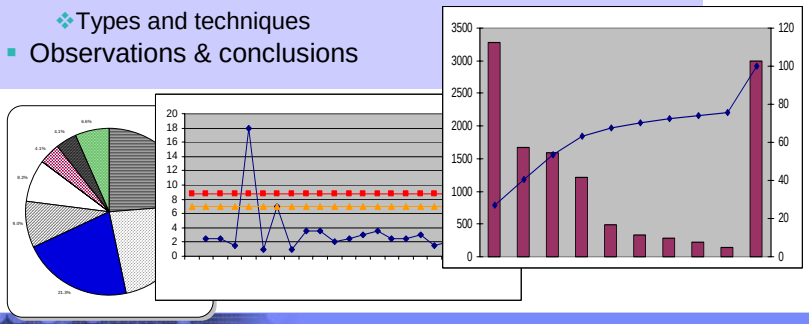
SPC in Software Development? .....Innovation Needed!

10th Annual PSM Users' Group Conference
July 25, 2006
Diane Manlove & Steve Kan
IBM System i, Rochester, MN

Agenda

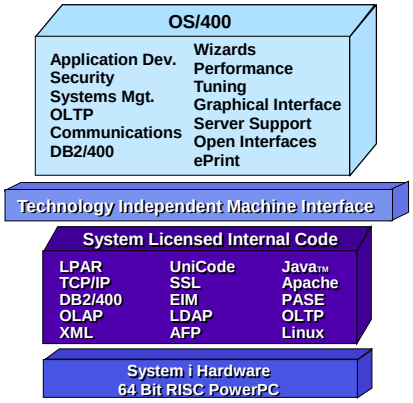
- IBM System i software development, quality, and in-process metrics
- Challenges in applying SPC (Statistical Process Control) in software development
- SPC application (and invention) in System i software development
  - Examples by development phase
  - Types and techniques
- Observations & conclusions



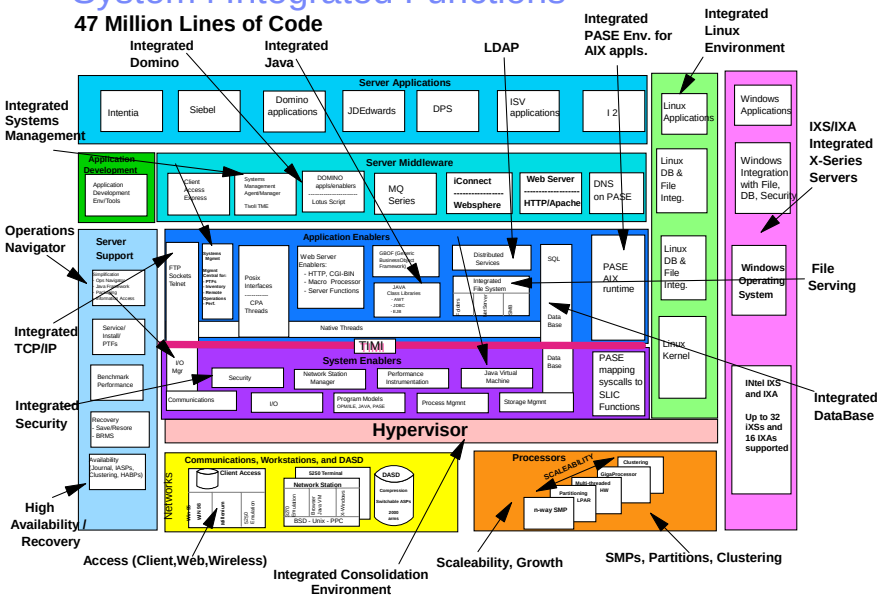
Background

- Data in SPC examples is from System i operating system releases
- Development lab of 1,400 employees
- A typical release is 3.0-3.5MLOC (million lines of code)
- Supports 51 national languages & over 300,000 customers
- Integrated system architecture

