Applying Six Sigma and Statistical Quality Control to Optimizing Software Inspections

New Jersey SPIN

Wednesday, November 19, 2003

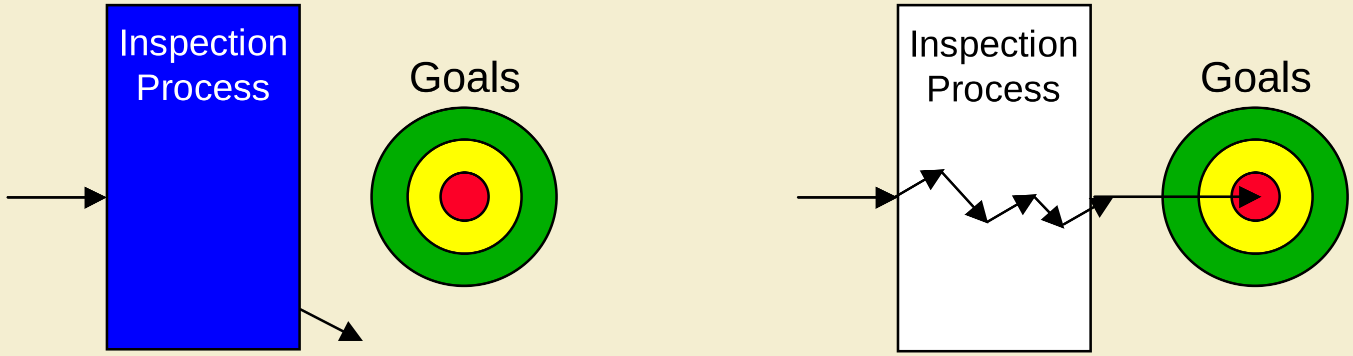
Ellen George

Steve Janiszewski

PS&J Software Six Sigma

Goals and Measurement

- An inspection process that is not *actively managed* will probably be less effective in achieving its goals. It might even be counterproductive
- "You can't manage what you can't measure"
- Goals should be stated measurably
- Measures should be defined



Measurements of the inspection process are key to managing the process and achieving the goals

Measurements

- Only three basic measurements
  - Effort: the effort required to prepare for, hold, and fix the defects found in, the inspection
  - Size: the size of the work product inspected, often measured in lines of code (LOC)
  - Defects: the number and type of defects, effort required to fix, point of injection and point of removal, description
- Development effort should be proportional to size
- Defect density should be proportional to size
- Size units should be chosen so that average defect density is not “too small”
- Simple and economical to collect in-process with an automated tool
- All other metrics are derived from these three measurements

Derived Measurements

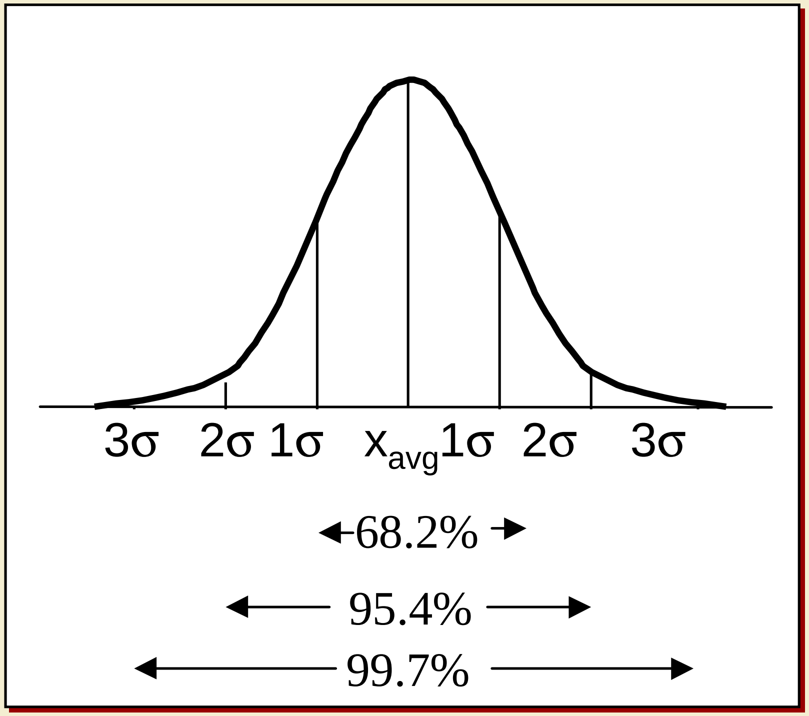
- Review Rate - LOC/hr
- Defect Density - Defects/KLOC
- Defect Injection Rate - Defects/hr
- Defect Removal Rate - Defects/hr
- Yield - Defects Removed/Defects Present
- Defect Removal Leverage - Inspection Removal Rate/Test Removal Rate
- Appraisal Cost of Quality – cost of all inspection activities expressed as a % of project cost
- Failure Cost of Quality – cost of all re-work related activities required to complete compilation and test expressed as a % of project cost

