Ritmico Progress, Rayney Wong

Developing
Process Performance Baselines
Process Performance Objectives
Process Performance Models

About the Presentation

- About how a few companies at high maturity developed their PPB, PPO, PPM to meet business goals.
- The companies performed project based software development.
- Each company only has one type of methodology and life-cycle:
  - Iterative (Agile) or Waterfall.

About the Presentation

- About how they took on a path that made high maturity acceptable by the staff.

Ritmico Progress, Rayney Wong

- Ritmico Progress is led by Rayney Wong who is a SCAMPI High Maturity lead appraiser, and a CMMI Introduction instructor. Ritmico Progress is a SEI Agreement Partner for the CMMI Product Suite and is a registered company in Singapore.
- Rayney has over 20 years of software development and project management experience, ranging from radar communication systems, network systems, to publishing printer drivers and windows applications, and developing common coherent processes shared by offsite development centers.
- Rayney's experience includes high maturity knowledge in developing models and Statistical process control toolkits, developing business strategic initiatives and staff development activities to achieve business goals, and training in implementing process improvements and software development. Companies have grown from 50 to over 500 people under Rayney's guidance.
- Rayney@RitmicoProgress.com

- Since 1987 NashLabs® has helped Clients achieve a strategic advantage in the production of world-class software. We're focused on the measurement and improvement of software processes that work in the real world.
- Nash Laboratories® is a Partner of the Software Engineering Institute at Carnegie-Mellon. As a Partner, the company is licensed to provide the latest generation of SEI technologies:
  - Introduction to the CMMI®
  - Introduction to the People-CMM®
  - SCAMPI℠ High-Maturity Appraisals
  - CMMI® Process Consulting
  - Six Sigma Training and Consulting
- Rayney is an Associate with NashLabs®.

® CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

BEIJING NTT DATA SYSTEMS INTEGRATION CO., LTD.

Founded in October 1, 1998

Full Name: BEIJING NTT DATA SYSTEMS INTEGRATION CO., LTD.

Location: BEIJING, CHINA, Headquarters

Number of employees： 640

The main business

- Off-shoring Software Development for JAPAN
- System integration for Domestic business of CHINA
- Business support for Domestic business of CHINA

Offshore development base in:

- BEIJING, SHANGHAI, TIANJIN
- Beijing NTTDATA JAPAN: Sales/SE Dispatch etc

Main skills:

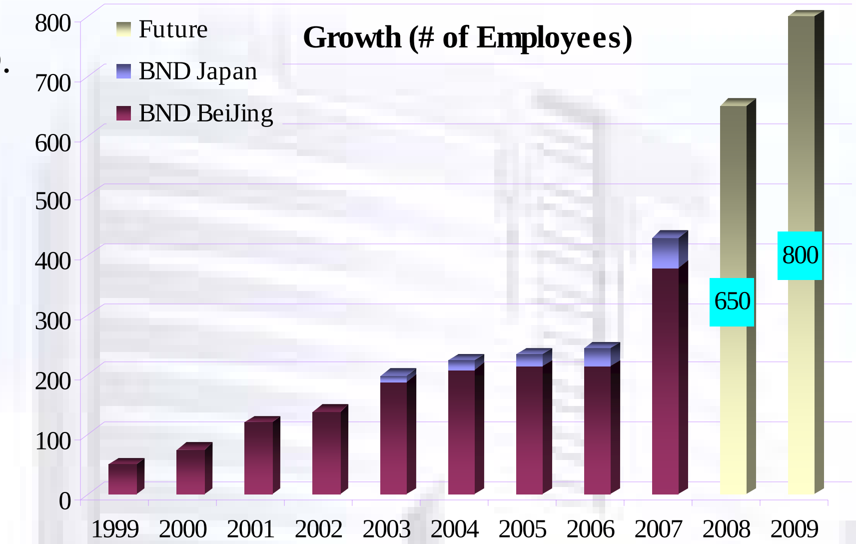
- Skill is widely distributed that covers open system trends
- Acquisition of qualified skills: Oracle, MS systems, PMP

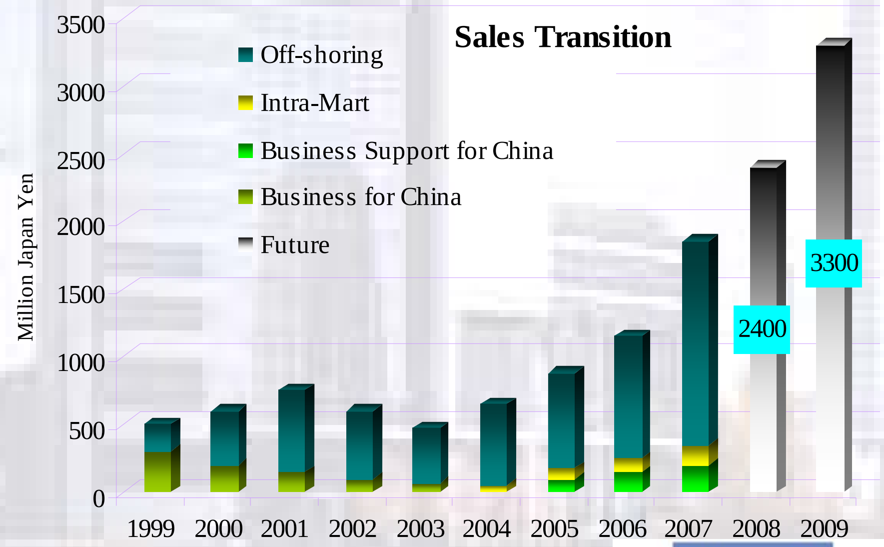
Project Management & Security – CMMI and ISO27001

Image of the future: Current: off-shoring Software Development business.

Future:

- High Level off-shoring Software Development business.
- Service for the advance of Japanese Company into China Market
- Domestic business of CHINA
- Roll out Business for European and American enterprise





北京NTT DATA系統集成有限公司

VanceInfo Technologies Inc.

Founded in 1995 - 13 year track record of working with global companies

Full Name: VanceInfo Technologies Inc.

Location: Beijing , Headquarter

NYSE: VIT First China based Outsourcing firm listed in US markets

Over 4500 diverse employees: 4412 developers

Substantial Global Footprint
USA (New York, Seattle, San Francisco),
- China (Beijing, Shanghai , Nanjing, Tianjin, Hangzhou, Xian, Dalian Chengdu, Shenzhen and Hong Kong)
- Singapore & Japan
- Australia (Melbourne)

Core capabilities
- IT Services for Fortune 1000 companies and SMEs
- Research & Development Services (Product Development)
- Infrastructure Services
- ITES/BPO

Domain knowledge & Vertical focus
- Banking Financial Services and Insurance (BFSI)
- Manufacturing & Retail & Distribution
- Telecom
- Technology

Centers of Excellence
- Microsoft & Java
- Enterprise Solutions: SAP, Oracle, PeopleSoft & Siebel
- Business Intelligence & Data Warehousing
- Messaging, EAI/B2Bi and SOA
- QA & Testing Services

Quality delivery — CMMI and ISO certified

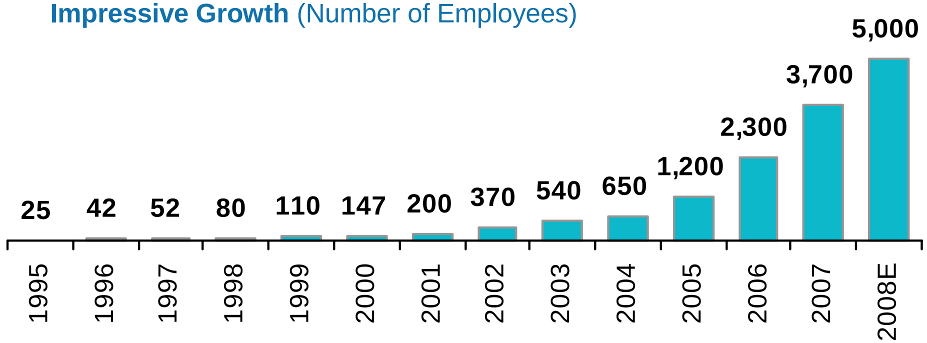
People Oriented Firm
- Management Team with global experience
- Voted "Top 100 Employers Most Favored by University Graduates"

Publicly Listed on NYSE

December 12, 2007

VIT LISTED NYSE | VanceInfo

Impressive Growth (Number of Employees)



Blue-Chip Customers

TIBCO, ORACLE | PeopleSoft, ABB, Cisco, IBM, adidas, 3M, citi, NRI, The People's Bank of China, SONY

Best-of-Breed Partners

ORACLE PARTNERNETWORK, SIEBEL, Microsoft CERTIFIED Partner, hp, IBM Core Service Provider

VanceInfo

Perficient China Ltd.

Facts and History

- Perficient's **G**lobal **D**elivery **C**enter was established in 2004
- 130+ consultants -- 200 by EOY 2008
- Located in Hangzhou - Silicon Valley of China
- All business in Perficient China is conducted in English
- Agile methodology delivering high priority requirements incrementally

China Global Delivery Center

Main Business

- Web Application and Portal Development
- Content Management Development
- CRM / Siebel Implementation
- SOA, Integration and Messaging Implementation
- BPM Implementation

Terminologies

PPB	Process-Performance Baselines A documented characterization of the actual results achieved by following a process, which is used as a benchmark for comparing actual process performance against expected process performance.
PPO	Quality and Process-Performance Objectives Objectives and requirements for product quality, service quality, and process performance. Process-performance objectives include quality; however, to emphasize the importance of quality in the CMMI Product Suite, the phrase quality and process-performance objectives is used rather than just process-performance objectives.
PPM	Process-Performance Models A description of the relationships among attributes of a process and its work products that is developed from historical process-performance data and calibrated using collected process and product measures from the project and that is used to predict results to be achieved by following a process.

From SEI CMMI v1.2

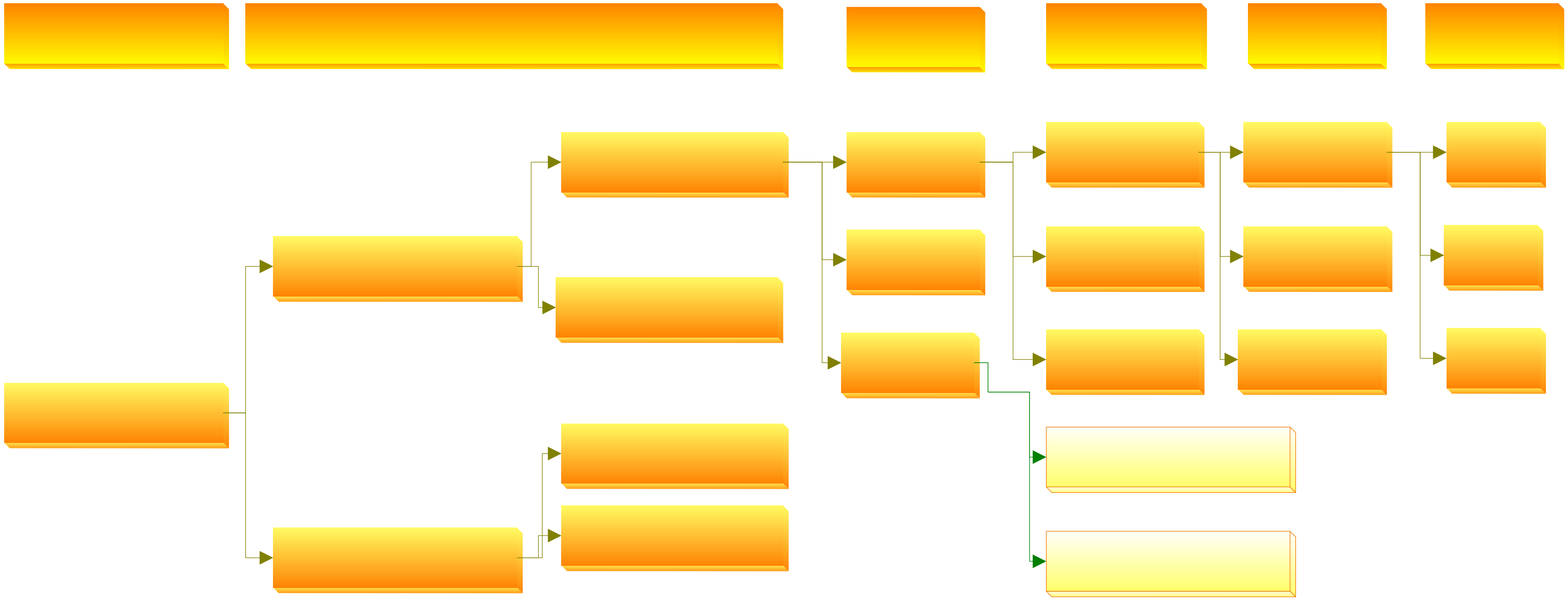
Terminologies

Base Measures	A distinct property or characteristic of an entity and the method for quantifying it. E.g.: ▪ Number of defects, ▪ Size of Module in KLoc (Thousand Lines of code)
Derived Measures	Data resulting from the mathematical function of two or more base measures. E.g.: ▪ Defect Density = (Number of Defects) / Module Size KLoc

From SEI CMMI v1.2

BGS, VOP-MAR

- Purpose of all improvements are derived from the Business Goals Strategy (BGS).



Copyright Rayney Wong

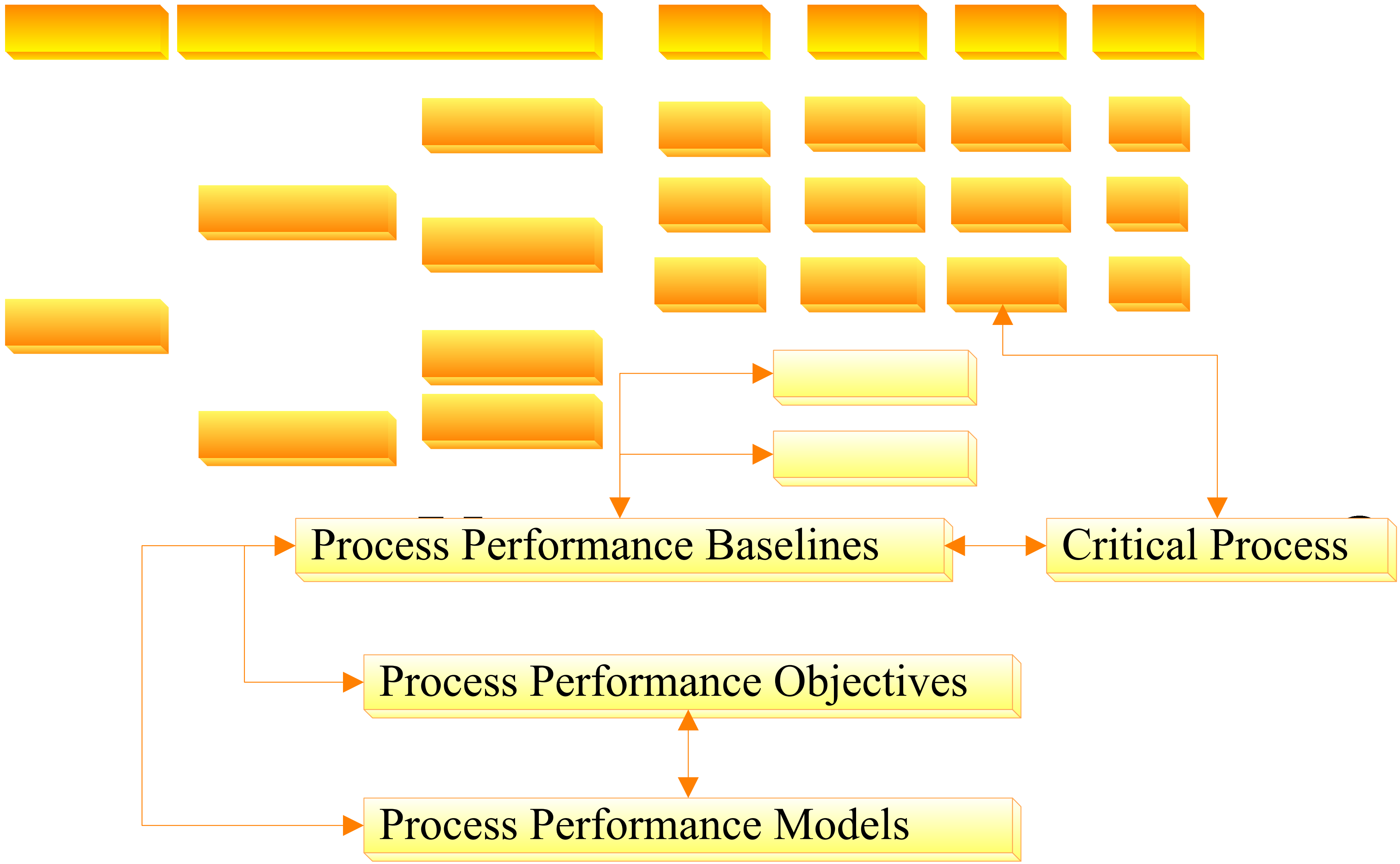
VOP-MAR

- A BGS exercise typically takes up a period of several weeks and is performed annually.