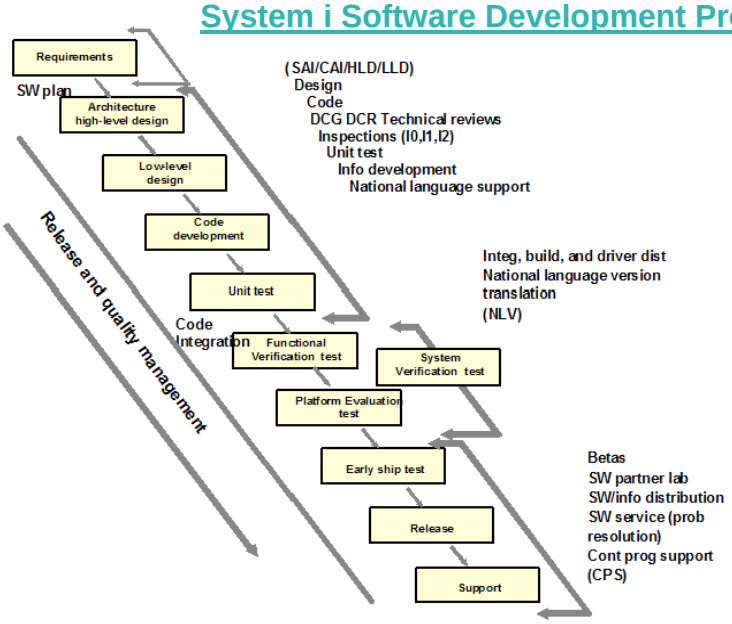
System i



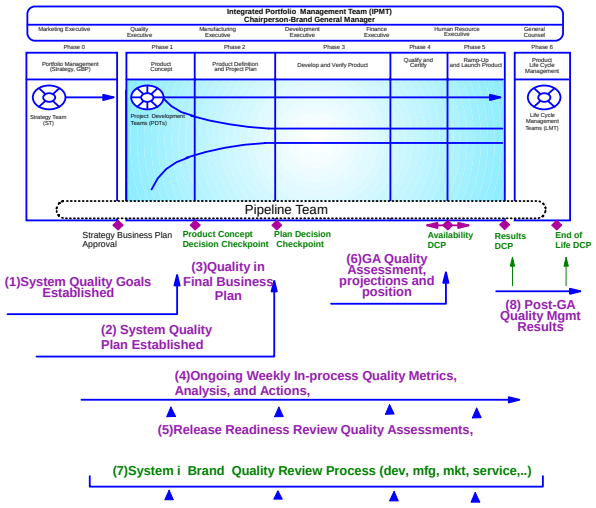
System i Integrated Functions



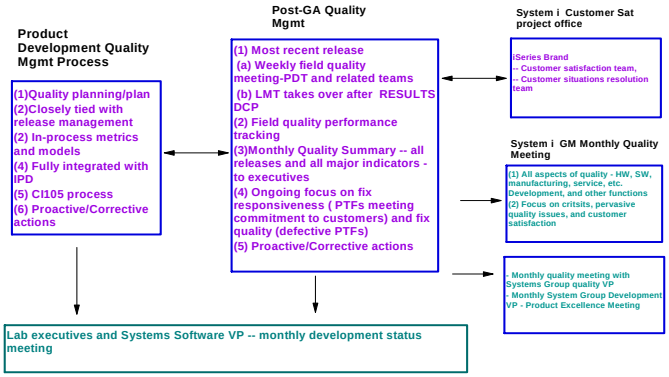
System i Software Development Process



System i Software Development Quality Mgmt Process



System i Software Quality Management System



Selected quality metrics and indicators

Pre-GA indicators

- DCG review issues/data
- I0, I2 progress
- Defects arrival rate and pattern
- Testing progress
- Defect Severity distribution
- Defect backlog
- Integration/build and driver probs
- Error-prone component analysis
- Critical problem list
- SVT System crashes & CPU utilization
- PET (Patform Integration test) stability (Mean time to IPL)
- Pending PTFs
- Early programs feedback
- Reliability growth models to estimate field defects
- etc.

GA quality assessment

Post-GA Indicators

- Field Defect arrival rate and pattern
- LUG customers availability/downtime
- Fix quality - defective PTFs
- Fix responsiveness - meeting committed dates to customers
- Customer satisfaction
- Customer complaints
- Supportline satisfaction
- etc.

Focus on release-to-release improvement and Statistical Quality Control

Why SPC?