www.SoftwareSixSigma.com
(201) 947-0150, (201) 947-8828

Characterizing Variation

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

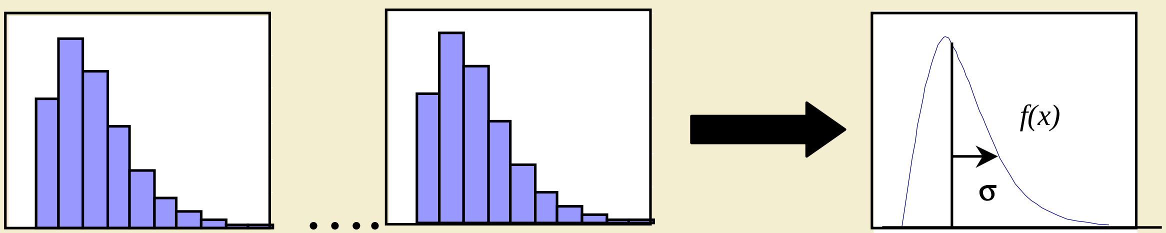
$$\sigma = \sqrt{\frac{1}{n-1}\sum (x - x_{avg})^2}$$



- **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**

Process Stability and Statistical Control

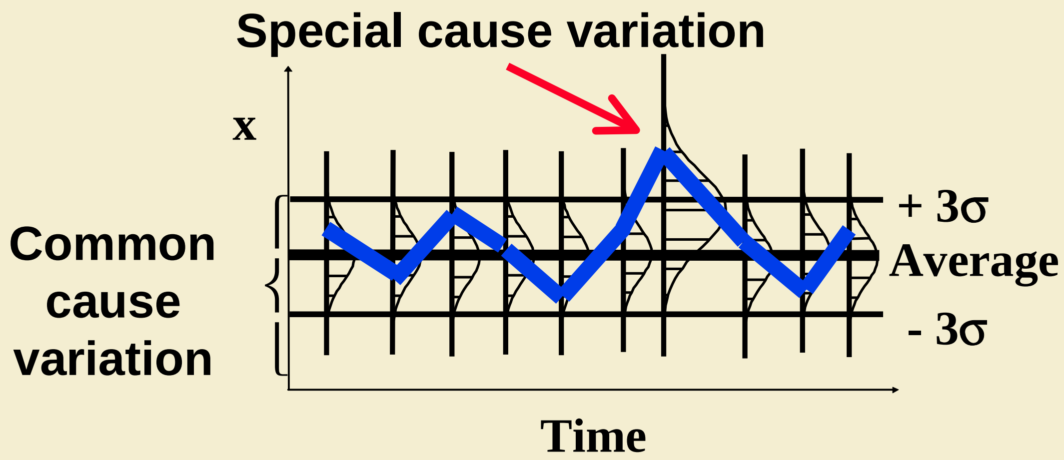
- A process exhibits statistical control when a sequence of measurements $x_1, x_2, x_3, \ldots x_n, \ldots$ has a consistent and predictable amount of variation
- It is possible to model this pattern of variation with a stationary probability density function f(x)



- 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 and Process Variation

- 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

- **Used with continuous data (measurements)**
- **no assumptions about underlying distribution**
- **Appropriate for items that are not produced in “batches” or when it is desirable to use all available data**
- **two charts: X and mR (moving Range of X)**
- **mR_{avg} is used to estimate σ for X as well as mR**

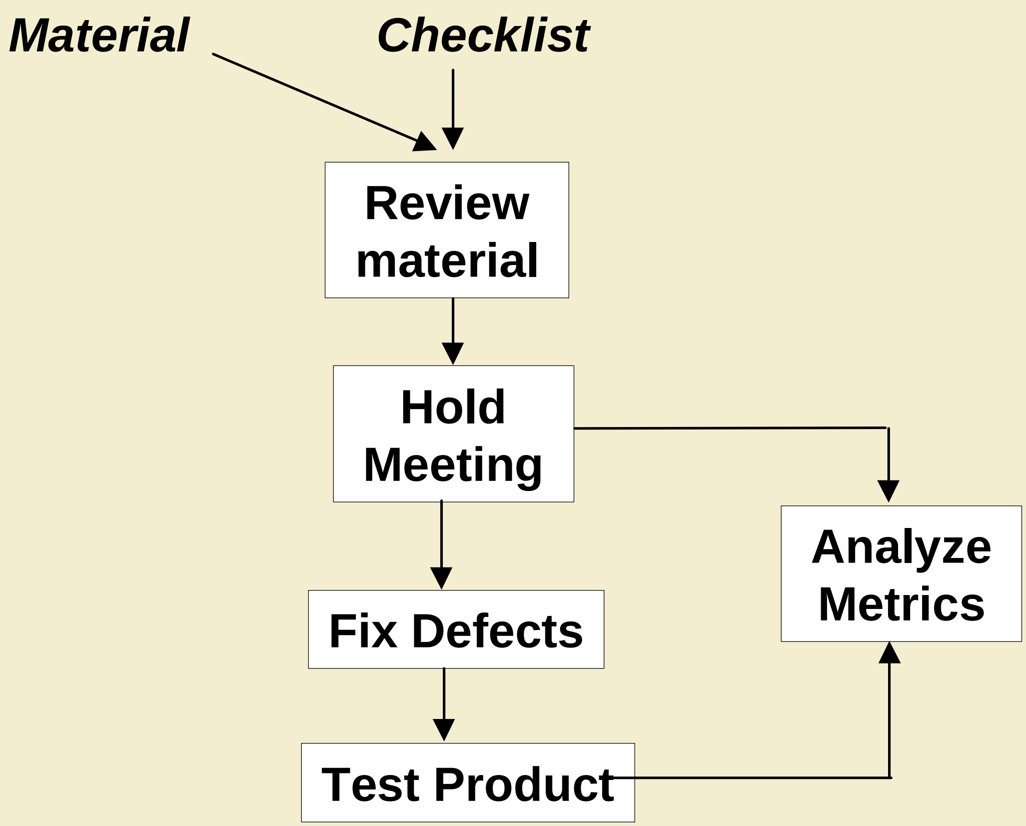
- **$mR_i = | X_i - X_{i-1} |$**
- **X chart mean: X_{avg}**
- **X chart control limits: $X_{avg} \pm 2.660\ mR_{avg}$**
- **mR chart mean: mR_{avg}**
- **mR chart control limit: $3.268\ mR_{avg}$**