1	**V**ision	Realizing and understanding the **v**ision, breaking the vision down into its constituent parts.
2	**O**bjectives	Developing and prioritizing the goals and **o**bjectives that must be achieved to fulfill each part of the vision.
3	**P**roblems	Identifying and analyzing the **p**roblems and root causes that are preventing us from reaching the goals, objectives, and vision.
4	**M**easures	Determining the **m**easures to understand the extent of the problems and target measures to meet the objectives.
5	**A**ctions	Developing the **a**ctions for resolving the problems and reaching the goals. Improvements are aligned towards the objectives, vision and goals.
6	**R**isks	Considering the side effects and costs of the actions in order to mitigate **r**isks and side effects caused by the actions.

Copyright Rayney Wong

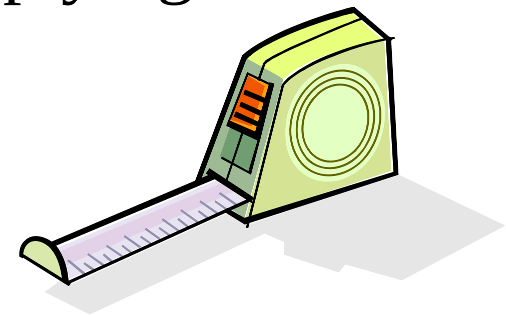
BGS at High Maturity



Copyright Rayney Wong

About the Measures in this Presentation

- Measures were from one of the companies.
- Unit Testing of software modules with Test Cases.
- Unit testing is performed after source codes have been reviewed:
  - Co-worker cross-check review of all source codes
  - Peer Review of critical module's source codes
- Measures have been adjusted by multiplying with factors as true measures cannot be shown.



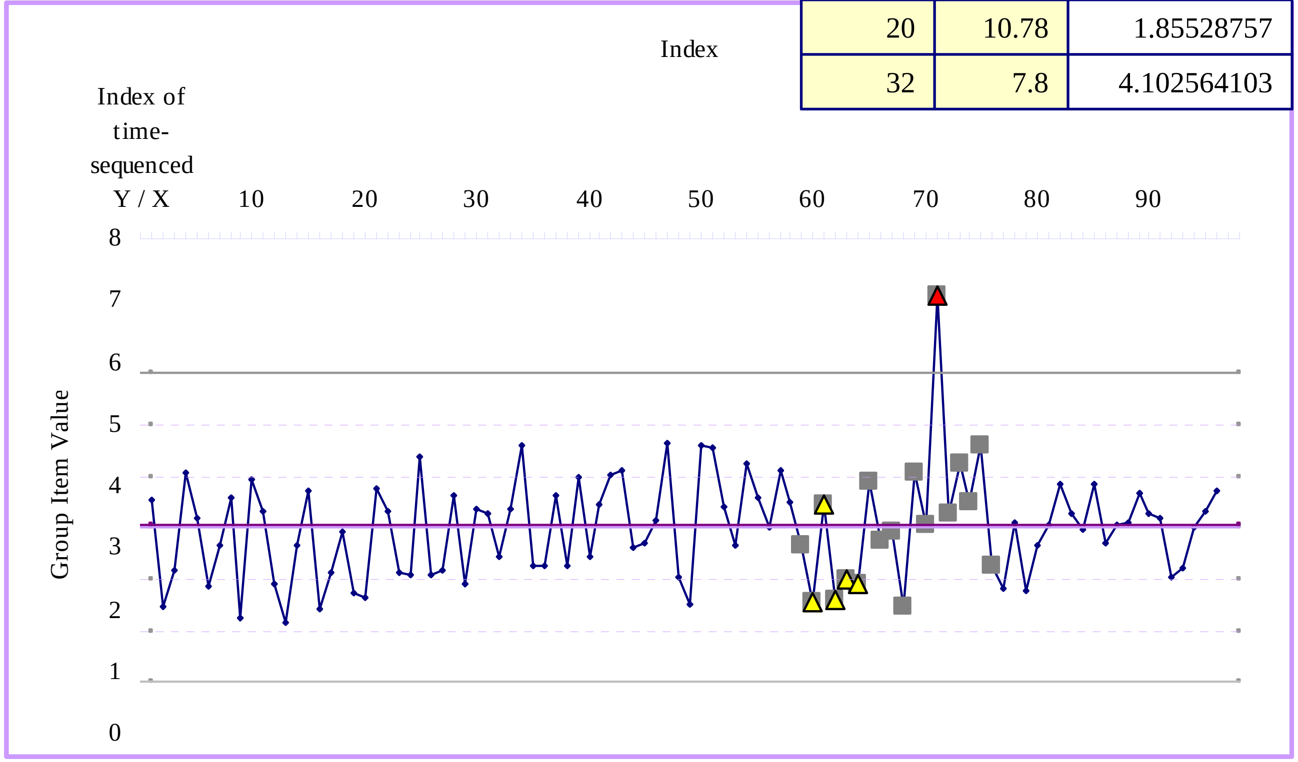
PPB – Define the derived measures (part of BGS)

- Unit Testing of software modules base measures:
  - #Defects found by the developer during unit testing of his module.
  - Module code size in KLoc.
  - #Test cases used to unit test the module.
  - Total time in hours taken to test the module using the test cases.
- Possible PPBs that can be derived:
  - Defect Density = #Defects / Size KLoc
  - Test Case Density = #Test cases / Size KLoc
  - Test Speed = #Test cases / Testing time

PPB – Perform Statistical Analysis

- Defect Density for Unit Testing
  - XmR or ImR requires time-sequenced data
  - #Defects/Size KLoc
- PPB:
  - UCL = 5.828
  - LCL = 0.833

# Defects	Code Size KLOC	# Defects / Code Size KLOC
59	15.6	3.782051282
57	27.8	2.050359712
54	20.4	2.647058824
77	18.2	4.230769231
84	24	3.5
18	7.6	2.368421053
56	18.4	3.043478261
95	25	3.8
20	10.78	1.85528757
32	7.8	4.102564103



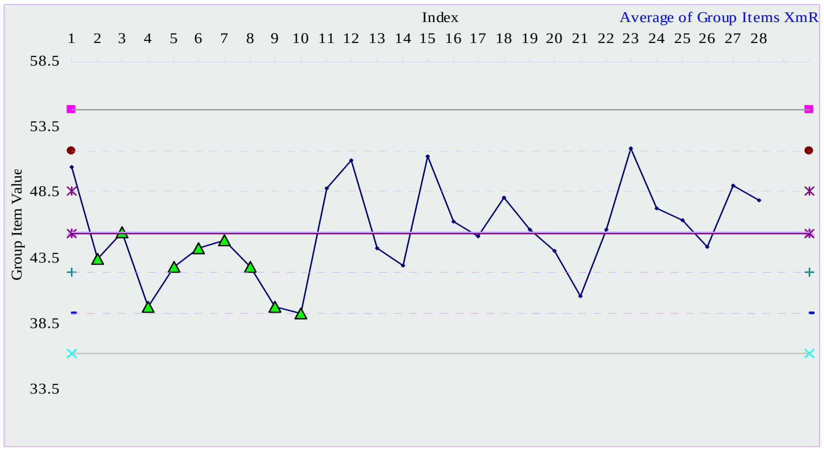
PPB – When to Develop?

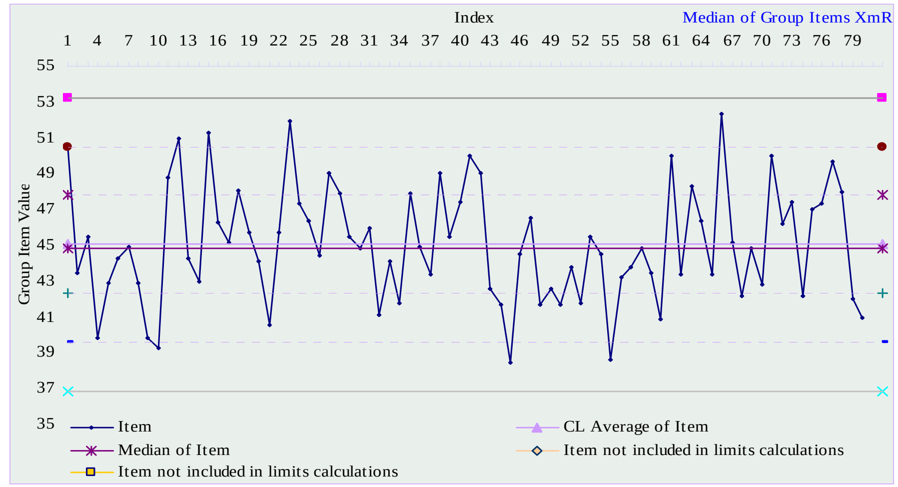
- Data are added into the XmR control charts as soon as each Unit Testing of a module is performed.
- How many data points before we can use the control charts?

XmR requires time-sequenced data.

X-Bar does not unless time-sequenced tests are performed.

False Alarms



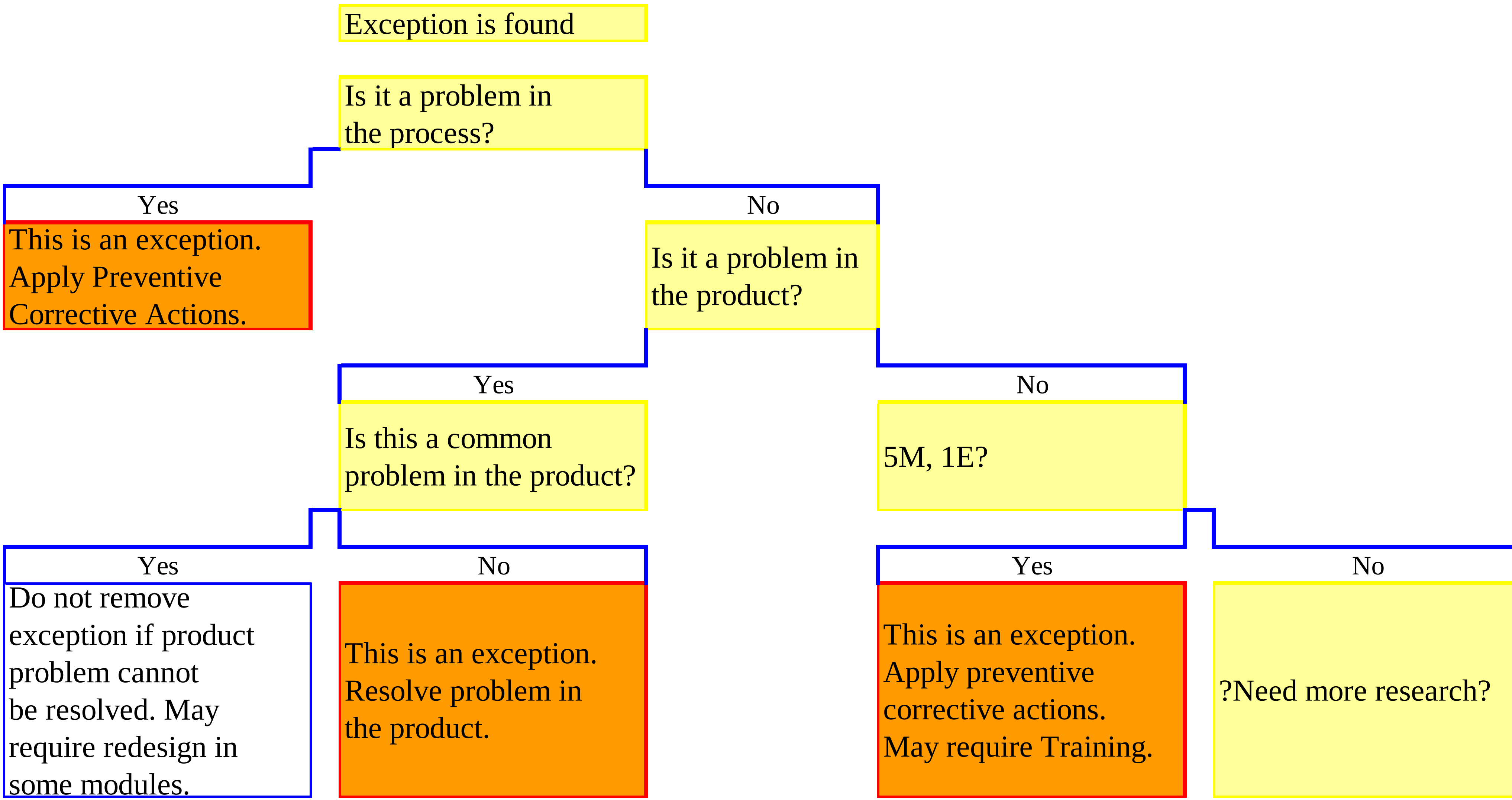


Drive with care. Small changes at a time.

Data shown are not from the organization.

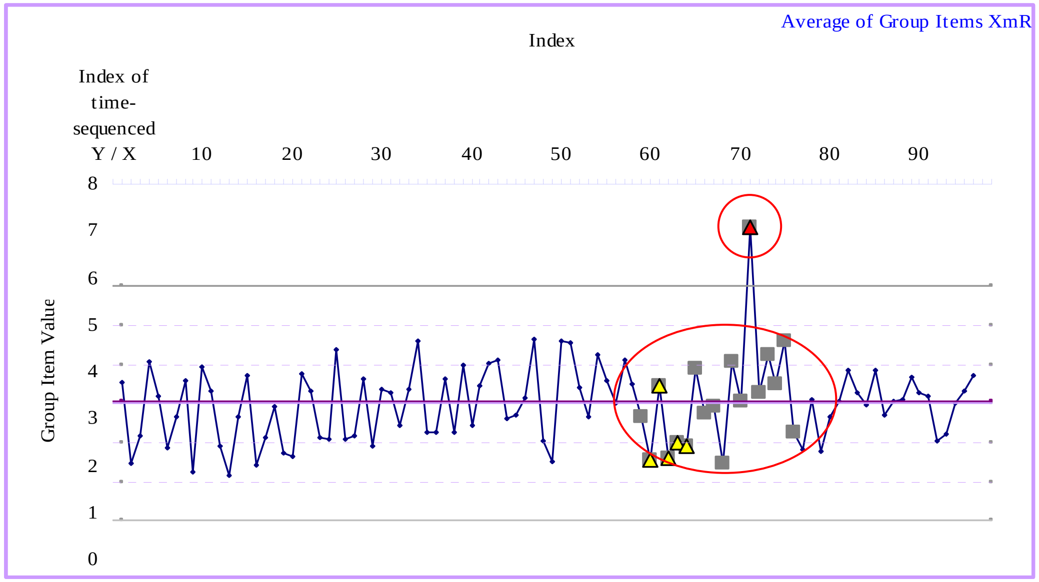
For illustration purpose only.

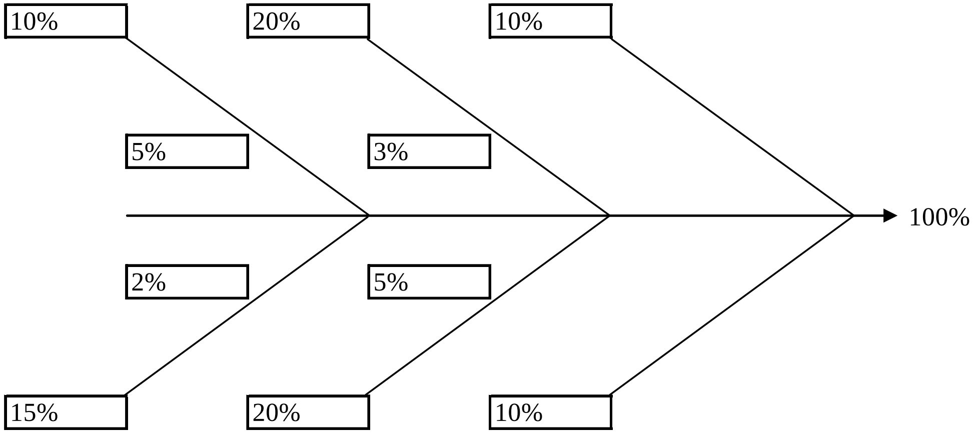
Can Exception be removed?



PPB → PPB'

- For each exception or set of exceptions, perform a problem solving process to consider improvements to prevent them.

