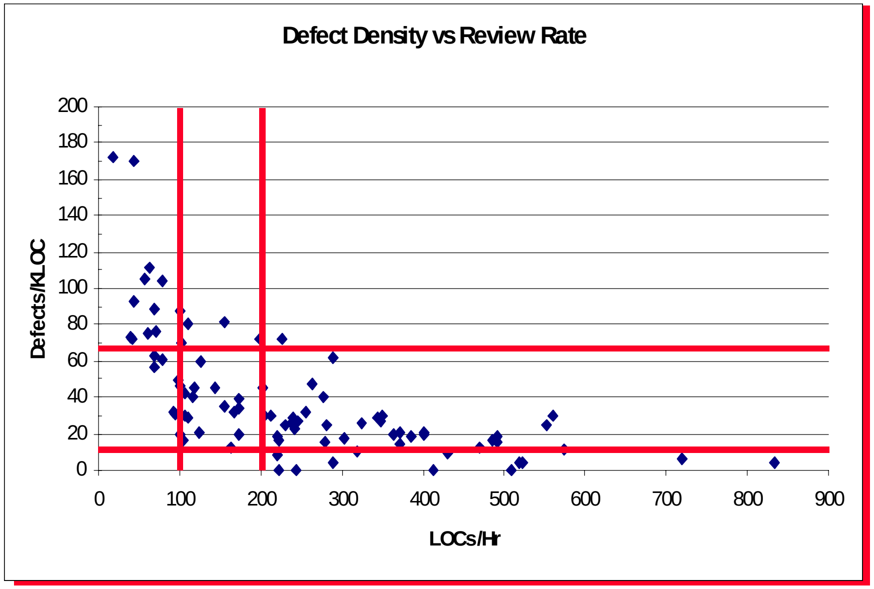
Inspection Action Plan

Slow Review Rate & Many Defects

Is the product really buggy?
Was the review really effective?
Was the review cost efficient?

Fast Review Rate & Many Defects => Buggy Product

The product is buggy.
Return to author for rework
Ask someone else to rewrite



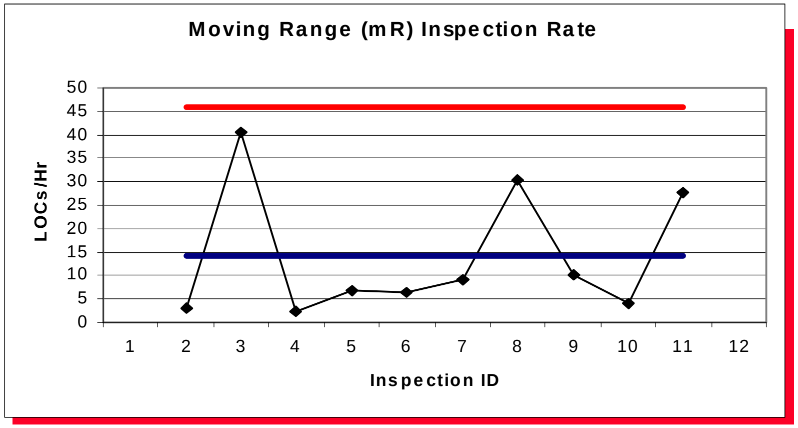
Slow Review Rate & Few Defects

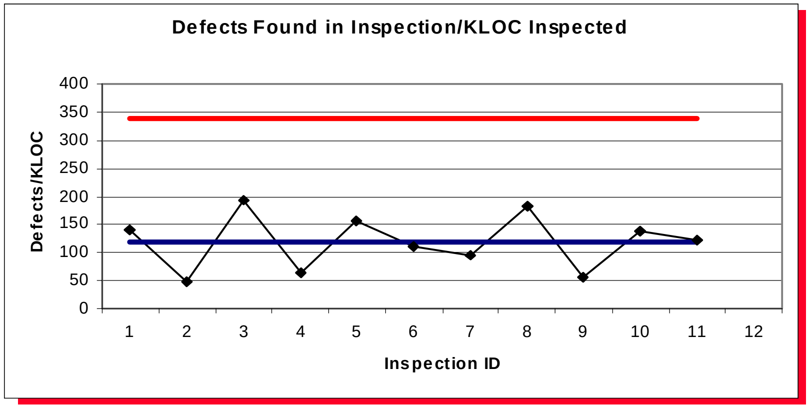
Is the product really good?
Was the review really ineffective?
Was the review cost efficient?

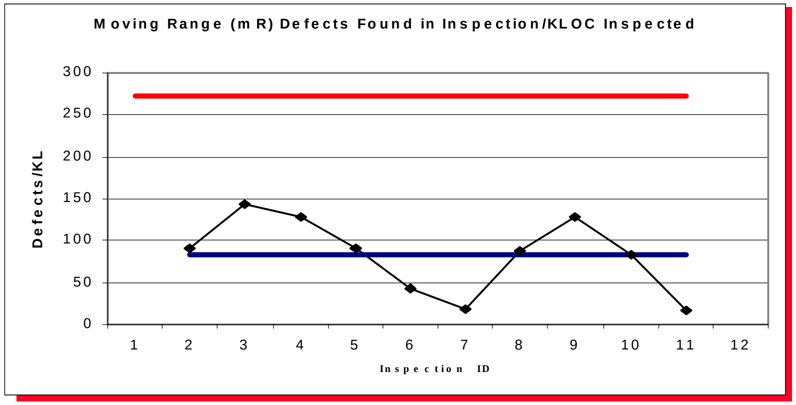
Fast Review Rate & Few Defects => Poor Review

Is the product really good?
Re-review at a slower rate
Make sure reviewers are using the checklist

Closed Loop Run Charts







- **Targeting rate yielded major decrease in variation**
- **Closed loop process achieved significant improvements**
  - **Average Review Rate 138 LOCs/hr**
  - **Average Defect Density 118 Defects/KLOC**
  - **Average Defect Removal Rate 15/hr**

Why Adopt Six Sigma?

- Achieve bigger savings, lower cycle times, and better predictability 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
- Demonstrate a quantitative connection between process improvement and business goals
- Maintain sponsorship through reorganizations and changes in senior management
- Exploit corporate Six Sigma sponsorship to boost your SPI initiative
- Accelerate progress to higher CMM levels

References

- For more on statistical process control see
  - *Understanding Statistical Process Control*, Donald J. Wheeler & David S. Chambers, SPC Press, 1992
- For additional information see our web site or for questions contact:

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

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

www.SoftwareSixSigma.com