Detecting Assignable Causes

- X is out of control whenever
  - a single point x_i falls outside the three sigma control limits CL_x
  - at least two out of three successive x_i's fall on the same side of, and more than two sigma units away from, the central line
  - at least four out of five successive x_i's fall on the same side of, and more than one sigma unit away from, the central line
  - at least 8 successive x_i's fall on the same side of the central line
- R is out of control when
  - 8 or more successive r_i's fall on same side of median
  - or 12 or more successive r_i's fall on same side of mR
- A trend is any upward or downward movement of 5 or more consecutive points
- Use of control charts to quantify normal variation and to identify the presence of assignable causes is called Statistical Process Control (SPC)

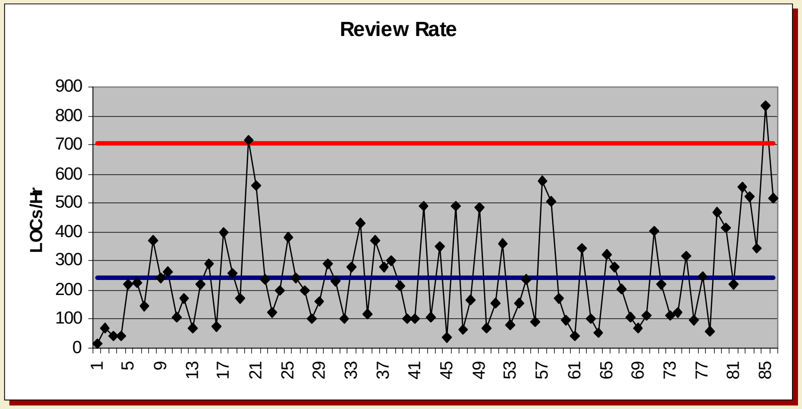
Never attempt to interpret the X chart when the mR chart is out of control !

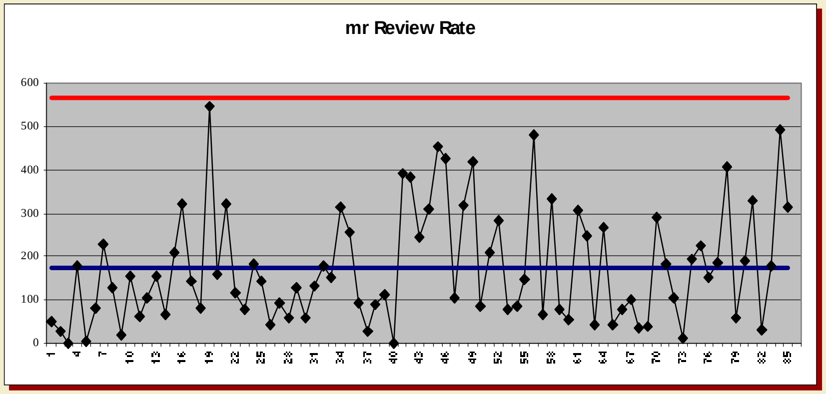
Open Loop Inspection Process - Tracking

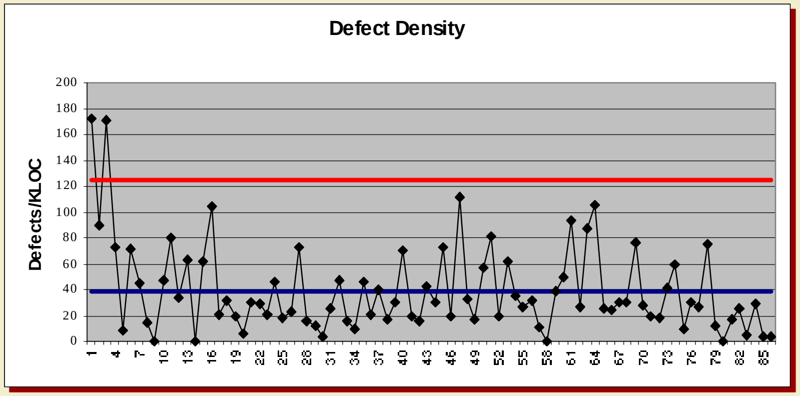


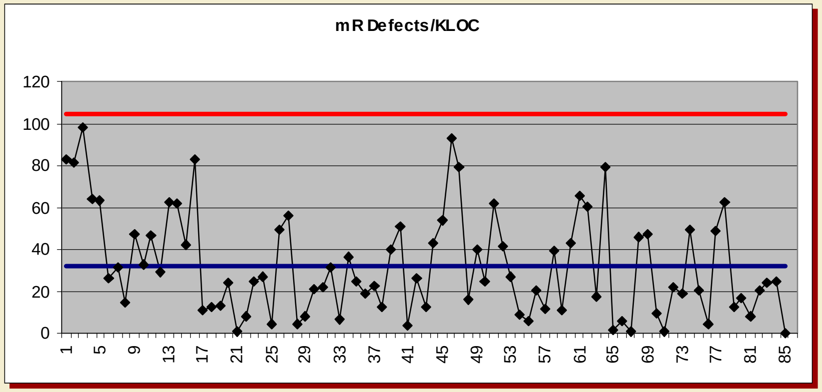
- Track process metrics:
  - rate vs yield