- Aid metrics interpretation
  - Quantify process capabilities
  - Quickly identifying process deviations
- Facilitate planning and target setting
- Control product quality
- Evaluate process changes

Software-unique SPC challenges

- Software is design and development, not manufacturing -- behavioral aspect of process implementation
- Significant sources of variability exist in software development -- difficult to control or minimize
- Software parameters follow the "phase" and "cycle" concept, different from continuous ongoing operations
- Assumptions for traditional control charts seldom met in software environment
- Process limits and the level of end product quality not as directly linked as in manufacturing operations -- the meaning of process capability
- Invention is required..."Pseudo-SPC"

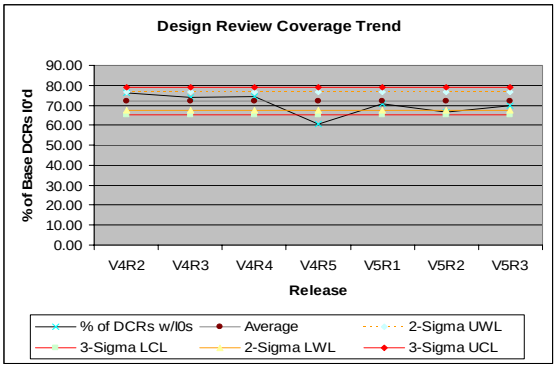
SPC in the Design Phase – Traditional Control Charts (with Careful Application)

Design Review Coverage = the number of design reviews held, expressed as the percentage of line items (requirements) for which formal design reviews are conducted

P-Chart Formula:
- UCL = p+3*Sqrt[p*(1-p)/n]
- LCL = p-3*Sqrt[p*(1-p)/n]
- UWL = p+2*Sqrt[p*(1-p)/n]
- LWL = p-2*Sqrt[p*(1-p)/n]

Where:
p=process average,
n=average sample size,
Sqrt=Square Root

UCL=Upper Control Limit
LCL=Lower Control Limit
UWL=Upper Warning Limit
LWL=Lower Warning Limit



SPC in the Design Phase

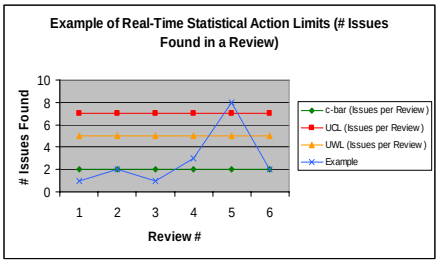
Real-Time control chart for the design phase = The number of issues found during a review using action limits based on a standard c-chart.

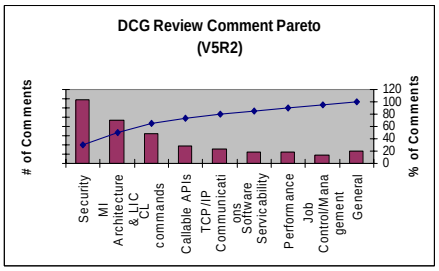
Limitation: c-chart assumes material reviewed is of constant size.

C-Chart Formula:
- UCL= c+3*Sqrt(c)
- LCL= c-3*Sqrt(c)
- UWL = c+2*Sqrt(c)
- LWL = c-2*Sqrt(c)

Where: c = process average number of non-conformities

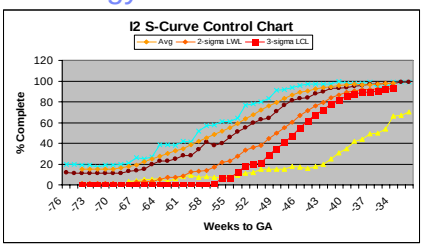
Note: Augment charts with other quality tools such as Pareto charts.

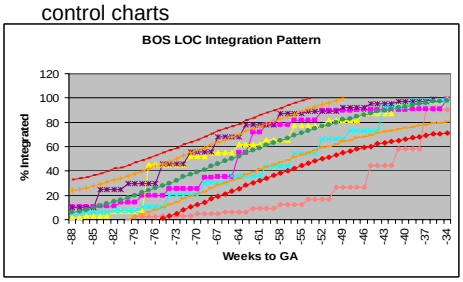


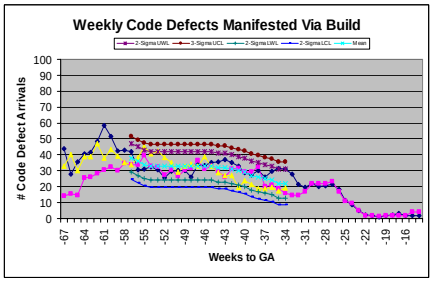


SPC in the Code Phase – Adaptations of Control Chart Methodology

- Pseudo-control charts based on polynomial models
- $y=a+b*x+c*x^2+d*x^3+e*x^4$ (based on historical data)
- estimate of the standard deviation is obtained from the analysis of variance of the model, using the mean square residual to estimate the variance
- UCL = y+3*s, where s=standard deviation
- Limitations: Not as sensitive as traditional control charts