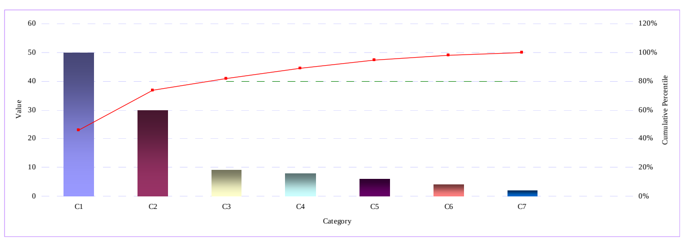


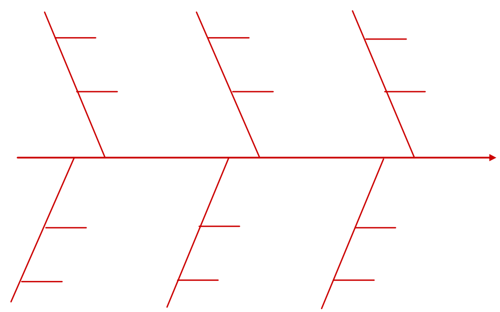


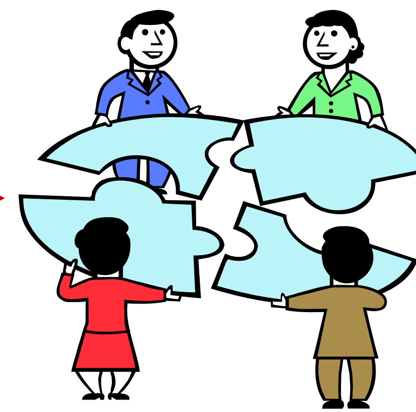
Quantitative Fish-Bone Diagram

PPB → PPB’

- Problem Solving Process must be done carefully to ensure improvements are able to prevent the exceptions.
- Problem Solving Process are performed by the practitioners with guidance from the EPG.
- Only remove the exceptions if there are improvements to prevent them.

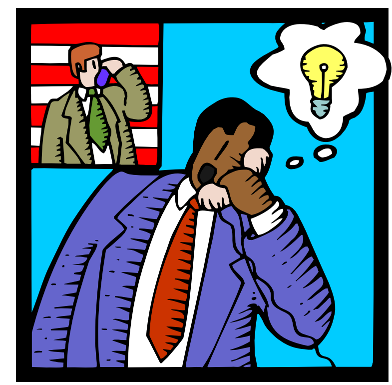






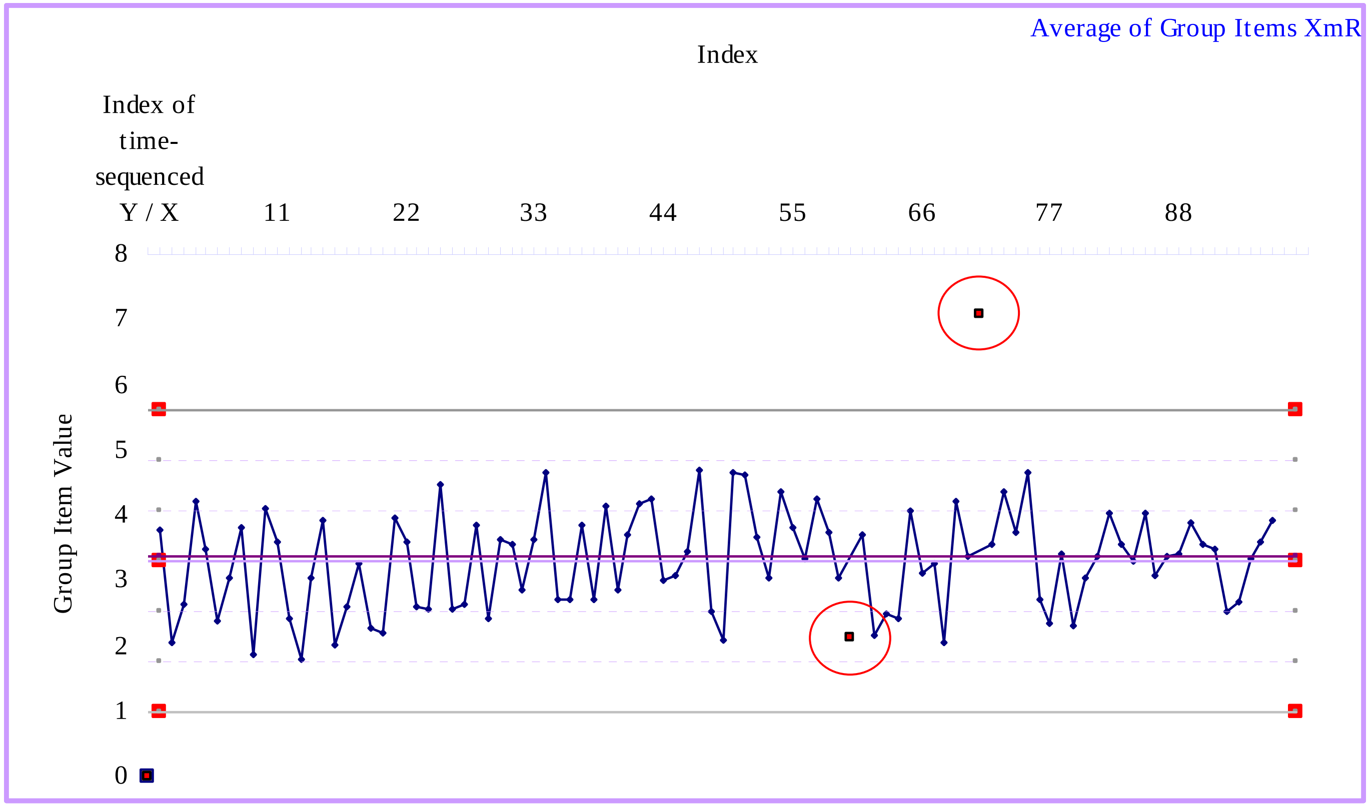
PPB → PPB’

- PPB’ is the improved PPB that the project may achieve after applying the improvements.
- Processes, templates, checklists, training must be updated so that improvements permeate across the organization and become institutionalized.
  - With Pilot projects to confirm improvements.



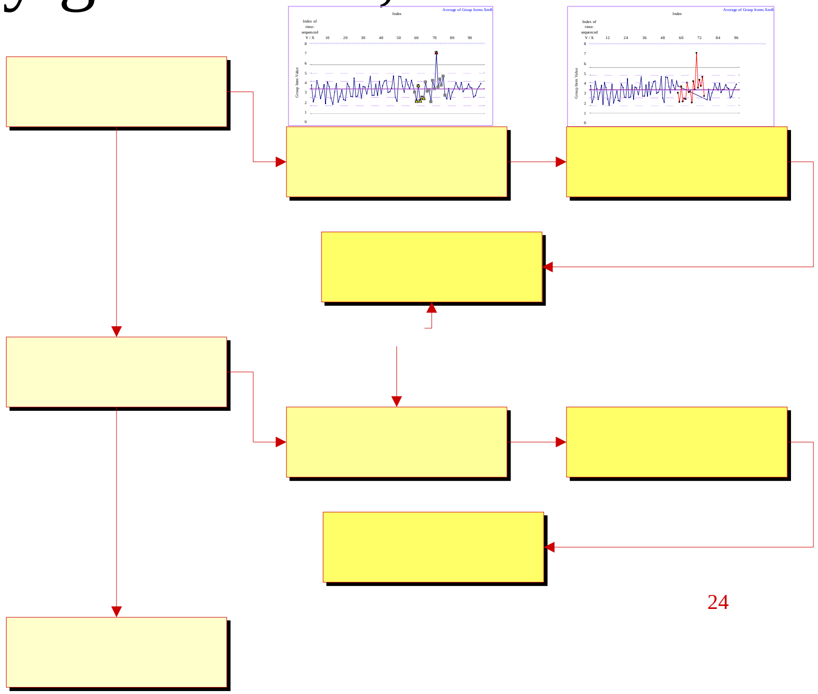
PPB’

- PPB’ of UT Defect Density (#Defects/Size KLoc)
  - UCL = 5.601
  - LCL = 1.005
- PPB earlier was:
  - UCL = 5.828
  - LCL = 0.833



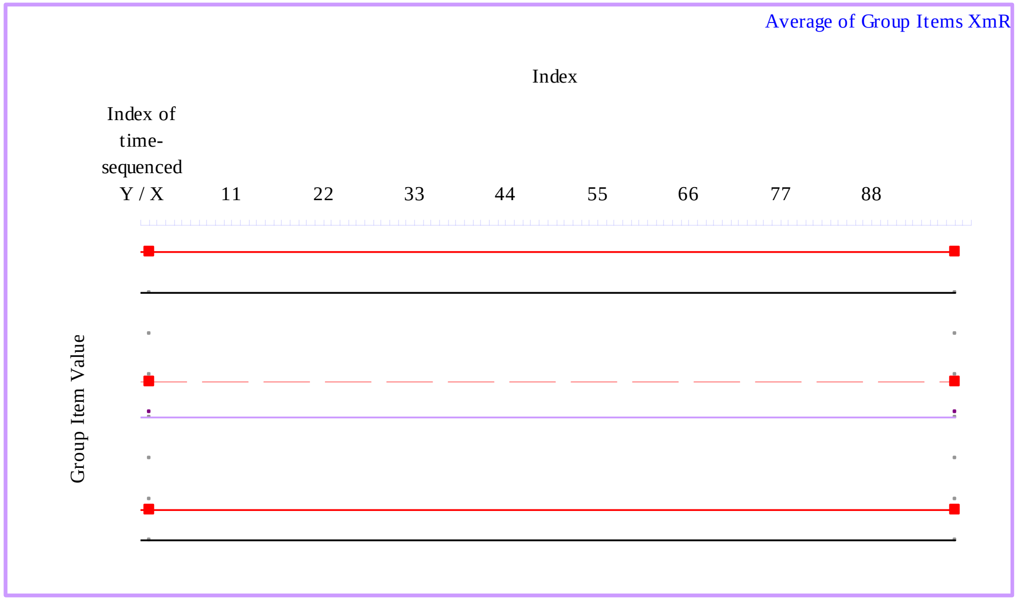
PPB' → PPO (before using PPM)

- Each iteration's PPB' is used as the interim PPO for the next iteration or similar project.
- PPB' as PPO must be derived and calculated from adjustments to historical data, not by guesswork, and is therefore a <u>realistic</u> <u>objective</u>.



PPB’ → PPO (before using PPM)

- Each subsequent iteration’s derived PPB and PPB’ gets better and better as improvements are continually and conscientiously applied by practitioners.
  - May not be for every iteration but for the overall project.

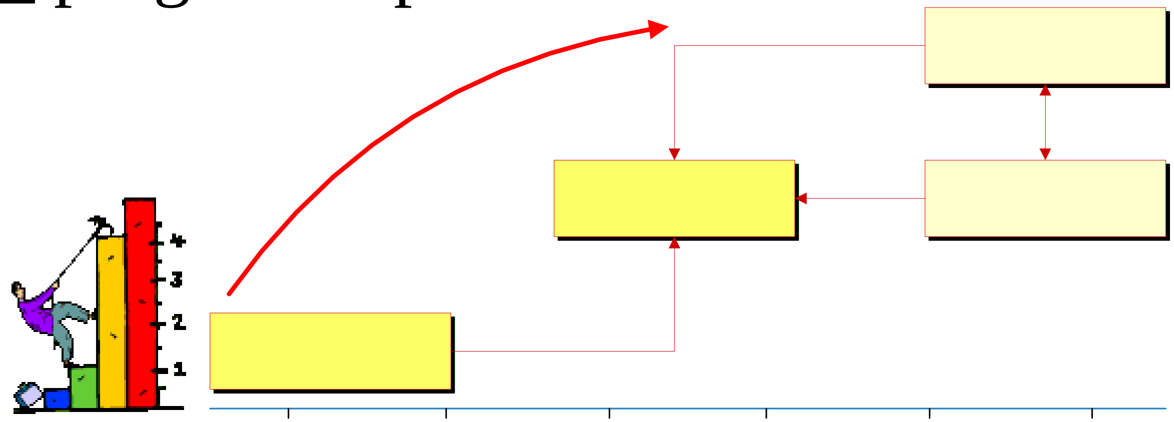


Data shown are not from the organization.

For illustration purpose only.

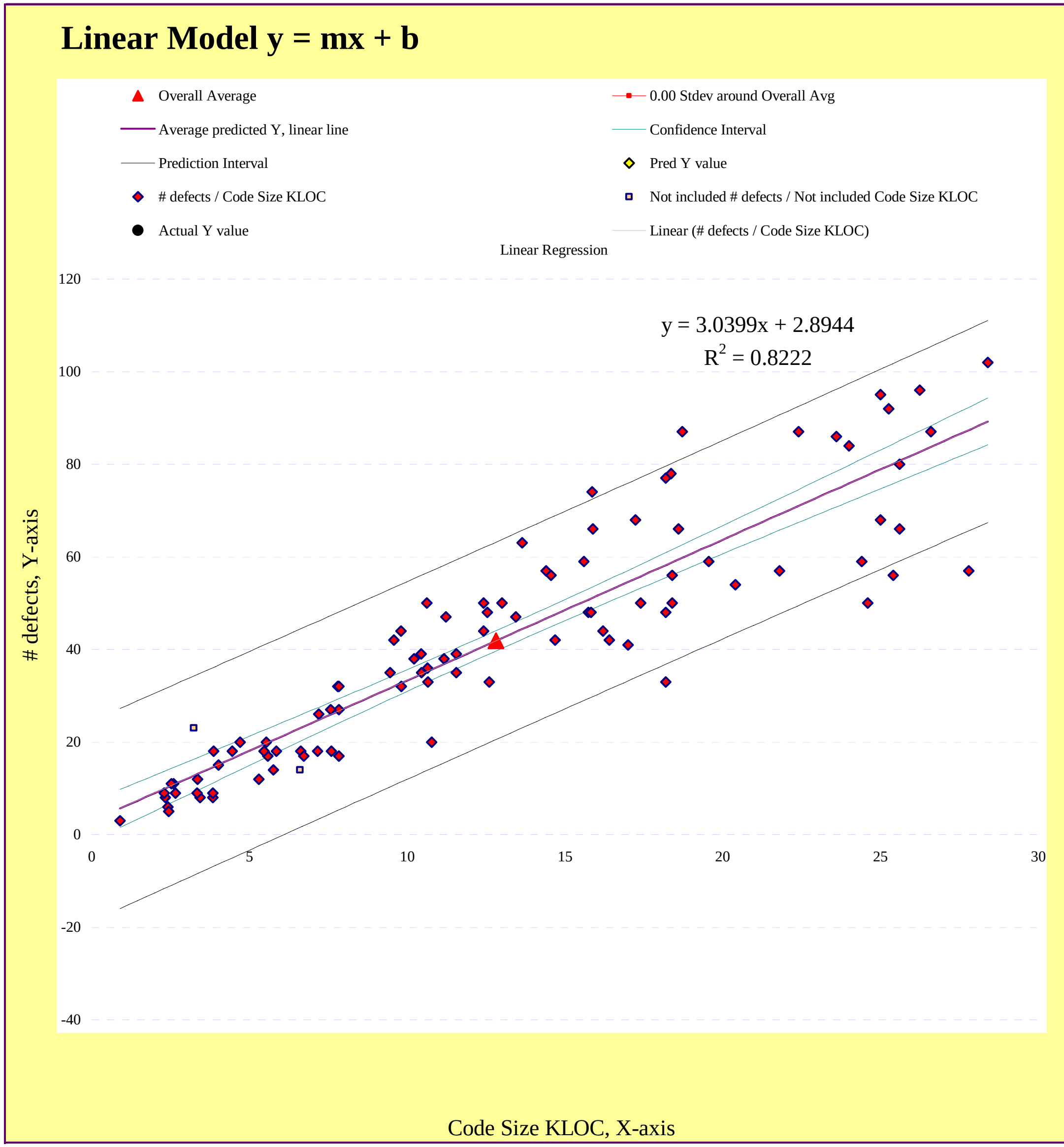
PPO (before using PPM)

- Each PPB' incrementally progresses towards the VOB and VOC as improvements are continuously applied.
- A process performance is therefore not immediately compared against its VOB or VOC.
- Incremental calculated progress is planned with realistic timelines.