Open Loop Process XmR Charts









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

www.SoftwareSixSigma.com
(201) 947-0150, (201) 947-8828

A Control System Viewpoint

- The outputs of a process, y, are usually a function, f, of a set of control variables, x, and include a process noise component ε:

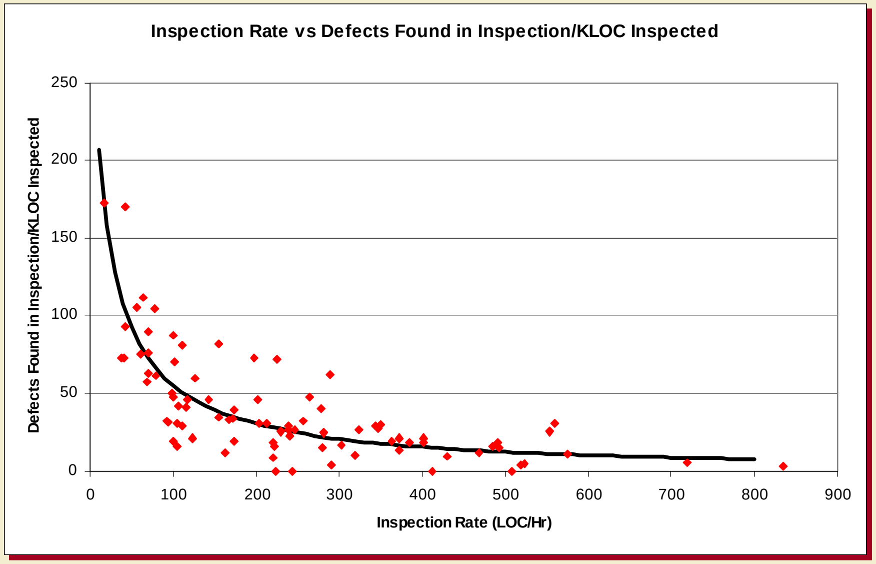
$$y = f(x) + \varepsilon$$

  - The y's are not directly controllable, but they can be controlled by the directly controllable x's.
  - Statistical measurements are necessary to avoid re-acting to the noise ε

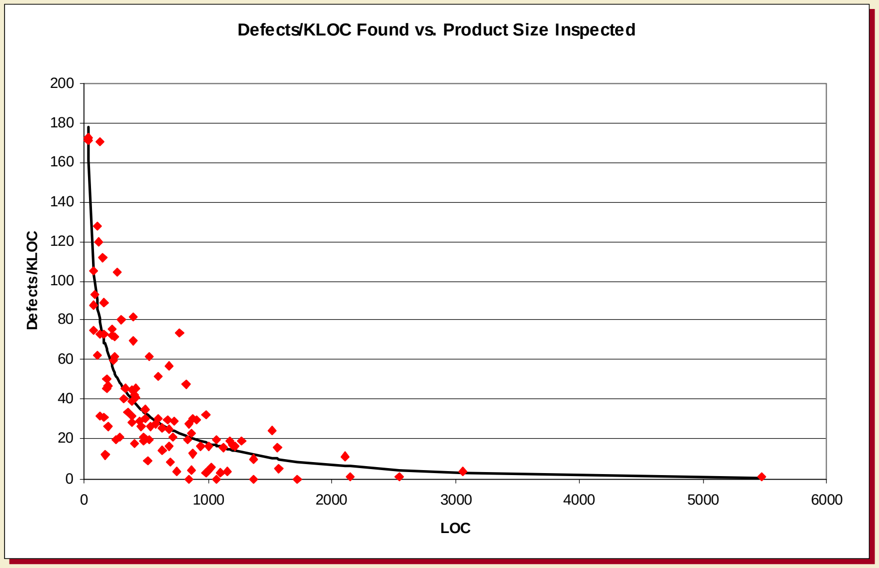
- Ideally we would like software inspection process that acts like a responsive, "closed loop" control system driving the x's to planned values and through their relationship to the y's, achieving overall product goals

Our experience has shown that review rate is the x that drives the inspection yield