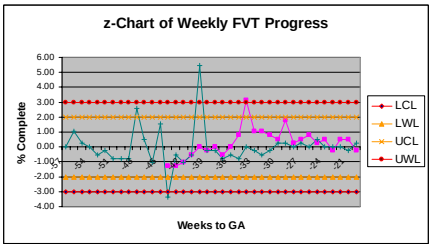


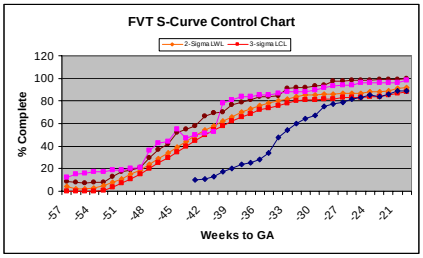
SPC in the Test Phase – Creative Approaches

- Use a z-transformation and z-chart to plot weekly progress (would work with code phase indicators shown as well)
- Use the polynomial approach for a more intuitive view
- Z-Chart conforms more closely to traditional control charts

  Z=(x-u)/s, where u=mean, s=standard deviation

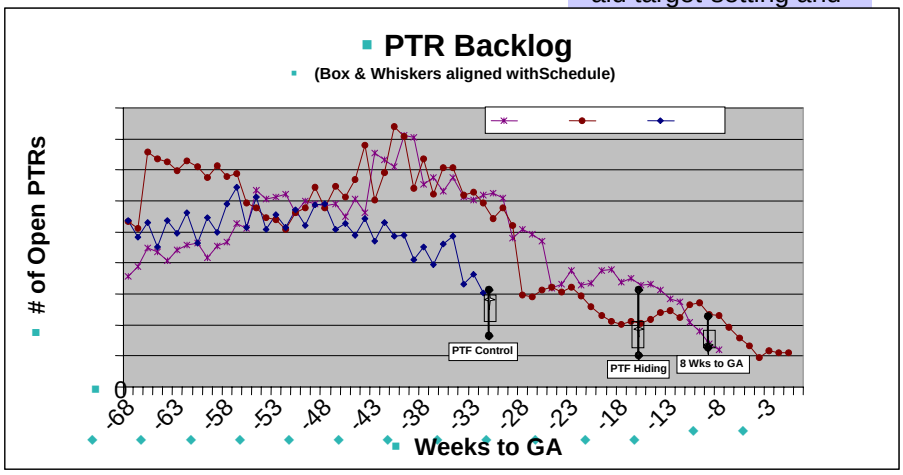
- Best approach is the 2 charts in combination





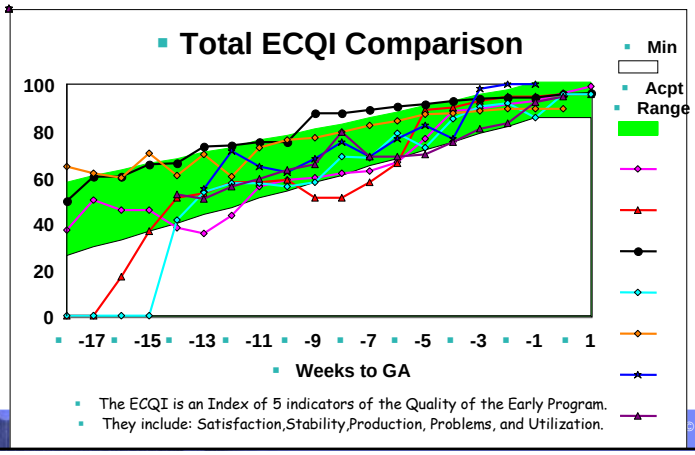
SPC in the Test Phase

Box-and-Whisker charts aid target-setting and



SPC in the Beta Phase

- The Early Customer Quality Index (ECQI) in an index combining 5 Beta Program indicators: Satisfaction, Stability, Production, Problems, and Utilization
- ECQI limits were established based on historical data & have been in use for many years. Recent analysis found that they align well with SPC-based limits.



Software-SPC Observations & Conclusions

- Real-time / active feedback loops are not always achievable
- A 4.5% false alarm rate (2-sigma limits) may not have the same meaning for software
- Augment software control charts by displaying the data using other quality tools and looking at other indicators
- Test new charts against historical data, pilot new charts before introduction

- SPC or pseudo-SPC tools are the right direction
- More study, invention, and validation are needed