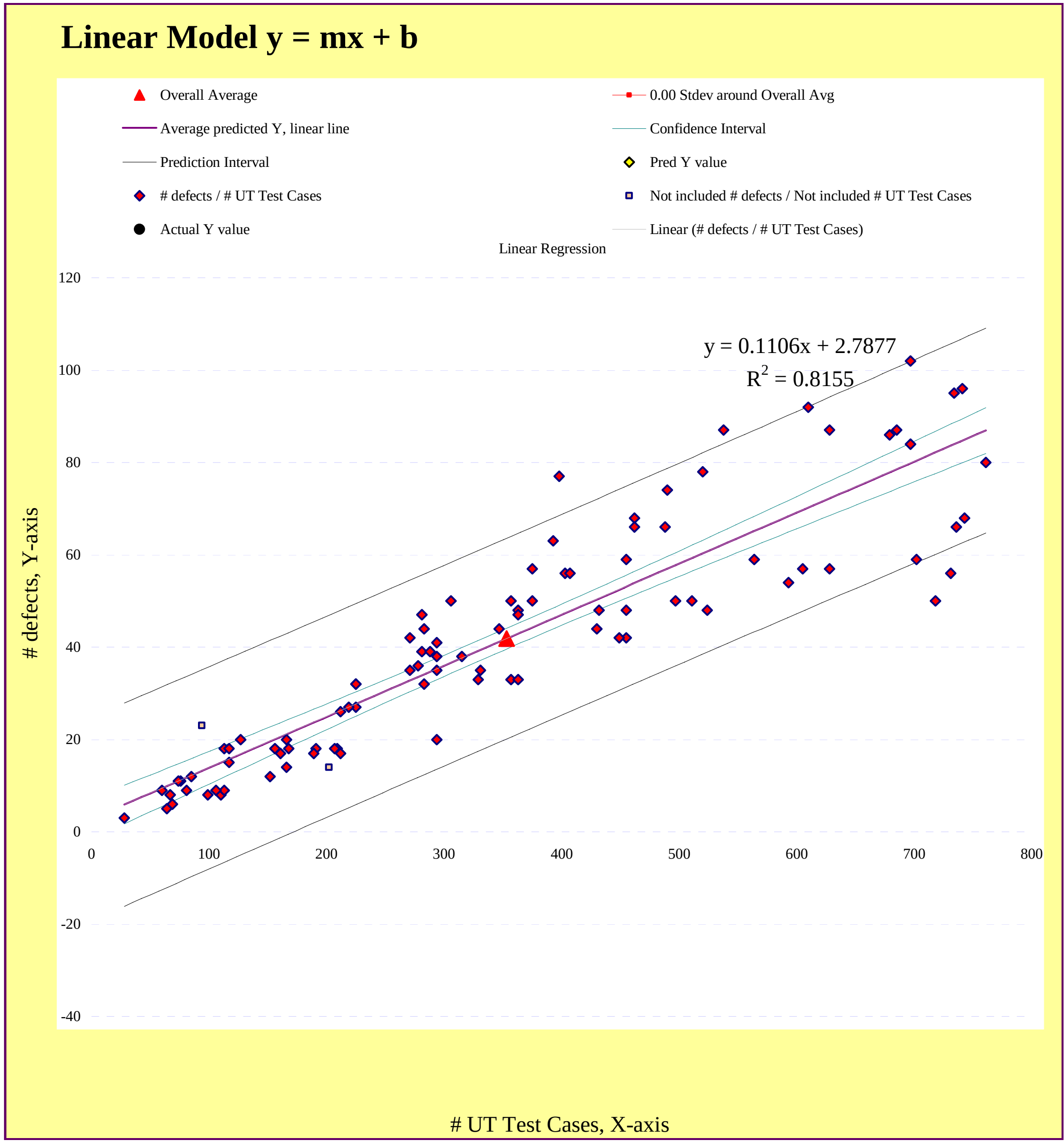
Correlation

- Use PPB' data to develop the correlations.
- Begin with a simple two variable regression that the practitioners can <u>see</u> and <u>feel</u>.
  - Output Y: #Defects found in a module during UT
  - Input X: Module Size KLoc
- Tool needs to be interactive.



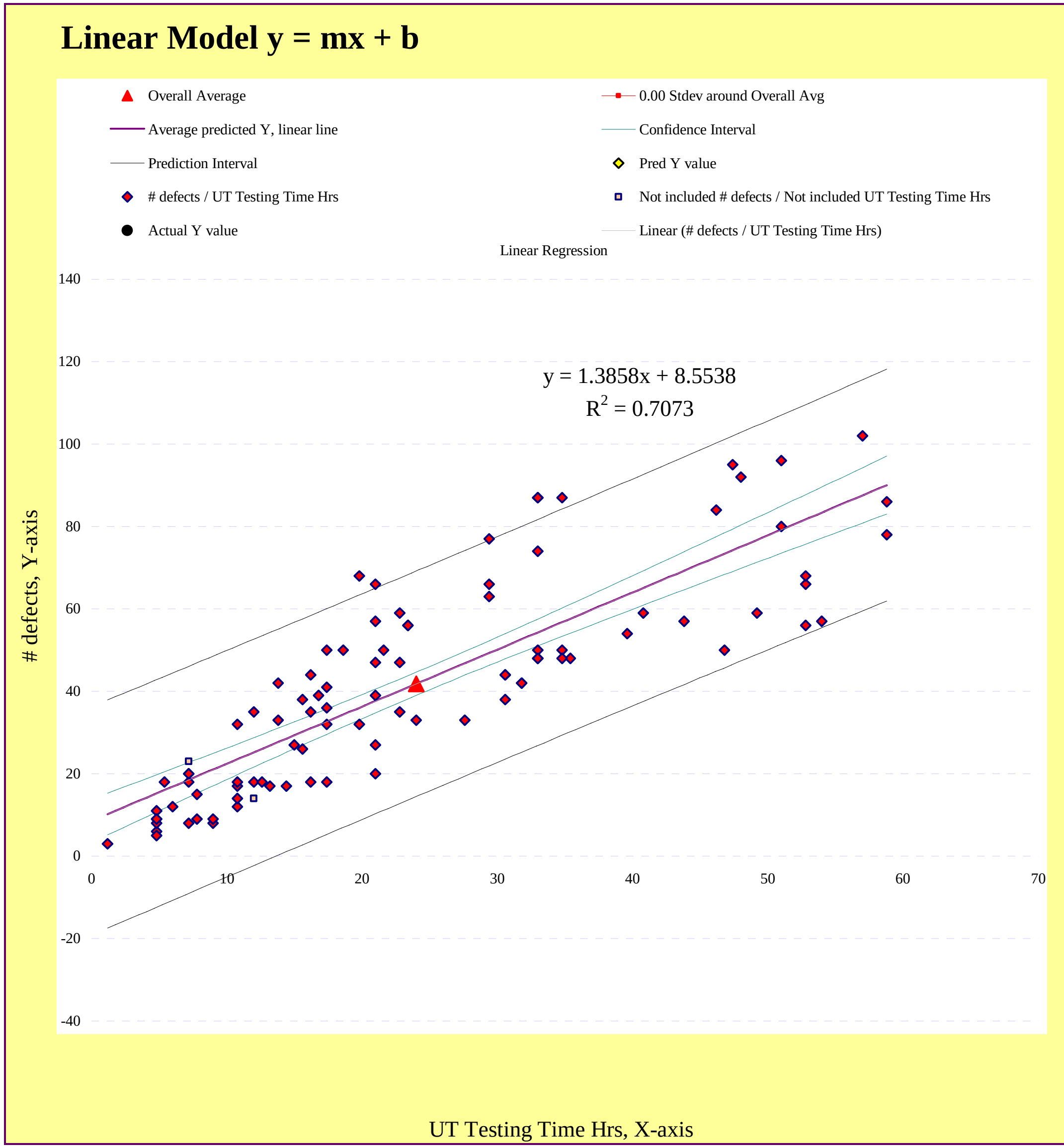
Correlation

- Develop other correlations in separate regressions so that the practitioners can see how other variables affect the output Y.
  - Output Y: #Defects found in a module during UT
  - Input X: #Test cases to test the module



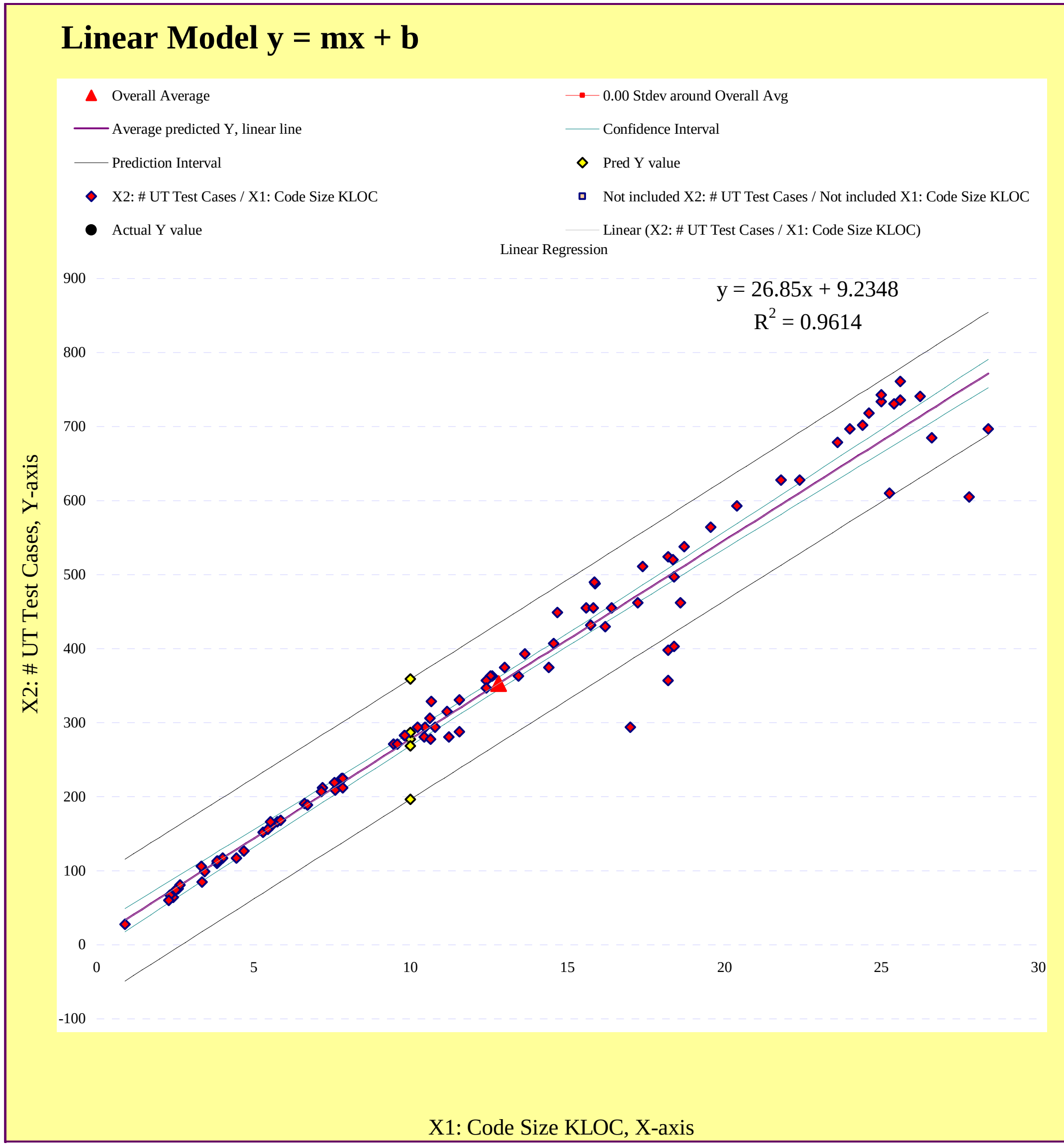
Correlation

- Exceptions or other data points that were removed would not be in the PPB' correlations
- Output Y: #Defects found in a module during UT
- Input X: Time spent to unit test the module



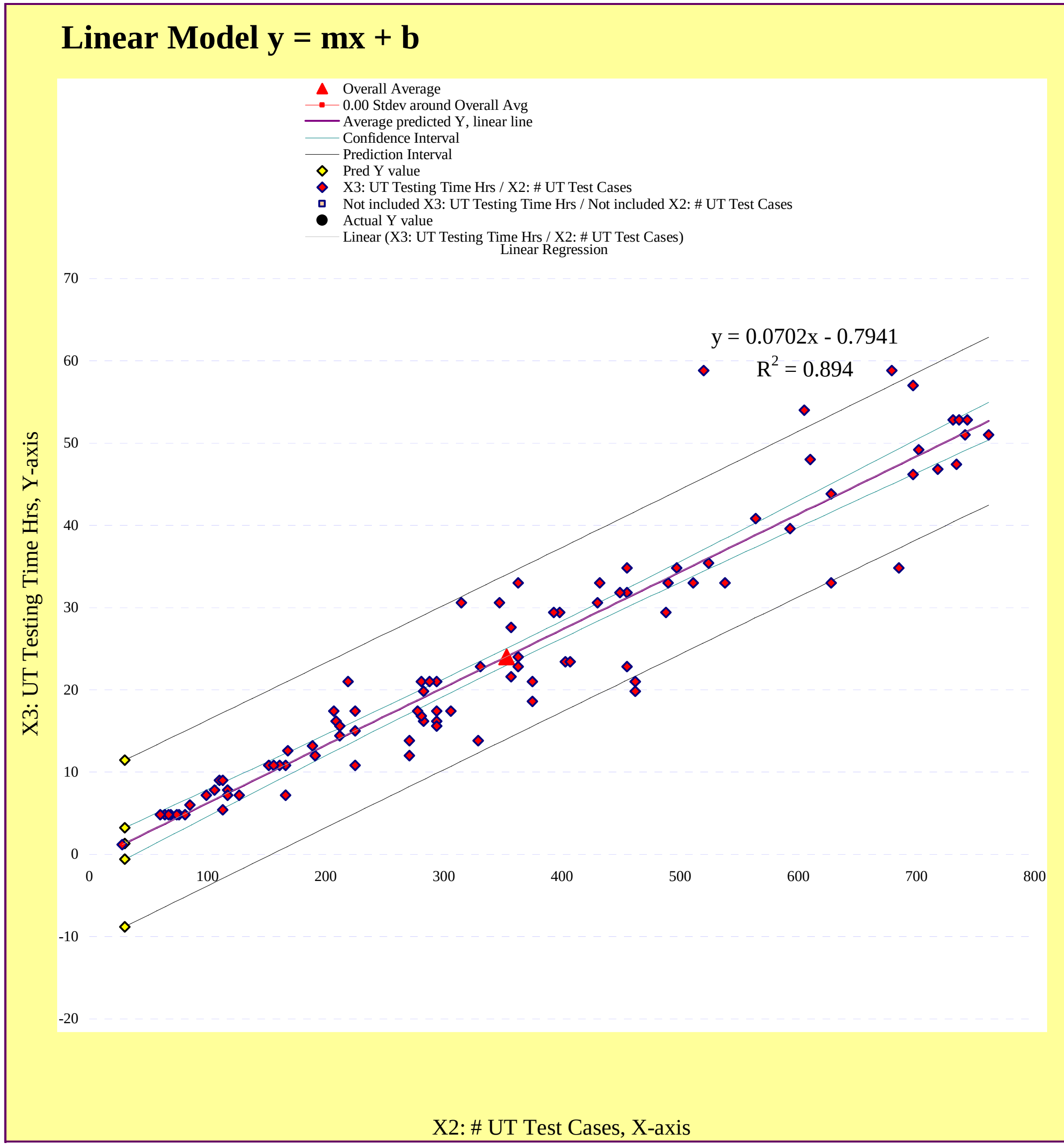
Correlation

- Include other correlations to see how variables affect each other.
  - Output: #Test cases to test the module
  - Input X: Module Size KLoc



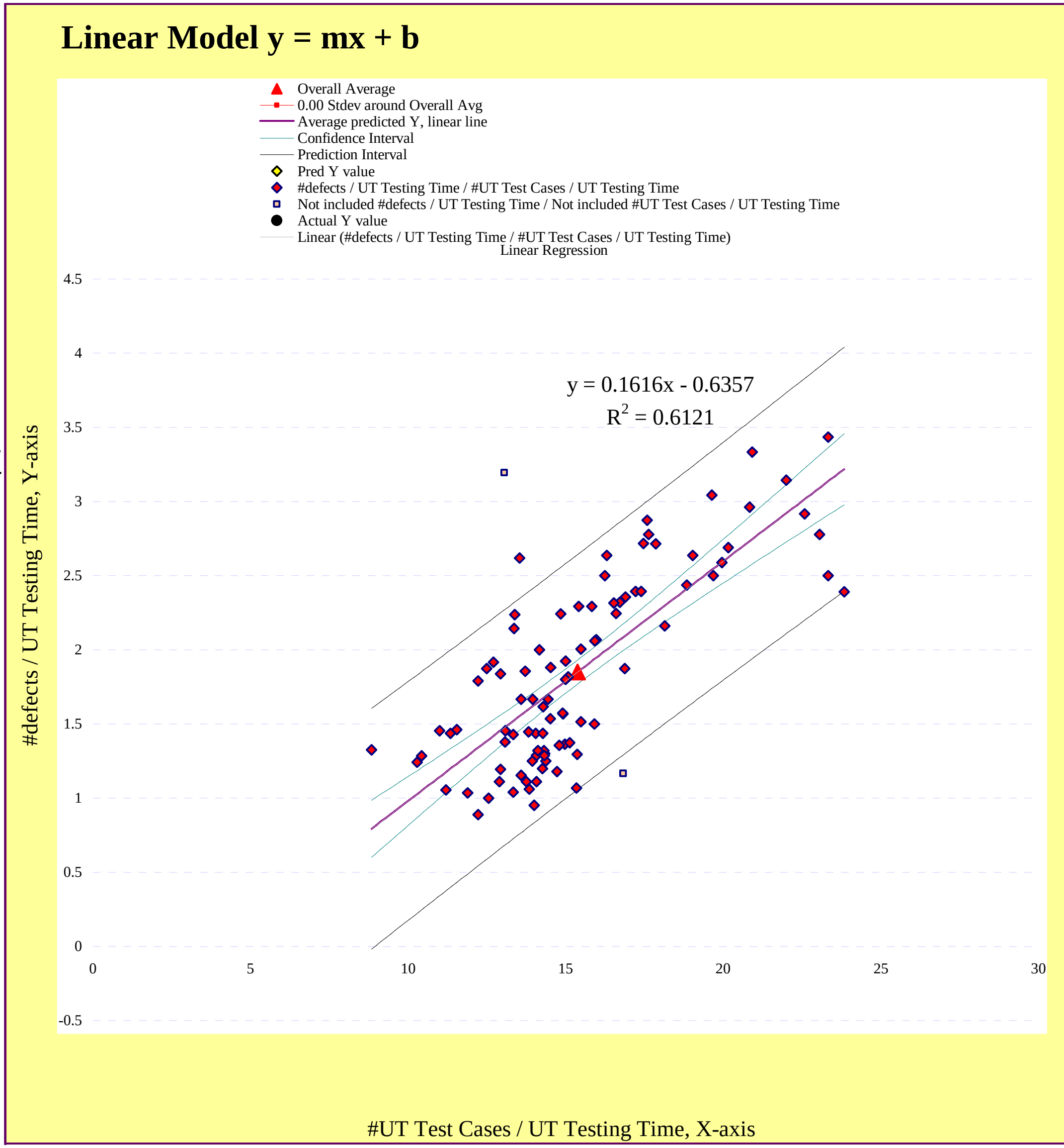
Correlation

- Include other correlations to see how variables affect each other.
  - Output: Time spent to unit test the module
  - Input X: #Test cases to test the module