Correlation Analysis

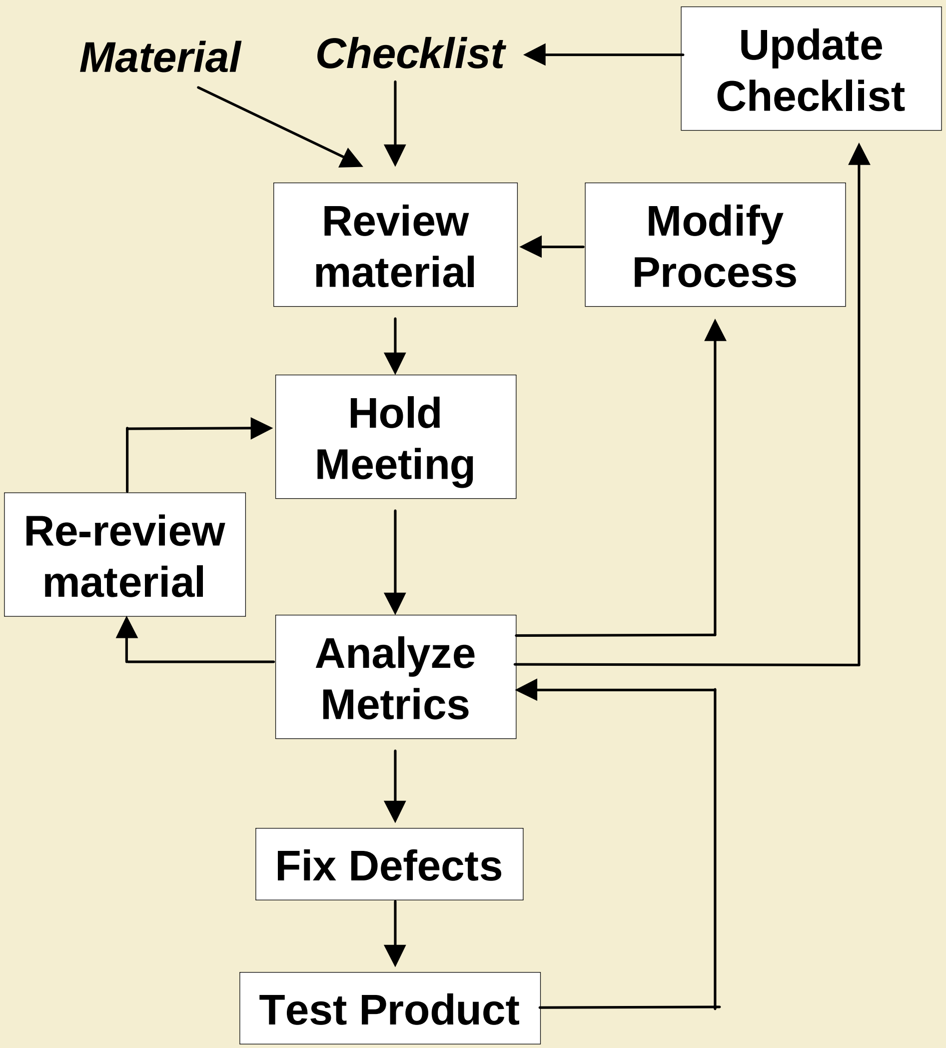


- To evaluate review rate for suitability as a control variable use correlation analysis
- $r^2 = 0.67$ – moderately good fit by hyperbola: $y = 1000/(0.1x + 3)$
- Chart suggests targeting review rate in the 100 – 200 LOCs hour range



- 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

Closed Loop Inspection Process



<u>Update Checklist</u>

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

<u>Modify Process</u>

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

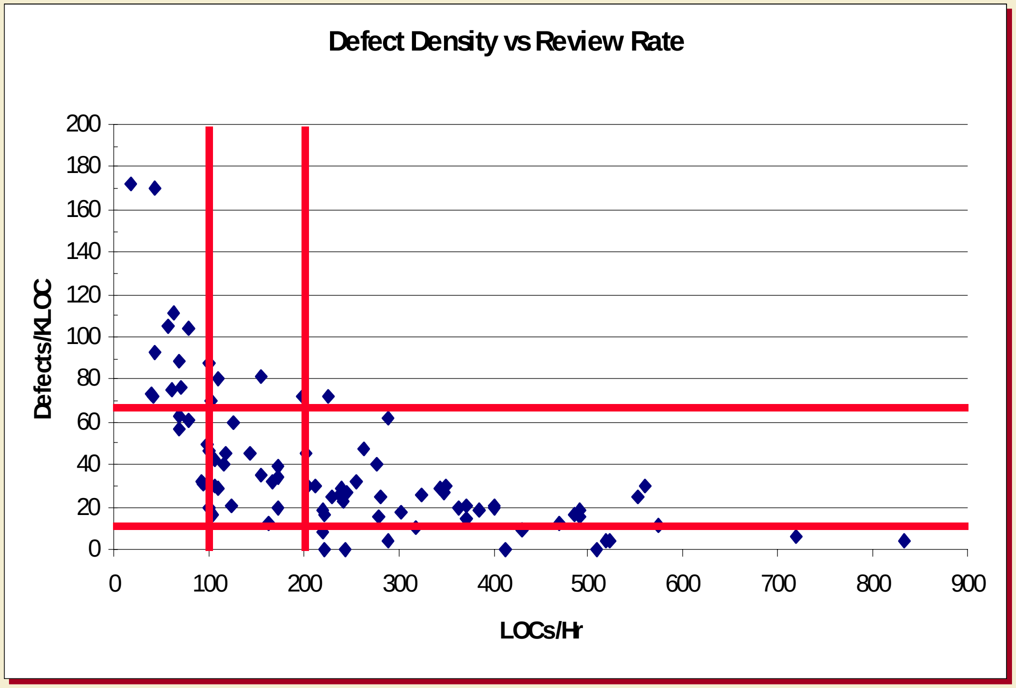
<u>Analyze Metrics</u>

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

Inspection Action Plan

<u>Slow Review Rate & Many Defects</u>
Is the product really buggy?
Was the review really effective?
Was the review cost efficient?

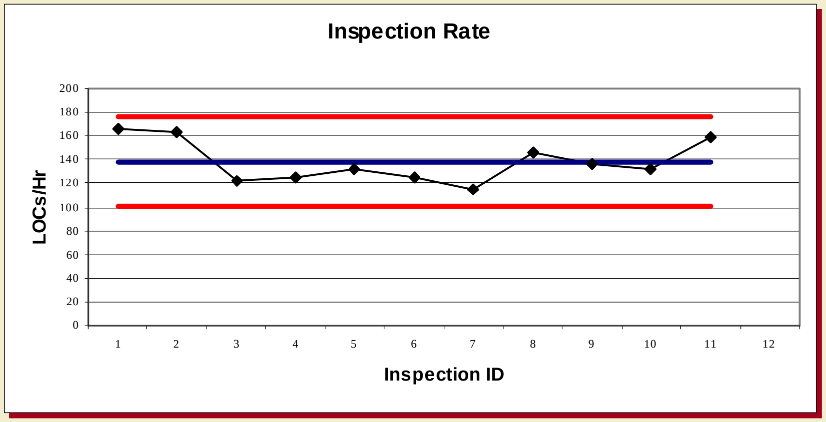
<u>Fast Review Rate & Many Defects => Buggy Product</u>
The product ***<u>IS</u>*** buggy.
Return to author for rework
Ask someone else to rewrite

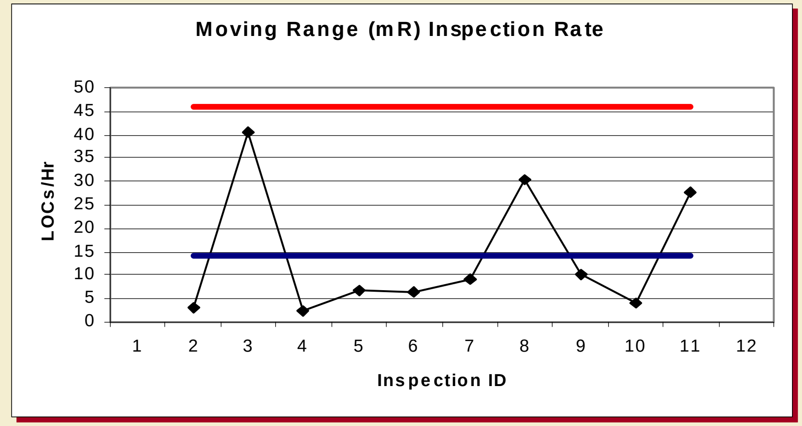


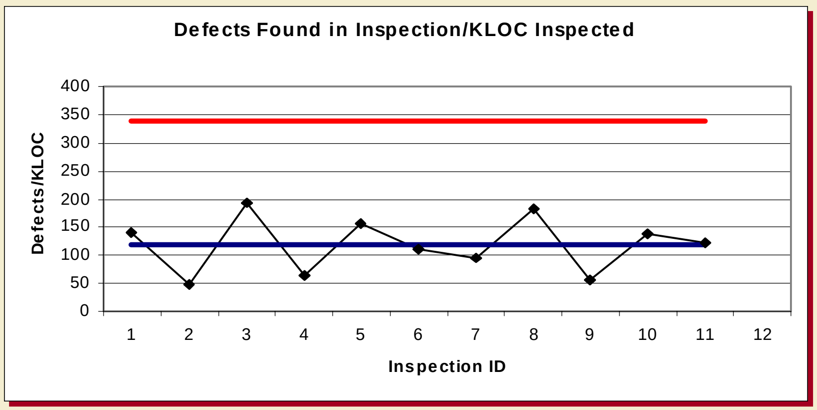
<u>Slow Review Rate & Few Defects</u>
Is the product really good?
Was the review really ineffective?
Was the review cost efficient?

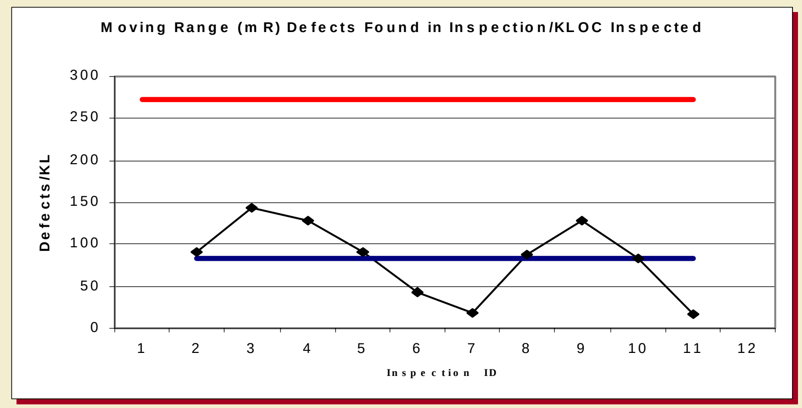
<u>Fast Review Rate & Few Defects => Poor Review</u>
~~Is the product really good?~~ *(can't tell !)*
Re-review at a slower rate
Make sure reviewers are using the checklist