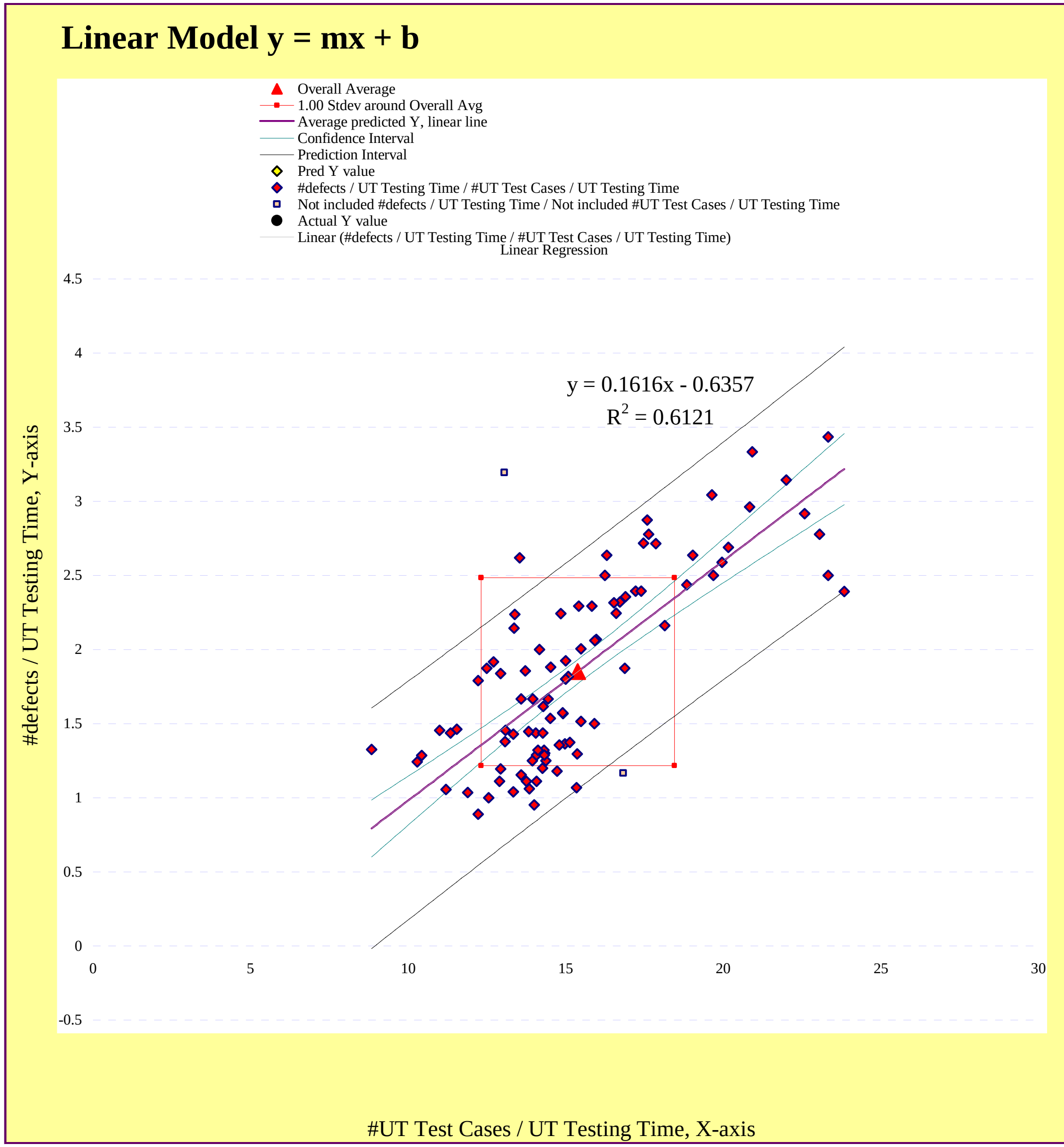
Modeling

- Later, include derived variables for modeling.
  - Output Y: #Defects found in a module during UT / Time Spent
  - Input X: #Test cases to test the module / Time Spent



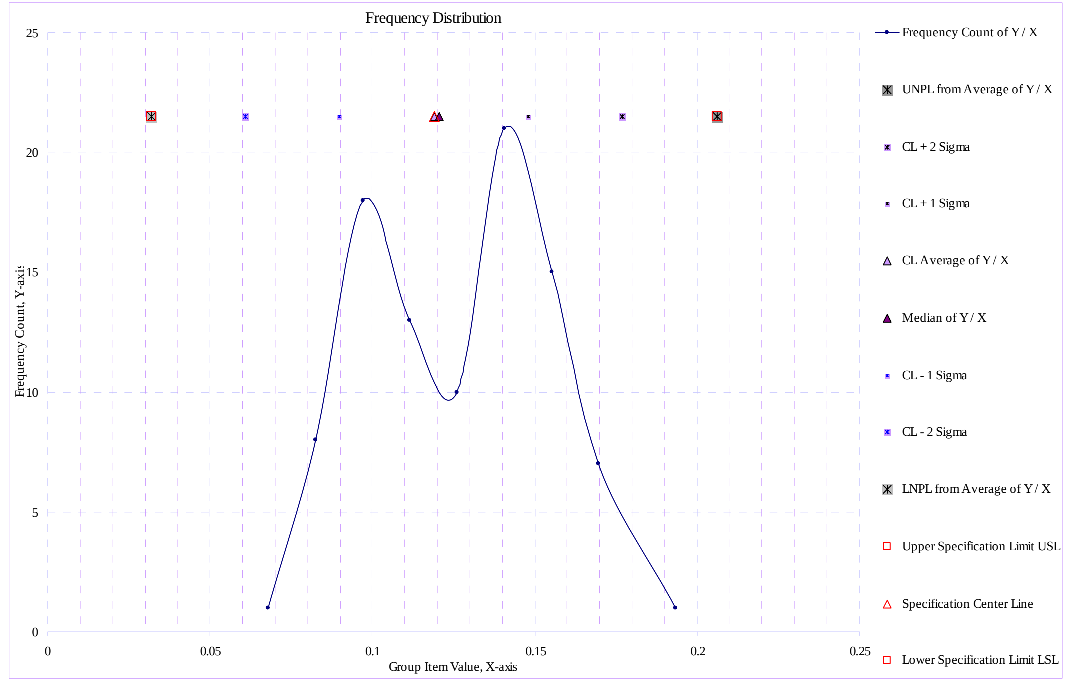
Modeling

- Include other analysis as required
  - One standard deviation around the average
  - Output Y: #Defects found in a module during UT / Time Spent
  - Input X: #Test cases to test the module / Time Spent



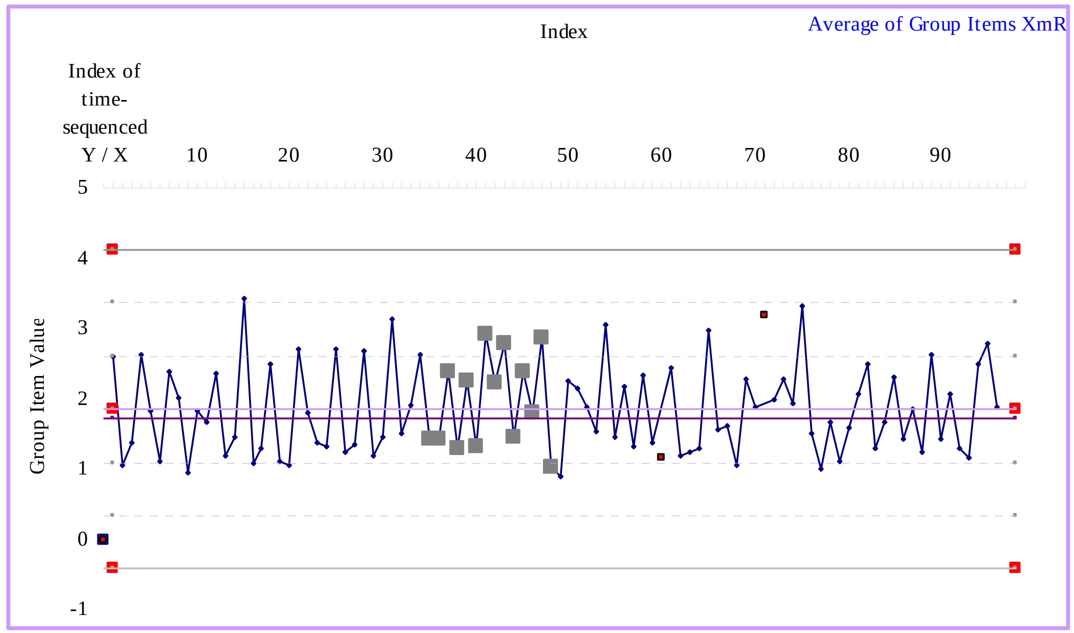
Frequency Distribution

- Frequency distribution
  - Y/X
  - Y: #Defects found in a module during UT / Time Spent
  - X: #Test cases to test the module / Time Spent
  - Senior developers
  - Junior developers
- Other tests of normality may be applied.



Exceptions

- There may be other exceptions to be improved.
  - Y/X
  - Y: #Defects found in a module during UT
  - X: Time spent to unit test the module



PPM

- When the practitioners are comfortable with the correlations, develop the multiple regression model using the X_n variables.

Y: # Defects	X_1: Code Size KLOC	X_2: # UT Test Cases	X_3: UT Testing Time Hrs
59	15.6	455	22.8
57	27.8	605	54
54	20.4	593	39.6
77	18.2	398	29.4
84	24	697	46.2
18	7.6	209	16.2
56	18.4	403	23.4
95	25	734	47.4
20	10.78	294	21
32	7.8	225	17.4

Data shown are just part of the complete set.

PPM

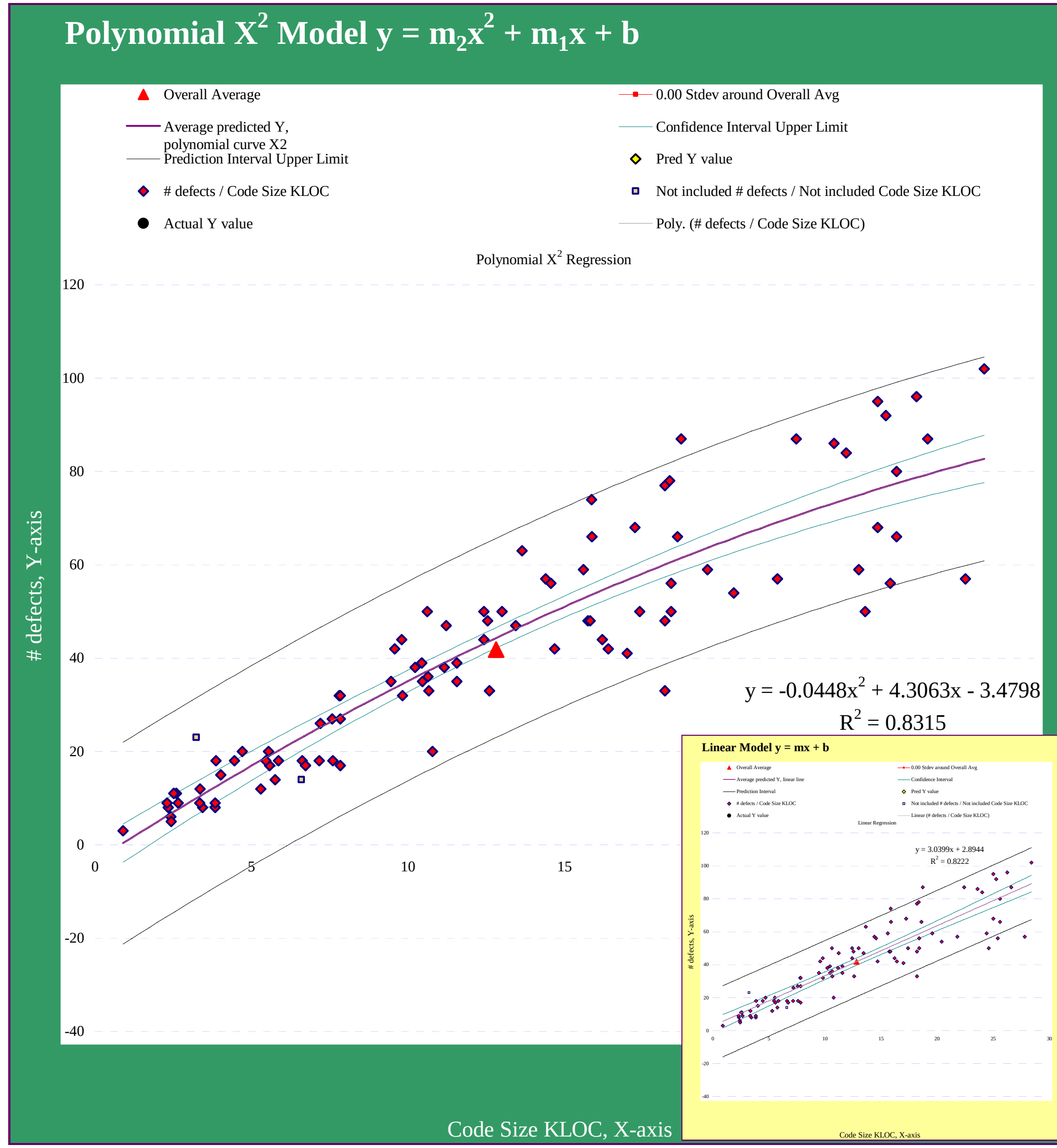
- $Y = 1.958602086*X_1 + 0.059436937*X_2 - 0.270573847*X_3 + 2.251835318$

Y: # Defects	X_1: Code Size KLOC	X_2: # UT Test Cases	X_3: UT Testing Time Hrs

Confidence Level	95.00%	0.05	Alpha				
Constant b set to zero?	Non Zero						
$y=m_1x_1+m_2x_2+m_3x_3+\ldots+b$		m_1	m_2	m3	b		
Coefficients		1.958602	0.059437	-0.27057	2.251835318	Constant b	
Standard Errors for m_n		0.74684	0.029757	0.221569	2.183538832	Standard error for b	
Upper 95.00%		3.44233	0.118555	0.169613	6.589816229		
Lower 95.00%		0.474874	0.000319	-0.71076	-2.086145592		
R^2	0.830087394	10.55925	Standard error for Y estimate				
F Statistics	146.5613558	90	df			1.5971E-34	F Distribution
ssreg	49023.8047	10034.8	ssresid				
t-observed values		2.62252	1.9974	1.221169	1.03127789	1.986674497	t-critical
P-values		0.01025	0.048802	0.225211	0.305173947		