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 - a 3.5x improvement in quality!**
  - **Average Defect Removal Rate 15/hr - a 2.5x improvement in removal cost!**

www.SoftwareSixSigma.com
(201) 947-0150, (201) 947-8828

Optimization Strategy

- Personal reviews performed prior to team inspections
  - Remove all the errors the author can detect at the lowest possible inspection cost
  - Checklist derived from author's own list of compilation and test defects flags high risk areas where author has a history of making mistakes
- Frequent short team inspections
  - Checklists focus on interface and requirements related issues that can't easily be found in the personal review
  - Small teams that include the internal "customers" for the product
  - Focus on a few hundred lines of code at a time
- Periodic Defect Prevention meetings provided the development team with an opportunity to review their data and define approaches to detect defects earlier or prevent or prevent them entirely
- Defect prone products "pulled" from integration and test and re-inspected

Goal: Minimize review cost while maximizing yield

Optimization Strategy Advantages

- Doesn’t waste team’s time with defects the author can easily find
- By inspecting a few hundred lines at a time, preparation time required is on the order of an hour
- Reviewers can stay focused and inspection can be held on the same day that product is available
- Eliminates lags, removes the temptation for the author to move forward into test before the review takes place
- Entire cycle can take as little as 2 – 3 hours from product availability to end of inspection
- Developers use their own data for defect prevention
  - Eliminates handoffs

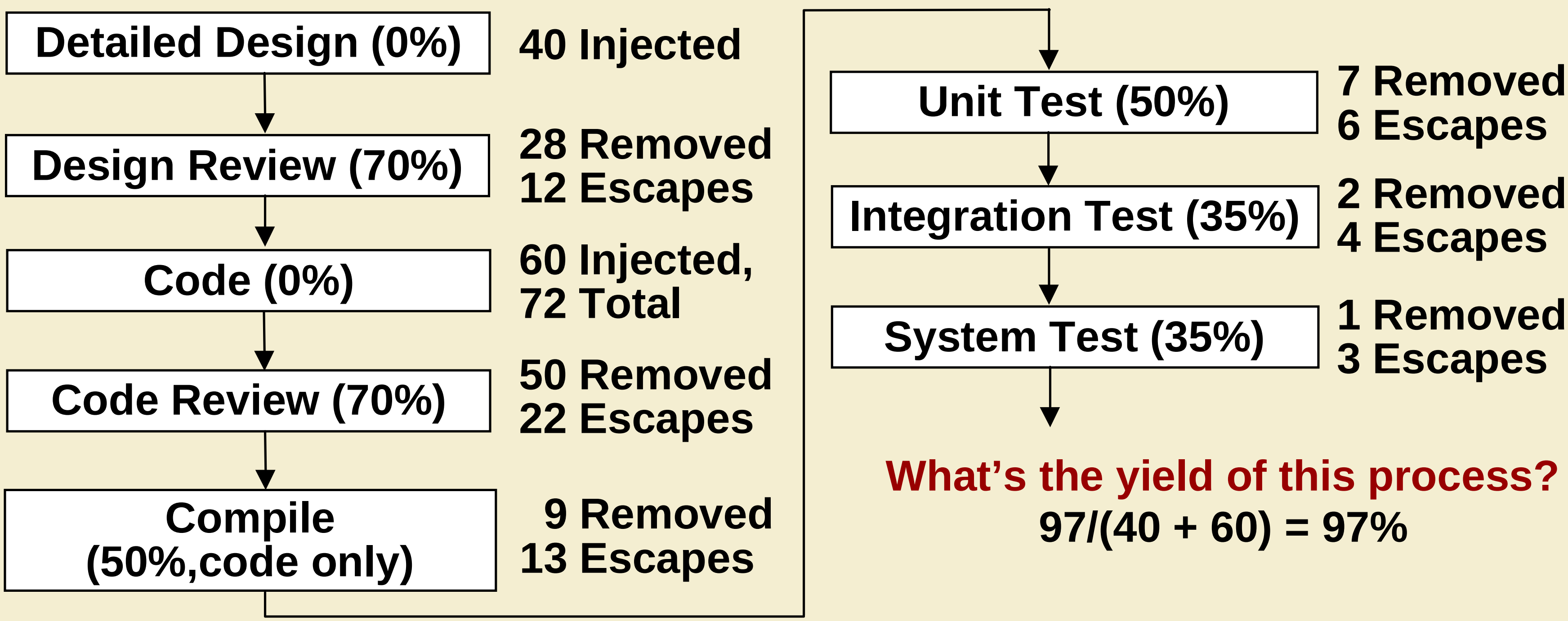
Defect Prevention

- Defect Prevention can be implemented by an organization that is performing inspections and collecting defect data.
- A Defect Prevention team sets and manages to their own goal.
- They use their own defect data, captured during inspections.
- Defects are analyzed using pareto charts to identify most expensive, most frequent, etc.
- Actions are taken to prevent a targeted defect type from occurring in the future.
  - Modify checklists, change coding and design standards
- The team members convince themselves of the value of the activity by calculating their own ROI.
- Lessons Learned are shared with other Defect Prevention teams on a periodic basis.

Data must be regularly used by the people collecting it, otherwise they will stop collecting it!

Yields and Quality Planning and Management

- Inspection process can be characterized by its yield
- Historical yields permit planning the number of defects that will be removed
- Manage to the plan by taking corrective action when actual values diverge from plan



What's the yield of this process?