Modeling improved

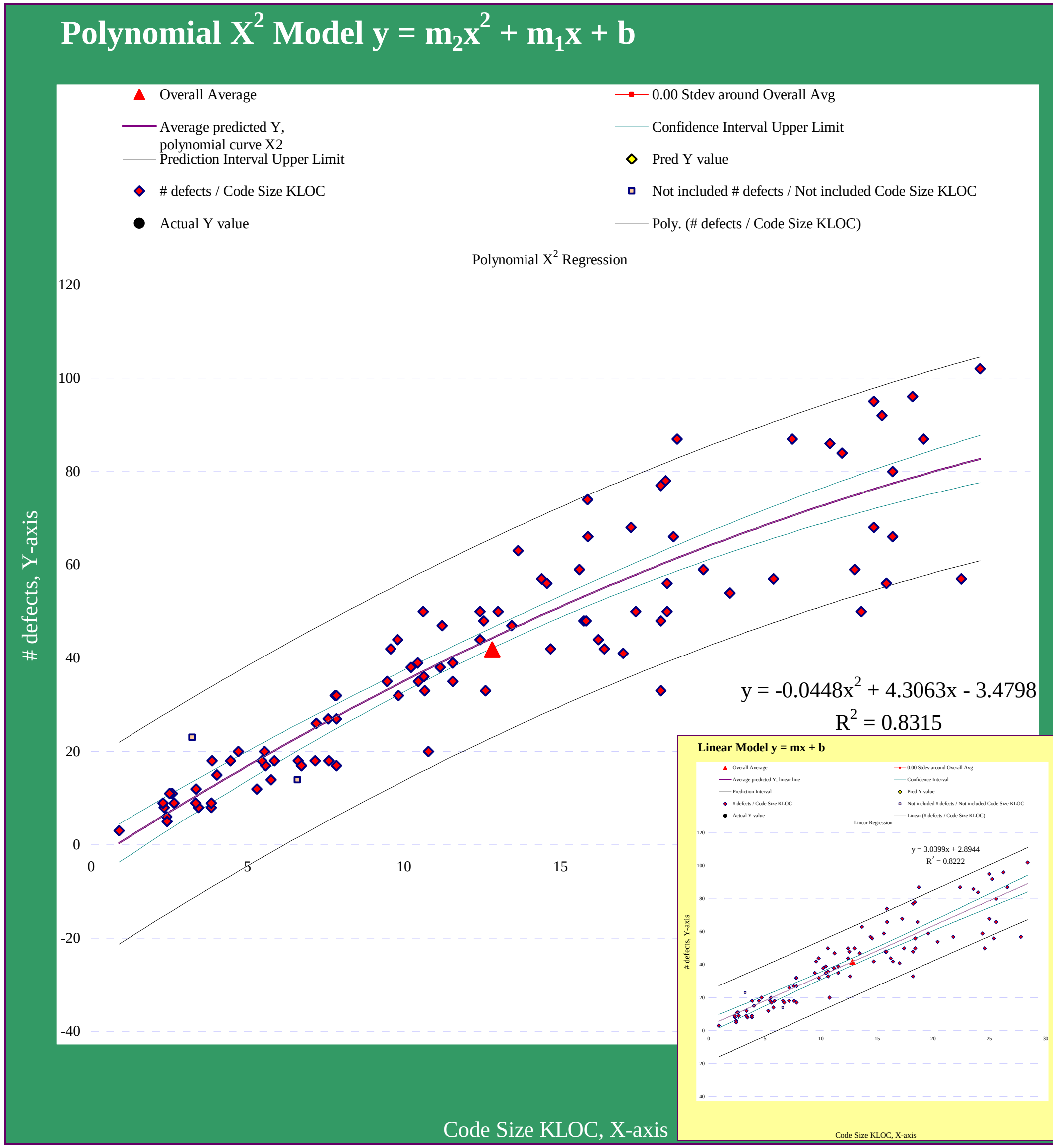
- As more analysis is performed, practitioners may realize that a linear regression may not be the case for some variables correlation.
  - Output Y: #Defects found in a module during UT
  - Input X: Module Size KLoc



Modeling improved

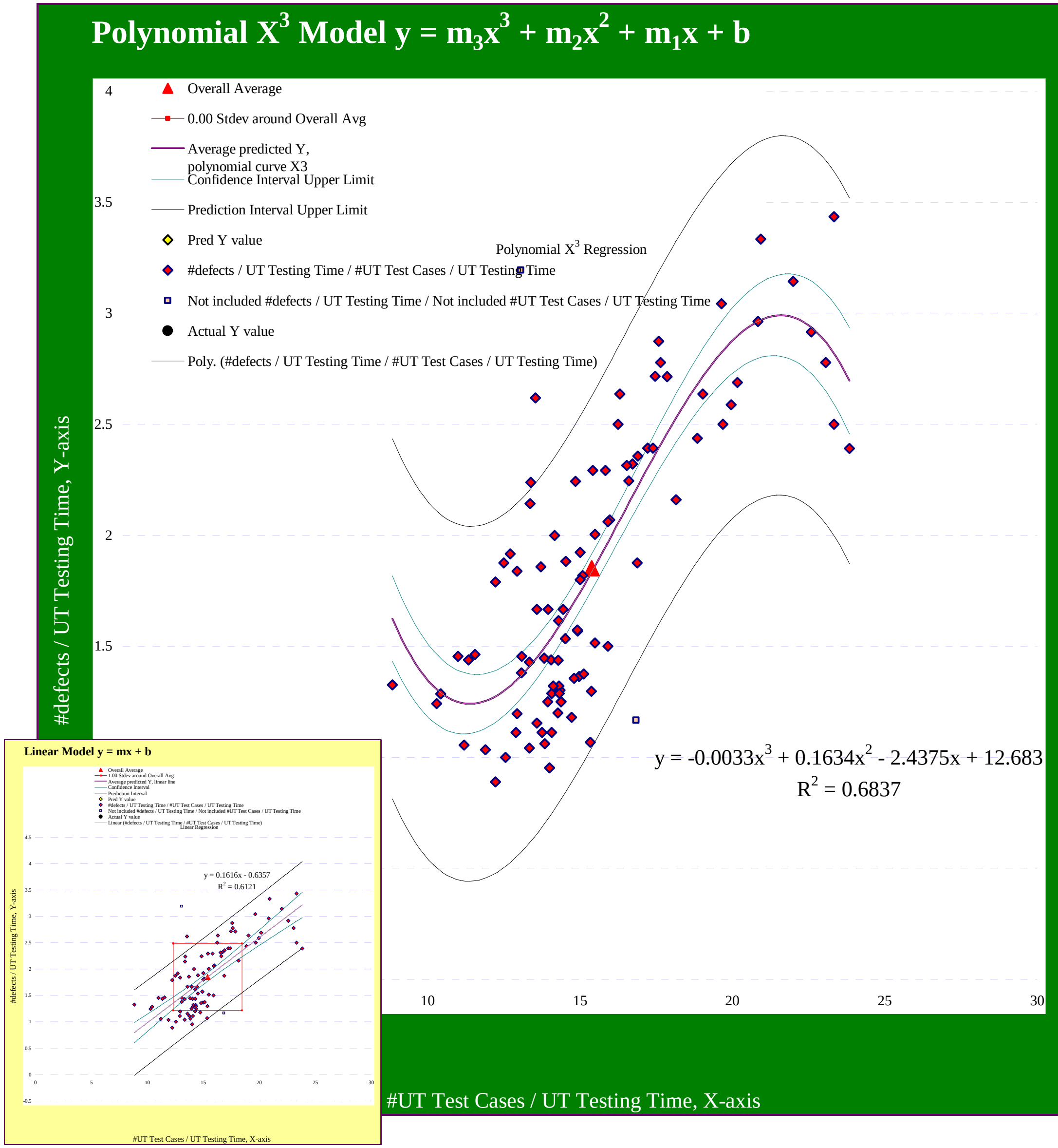
- Greatest gradient is at 9 KLoc

Defects	Code Size	Defect Density
0.781777055	1	0.781777055
4.953784418	2	2.476892209
9.036253663	3	3.012084554
13.02918479	4	3.257296197
16.93257779	5	3.386515559
20.74643268	6	3.45773878
24.47074945	7	3.495821349
28.10552809	8	3.513191012
31.65076862	9	3.516752069
35.10647103	10	3.510647103
38.47263532	11	3.497512302
41.74926149	12	3.479105124
44.93634954	13	3.456642273
48.03389947	14	3.43099282
51.04191129	15	3.402794086
53.96038498	16	3.372524061
56.78932055	17	3.340548268
59.52871801	18	3.307151
62.17857734	19	3.272556702
64.73889856	20	3.236944928
67.20968166	21	3.200461031
69.59092663	22	3.163223938
71.88263349	23	3.125331891
74.08480223	24	3.08686676
76.19743285	25	3.047897314
78.22052535	26	3.008481744



Modeling improved

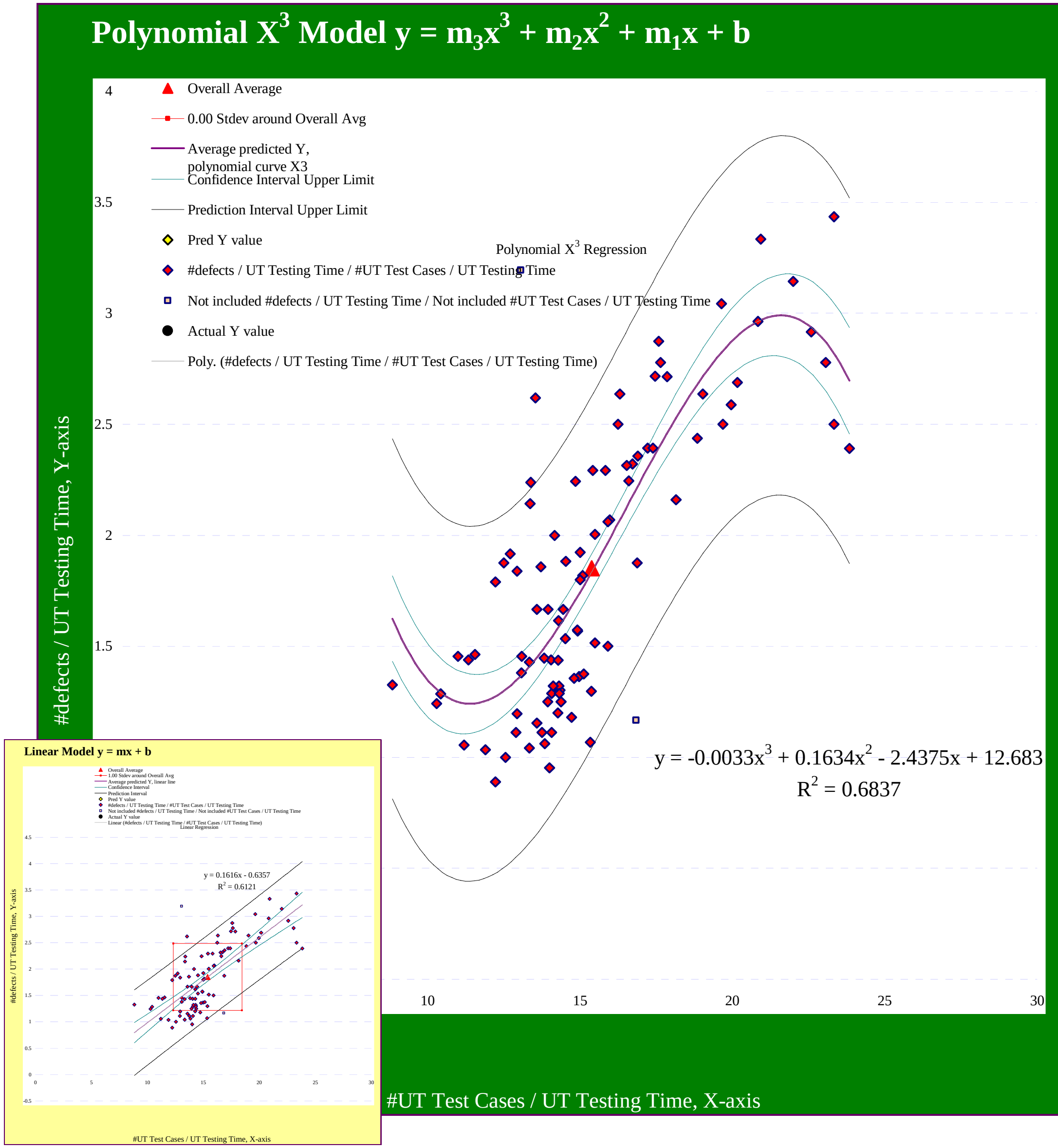
- Output Y: #Defects found in a module during UT / Time Spent
- Input X: #Test cases to test the module / Time Spent



Modeling improved

- Greatest gradient range:
  - 15 – 23 test cases per hour.

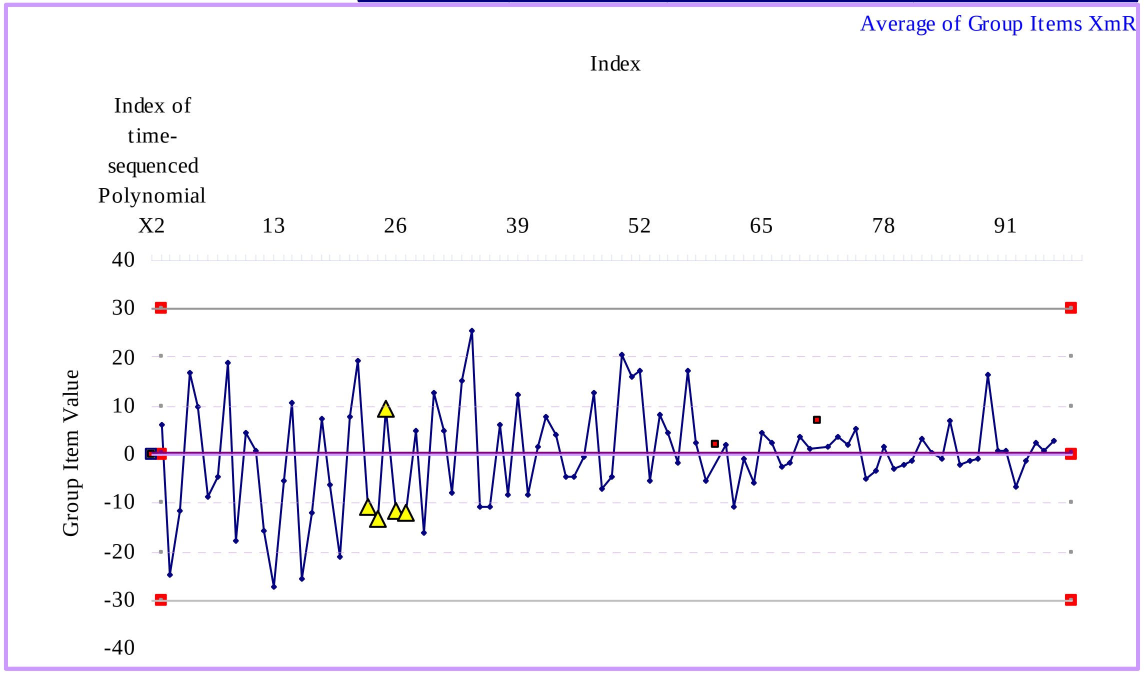
Defects / Testing Time	UT Test cases / Testing time	Defects / UT Test Cases
1.347993798	10	0.13479938
1.24901571	11	0.113546883
1.258854477	12	0.10490454
1.357688464	13	0.104437574
1.525696034	14	0.108978288
1.743055554	15	0.116203704
1.989945388	16	0.124371587
2.246543901	17	0.132149641
2.493029458	18	0.138501637
2.709580423	19	0.142609496
2.876375161	20	0.143818758
2.973592038	21	0.141599621
2.981409418	22	0.13551861
2.880005665	23	0.125217638
2.649559146	24	0.110398298
2.270248224	25	0.090809929
1.722251265	26	0.066240433
0.985746634	27	0.036509135



Modeling Improved

- The residual of the polynomial X^2 model should then be used in the XmR control chart to detect exceptions instead of Y/X.

# Defects	Code Size KLOC	# Defects / Code Size KLOC	Polynomial X^2 Residual
59	15.6	3.782051282	6.196259923
57	27.8	2.050359712	-24.63645579
54	20.4	2.647058824	-11.73795637
77	18.2	4.230769231	16.93414707
84	24	3.5	9.91519777
18	7.6	2.368421053	-8.662361209
56	18.4	3.043478261	-4.599406317
95	25	3.8	18.80256715
20	10.78	1.85528757	-17.73976155
32	7.8	4.102564103	4.614264586



PPM improved

- The preferred regression formula is used in the multiple regression:

Y: # Defects	Code Size KLOC ➔ $-0.0448X_1\hat{}2 + 4.3063X_1 - 3.4798$	X_2: # UT Test Cases	X_3: UT Testing Time Hrs
59	52.80374	455	22.8
57	81.63646	605	54
54	65.73796	593	39.6
77	60.06585	398	29.4
84	74.0848	697	46.2
18	26.66236	209	16.2
56	60.59941	403	23.4
95	76.19743	734	47.4
20	37.73976	294	21
32	27.38574	225	17.4

PPM

Y: # Defects	Code Size KLOC → $-0.0448X_1$^2 + $4.3063X_1$ - 3.4798	X_2: # UT Test Cases	X_3: UT Testing Time Hrs

Confidence Level	95.00%	0.05	Alpha			
Constant b set to zero?	Non Zero					
$y=m_1x_1+m_2x_2+m_3x_3+\ldots+b$		m_1	m_2	m3	b	
Coefficients		0.065908	0.119684	-0.21865	2.054101583	Constant b
Standard Errors for m_n		0.095014	0.018365	0.229078	2.40169675	Standard error for b
Upper 95.00%		0.254669	0.15617	0.236454	6.825491266	
Lower 95.00%		-0.12285	0.083197	-0.67375	-2.717288101	
R^2	0.818075672	10.92611	Standard error for Y estimate			
F Statistics	134.9037284	90	df			3.42911E-33 F Distribution
ssreg	48314.40909	10744.2	ssresid			
t-observed values		0.693667	6.516767	0.954476	0.855271001	1.986674497 t-critical
P-values		0.489677	4.05E-09	0.342399	0.394672276	

P-values did not improve so do not use the earlier regression formula for X_1.