97/(40 + 60) = 97%

Calculating Return on Investment - 1

- Costs can be directly measured
  - training, tools, performing the inspections
- The dominant costs are the inspection prep and the meeting time
- Savings require estimating the difference in cost between finding a defect in review and finding it later in the process

	Defects leaked from prev phase	New Defects Injected	Phase Yield	Defects Contained	Defects Leaked	Defect Removal Cost	Total Removal Cost (hrs)
Design	0.0	40	0%	0.0	40.0	n/a	0.00
Design Bench Check	40.0	0	0%	0.0	40.0	10 mins	0.00
Design Inspection	40.0	0	0%	0.0	40.0	30 mins	0.00
Code	40.0	60	0%	0.0	100.0	n/a	0.00
Code Bench Check	100.0	0	0%	0.0	100.0	5 mins	0.00
Compile	100.0	0	50%	50.0	50.0	1 min	0.83
Code Inspection	50.0	0	0%	0.0	50.0	15 mins	0.00
Unit Test	50.0	0	50%	25.0	25.0	15 mins	6.25
Integration Test	25.0	0	35%	8.8	16.3	18 hrs	157.0
System Test	16.3	0	35%	5.7	10.6	18 hrs	102.0
							267

- Without inspections, the cost of defect removal is 267 hrs per KLOC

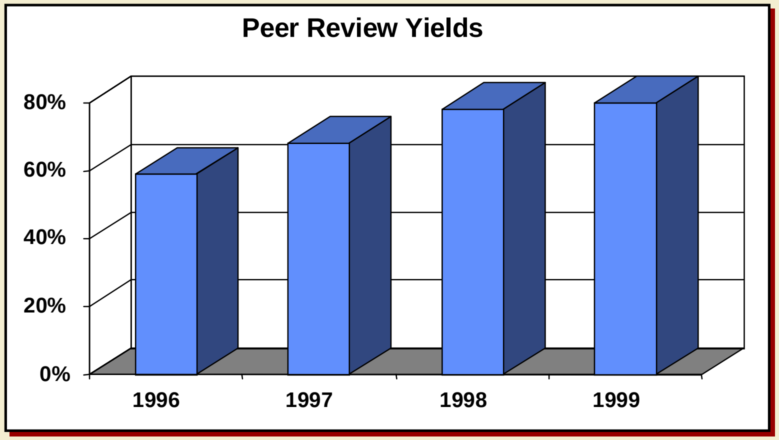
Calculating Return on Investment - 2

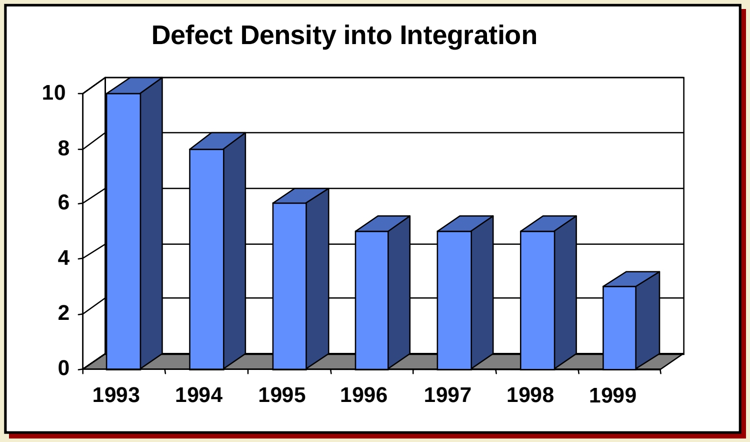
	Defects leaked from prev phase	New Defects Injected	Phase Yield	Defects Contained	Defects Leaked	Defect Removal Cost	Total Removal Cost (hrs)
Design	0.0	40	0%	0.0	40.0	n/a	0.00
Design Bench Check	40.0	0	50%	20.0	20.0	10 mins	3.33
Design Inspection	20.0	0	50%	10.0	10.0	30 mins	5.00
Code	10.0	60	0%	0.0	70.0	n/a	0.00
Code Bench Check	70.0	0	70%	49.0	21.0	5 mins	4.08
Compile	21.0	0	50%	10.5	10.5	1 min	0.18
Code Inspection	10.5	0	60%	6.3	4.2	15 mins	1.58
Unit Test	4.2	0	50%	2.1	2.1	15 mins	0.53
Integration Test	2.1	0	35%	0.7	1.4	18 hrs	13.23
System Test	1.4	0	35%	0.5	0.9	18 hrs	8.60
							37

- Without inspections,
  - The cost of defect removal was 267 hours.
- With inspections,
  - The cost of holding the inspections is about 40 hours (at 200 LOC/hr)
  - The cost of defect removal drops to 37 hours.
- The net savings is 267 – (40+37) = 190 hours

Results

- Over a period of 5 years, we gradually implemented the strategies described
- As Peer Review yields increased from 60% to 80% and we introduced personal reviews, defects into integration were reduced from 10/KLOC to 3/KLOC
- At the same time, cost of performing peer reviews decreased by 40% as we reduced the size of the inspection teams





The organization realized a net improvement of 190 hrs / KLOC!

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

Watch for our article on "*Optimizing Software Inspections*" in the December 2003 issue of *Software Quality Professional*.

For additional information visit our web site or contact us at:

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

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

www.SoftwareSixSigma.com