PPM

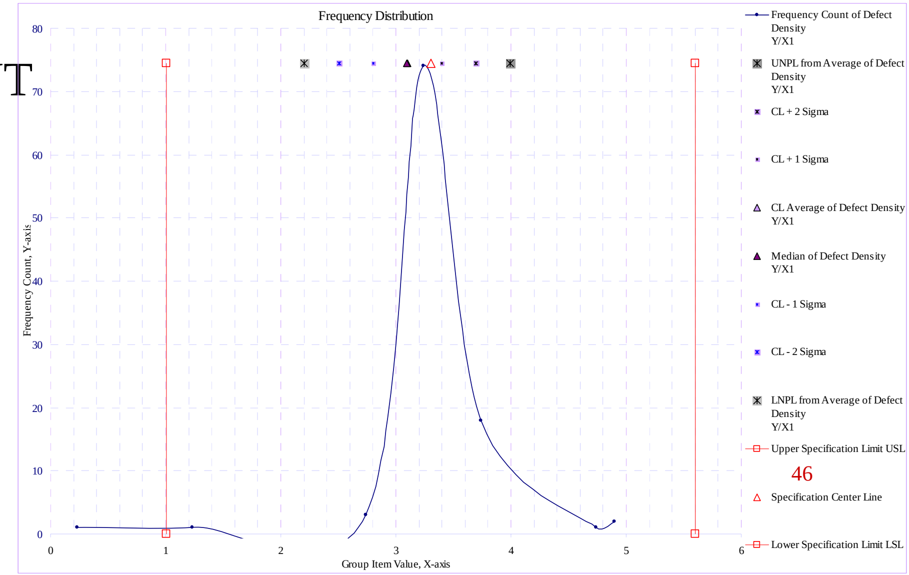
- $Y = 1.912166199*X_1 + 0.057942217*X_2 -0.003927848*(X_3)^2 + 0$
- Constant b (intercept) set to zero
- 90% confidence level. P-values have improved by using (X_3)^2.

Y: # Defects	X_1: Code Size KLOC	X_2: # UT Test Cases	X_3: (UT Testing Time Hrs)^2

Confidence Level	90.00%	0.1	Alpha				
Constant b set to zero?	Zero						
$y=m_1x_1+m_2x_2+m_3x_3+\ldots+b$		m_1	m_2	m3	b		
Coefficients		1.912166	0.057942	-0.00393	0	Constant b	
Standard Errors for m_n		0.733273	0.027162	0.002075	#N/A	Standard error for b	
Upper 90.00%		3.130698	0.103079	-0.00048	#N/A		
Lower 90.00%		0.693634	0.012805	-0.00738	#N/A		
R^2	0.955483871	10.45901	Standard error for Y estimate				
F Statistics	651.0676344	91	df			1.21531E-54	F Distribution
ssreg	213662.4368	9954.563	ssresid				
t-observed values		2.607713	2.133218	1.892992	#N/A	1.661771156	t-critical
P-values		0.010653	0.035598	0.061537	#N/A		

Monte Carlo with X_3 as (X_3)^2

- Simulation of the following:
  - X_1 ranges from 1 to 50 KLOC of Module Size
  - X_2 ranges from >= 1 Test Cases
    - (Max test cases simulated was up to 1448, correlated with file size)
  - (X_3)^2 ranges from >=1 Testing Time
    - (Max testing time simulated was up to 12624 hrs^2, correlated with # test cases)
    - 12624 hrs^2 = (112.35 hrs)^2
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 97.4% >=LSL , 98.85% <= USL
    - 96.25% within LSL and USL



Data shown are of one instance of the simulation.

Optimum range of X_1: Code Size

- To ensure PPO can be achieved or *exceeded*
- Arrange the input variables in the possible permutations (2^n) of their *reasonable* minimum and maximum values

$$Y = 1.912166199*X_1 + 0.057942217*X_2 - 0.003927848*(X_3)^2$$

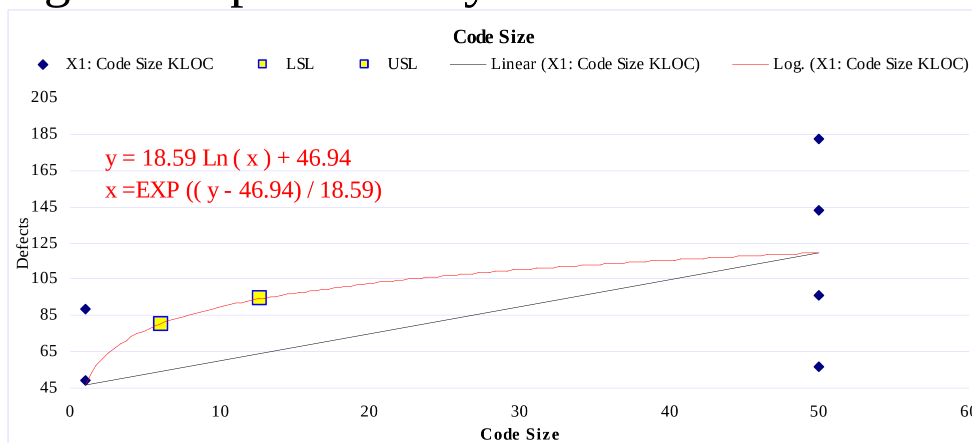
X1	X2	X3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

	Y: # Defects	X_1: Code Size KLOC	X_2: # UT Test Cases	X_3: UT Testing Time Hrs^2
	2.47587698	1	10	4
Remove -ve Y		1	10	10000
	88.8097803	1	1500	4
	49.5470133	1	1500	10000
	96.1720207	50	10	4
	56.9092537	50	10	10000
	182.505924	50	1500	4
	143.243157	50	1500	10000

Copyright Rayney Wong

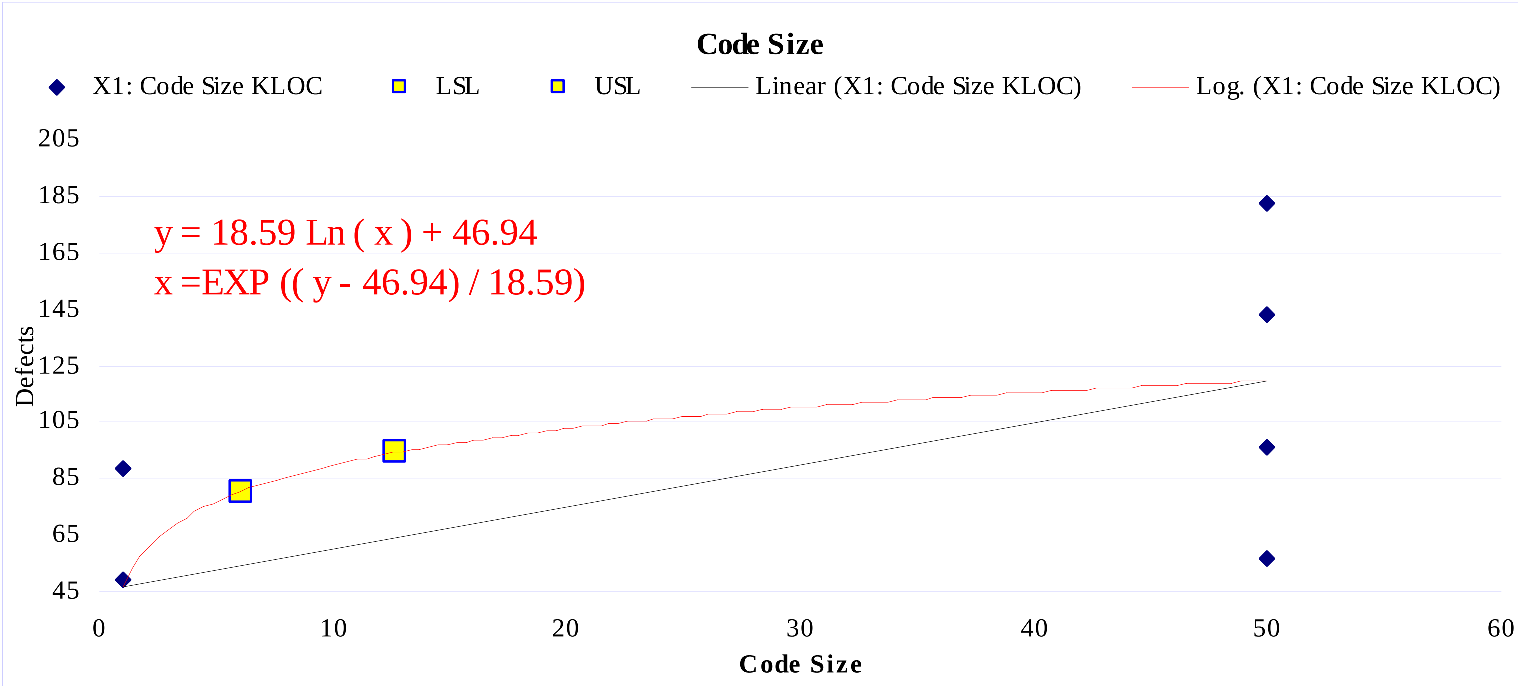
Optimum range of X_1: Code Size

- Plot Y against X_1: Code Size
- Code Size is the most important controllable factor
- Keep all file sizes <= 12 KLoc during planning of the modules' WBS (work breakdown structure)
  - The higher the gradient, usually the higher the productivity



Optimum range of X_1: Code Size

Defects	Code Size
95.83397868	13.8528494
95.02715121	13.26478748
94.22032373 →	12.70168915
93.50146522	12.22015383
93.48747921	12.21096848
91.55774891	11.00759279
90.42074775	10.35485998
89.61403259	9.915349726
88.9882957	9.58732721
88.66999985	9.424657582
88.47703144	9.327385204
86.95887644	8.596311257
86.15204896	8.231392594
85.34522148	7.88196495
84.53839401	7.547370724
83.73156653	7.226980226
82.92473905	6.920190501
82.11791157	6.626424188
81.3110841	6.345128433
80.50425662 →	6.075773856
79.69742914	5.817853544
78.89060167	5.570882108

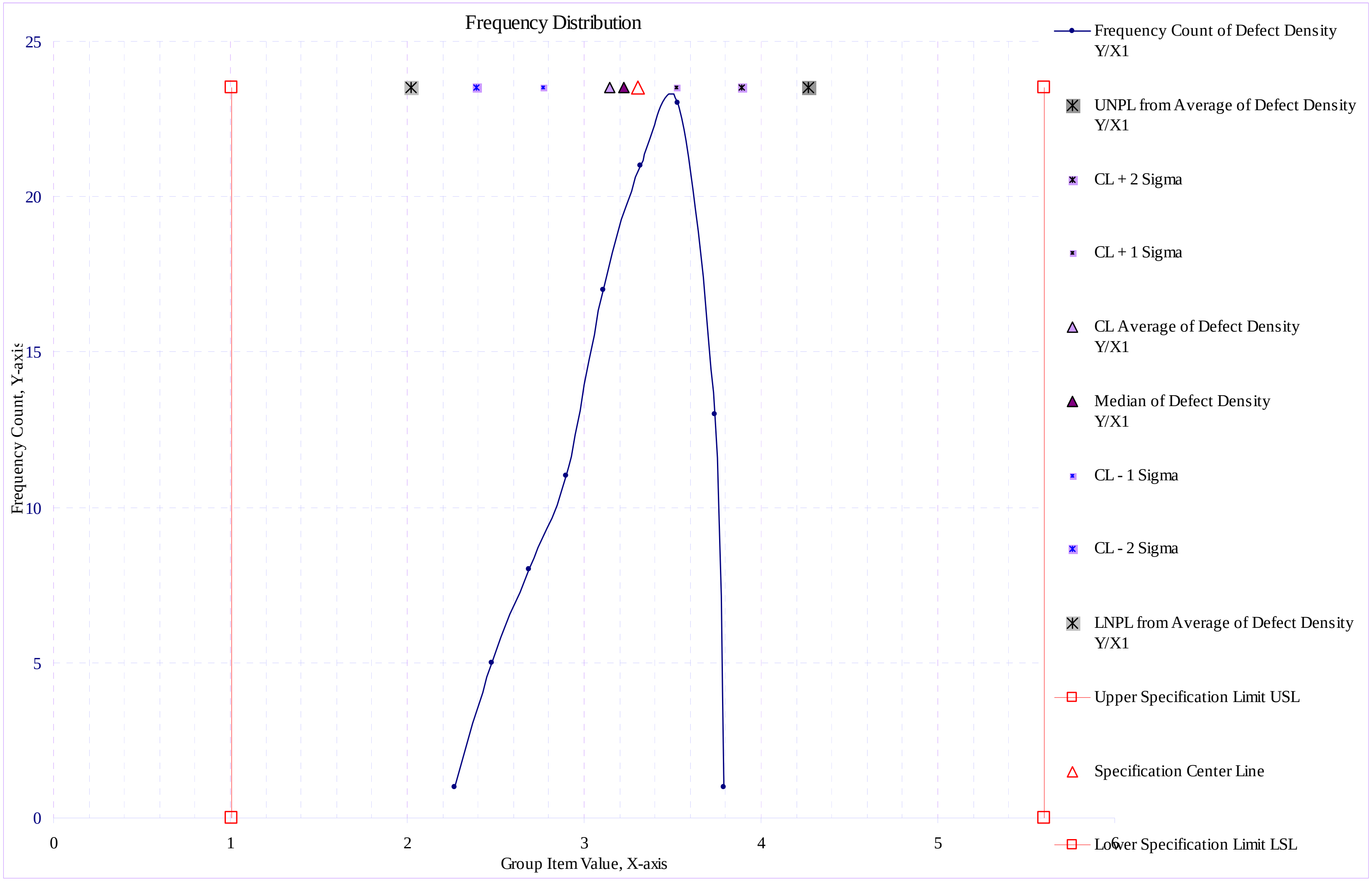


Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

- Simulation of the following:
  - X_1 ranges from 6 to 12 KLOC of Module Size
  - X_2 ranges from >= 1 Test Cases
    - (Max test cases simulated was up to 428, correlated with file size)
  - (X_3)^2 ranges from >=1 Testing Time
    - (Max testing time simulated was up to 3245 hrs^2, correlated with # test cases)
    - 3245 hrs^2 = (57 hrs)^2
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 99.95% >=LSL , 100% <= USL
    - 99.95% within LSL and USL

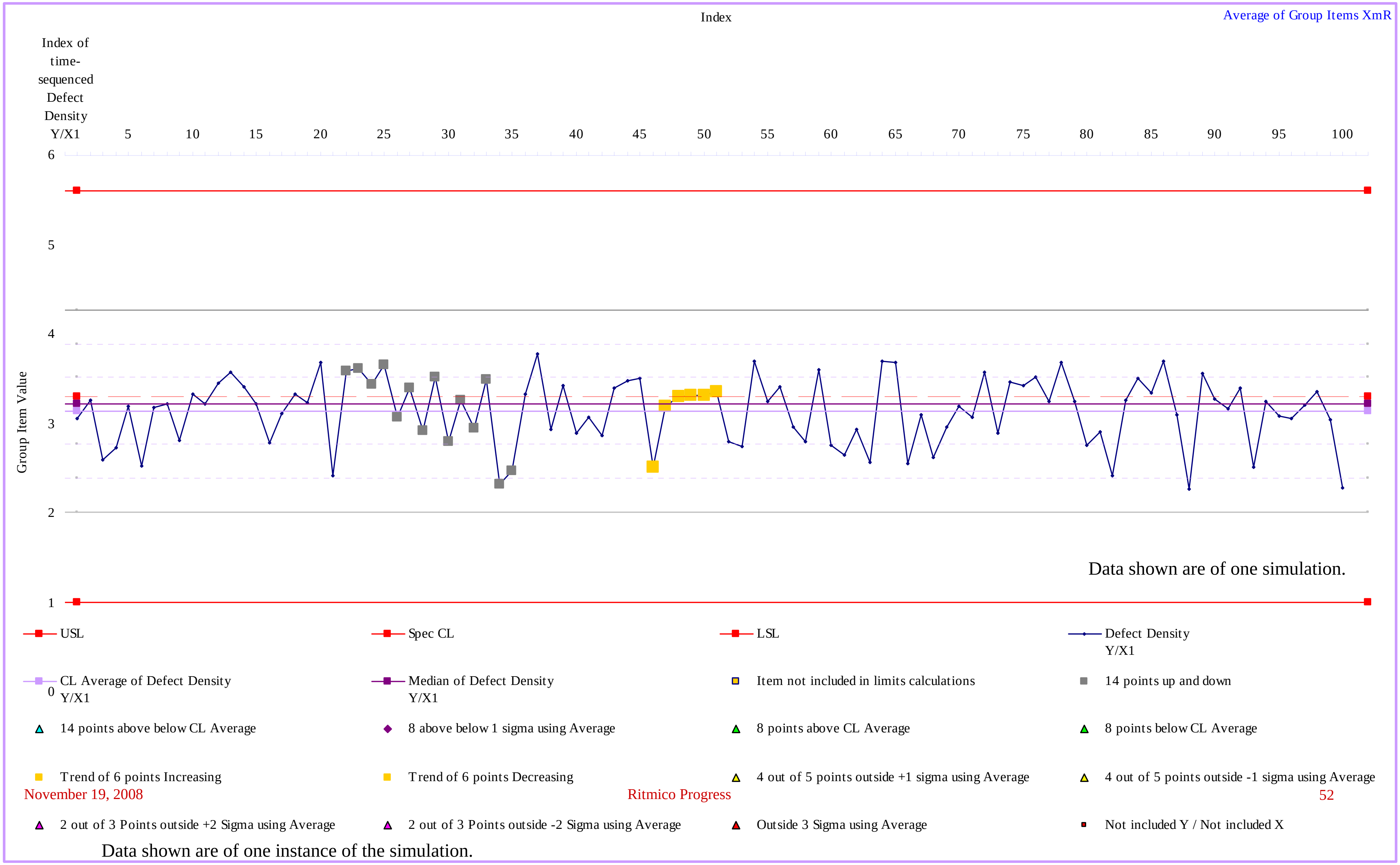
Data shown are of one simulation.

Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges



Data shown are of one instance of the simulation.

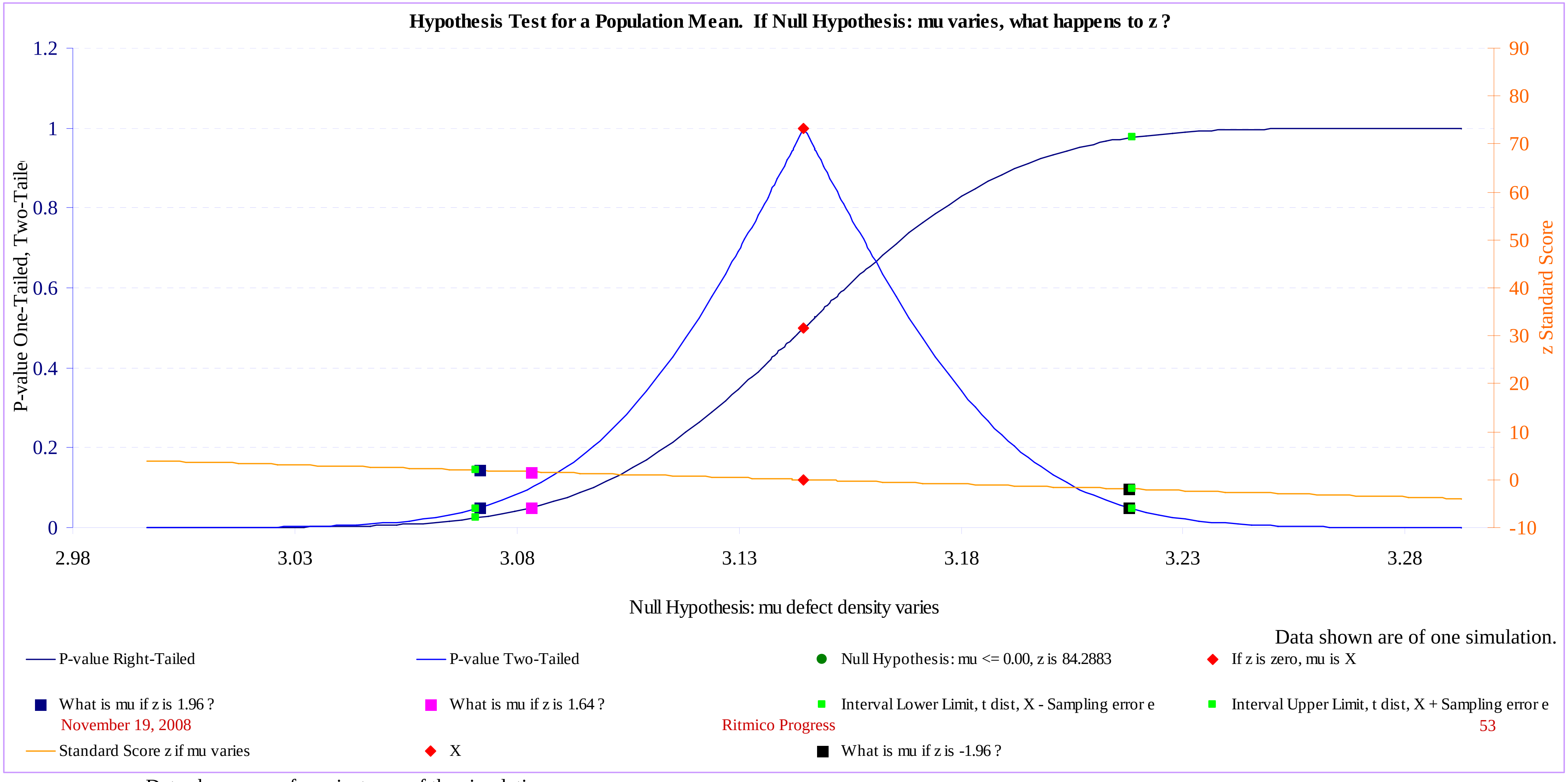
Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges



Data shown are of one instance of the simulation.

Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

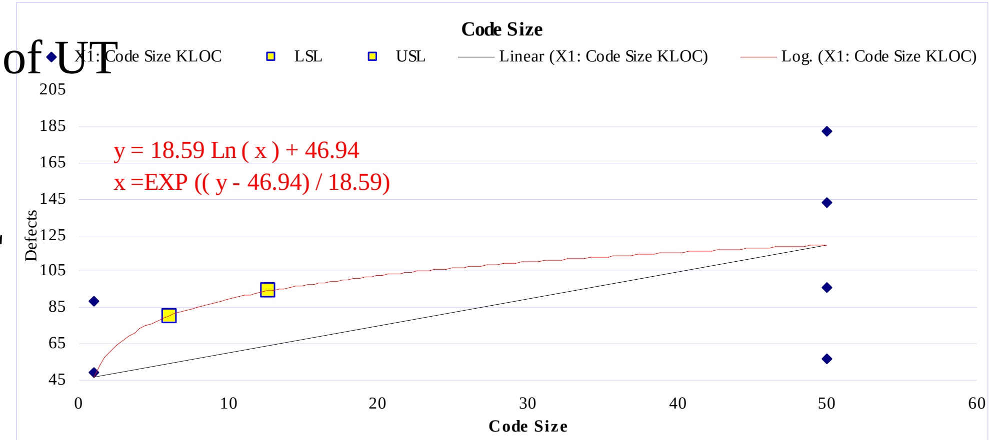
- 95% confidence level of defect density: 3.07 – 3.22



Data shown are of one instance of the simulation.

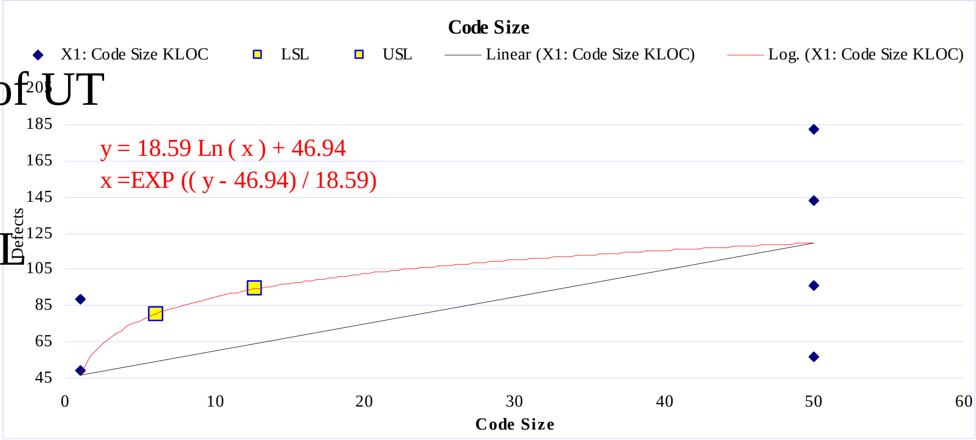
Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

- Simulation of the following:
  - X_1 ranges from 6 to 50 KLOC of Module Size
  - X_2 ranges from >= 1 Test Cases
    - (Max test cases simulated was up to 1444, correlated with file size)
  - (X_3)^2 ranges from >=1 Testing Time
    - (Max testing time simulated was up to 11418 hrs^2, correlated with # test cases)
    - 11418 hrs^2 = (106 hrs)^2
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 99.95% >=LSL , 100% <= USL
    - 99.95% within LSL and USL



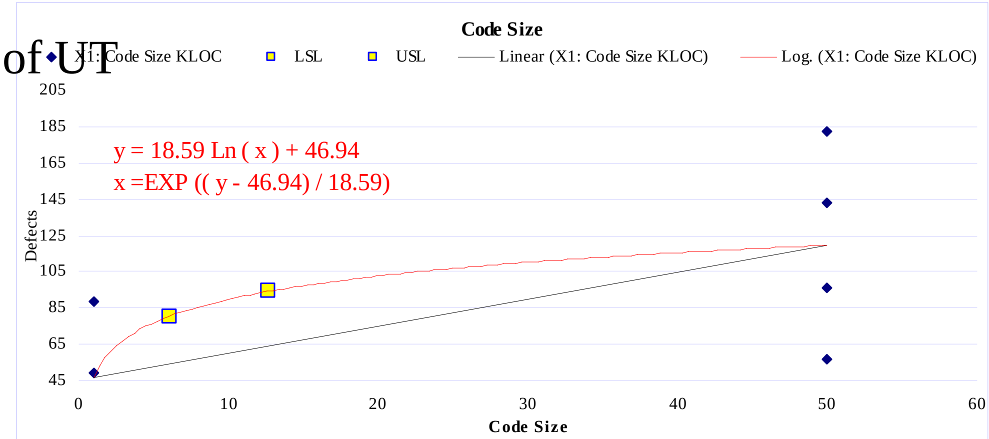
Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

- Simulation of the following:
  - In reality, there will be module Module Size of < 6
  - X_1 ranges from 1 to 12 KLOC of Module Size
  - X_2 ranges from >= 1 Test Cases
    - (Max test cases simulated was up to 428, correlated with file size)
  - (X_3)^2 ranges from >=1 Testing Time
    - (Max testing time simulated was up to 3273 hrs^2, correlated with # test cases)
    - 3273 hrs^2 = (57.2 hrs)^2
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 92.55% >=LSL , 96.85% <= USL
    - 89.40% within LSL and USL



Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

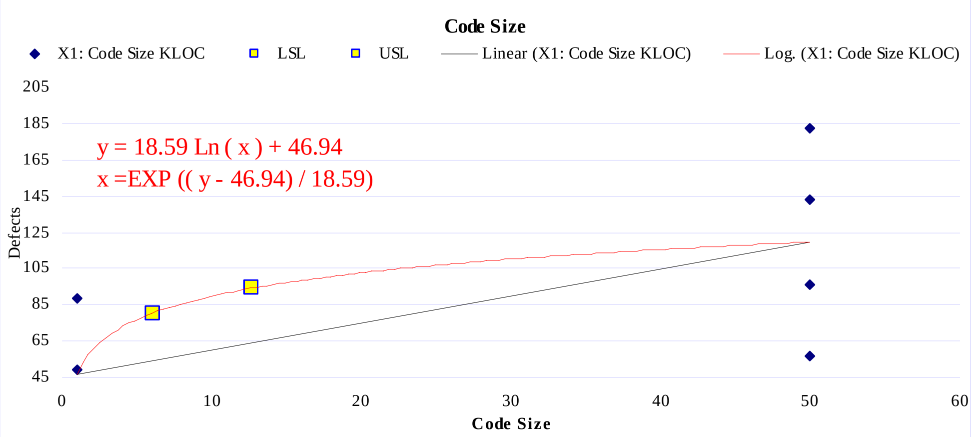
- Simulation of the following:
  - X_1 ranges from 1 to 6 KLOC of Module Size
  - X_2 ranges from >= 1 Test Cases
    - (Max test cases simulated was up to 264, correlated with file size)
  - (X_3)^2 ranges from >=1 Testing Time
    - (Max testing time simulated was up to 2725 hrs^2, correlated with # test cases)
    - 2725 hrs^2 = (52.2 hrs)^2
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 85.2% >=LSL , 93.8% <= USL
    - 79% within LSL and USL

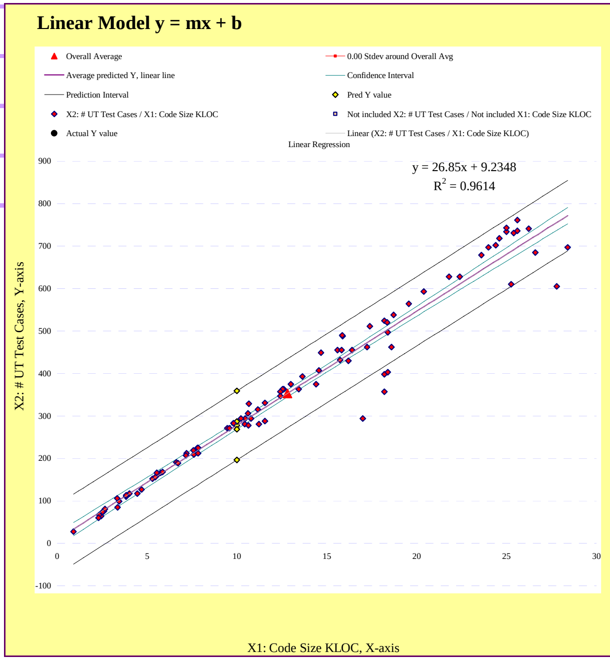


Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

- In the simulation of module size between 1 to 6, reasons for having many instances below LSL:
  - # of test cases was not enough or there were zero defects simulated.

Module Size Range KLOC	1 to 6	1 to 12	1 to 50	6 to 12	6 to 50
LSL >=	85.20%	92.55%	97.40%	99.95%	99.95%
<= USL	93.80%	96.85%	98.85%	100.00%	100.00%
Within LSL and USL	79.00%	89.40%	96.25%	99.95%	99.95%





Final Decision

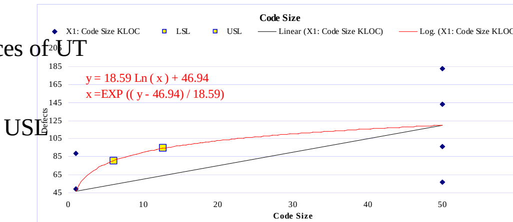
- X_1 ranges from 1 to 12 KLOC of Module Size
  - Only a guideline, not an enforcement
  - 6 KLOC was too stringent an upper limit, and
  - There will also be modules requiring < 6 KLOC, but
  - When breaking the modules into sub modules, aim for sub module size >= 6, E.g.:
    - Two sub modules, each 6 KLoc is better than (2, 10) or (3, 3, 3, 3)
    - Need practitioners to agree this makes sense
- X_2 Test Cases:
  - Ensure there is enough, use the PPM for guidance
- (X_3)^2 Testing Time:
  - Likewise, use the PPM for guidance

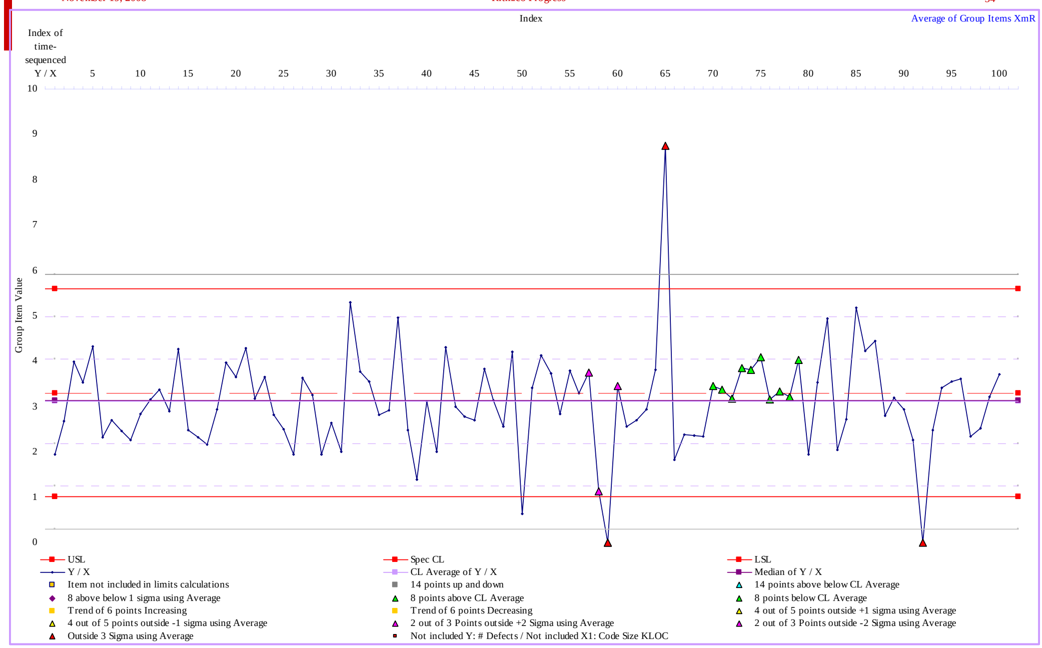
Final Decision

- Simulated PPB ctrl limits:
  - UCL = 5.92 defect density
  - LCL = 0.31
- PPB’
  - UCL = 5.601
  - LCL = 1.005
- Need to also control:
  - # Test Cases

Monte Carlo with X_3 as (X_3)^2 with Optimum Ranges

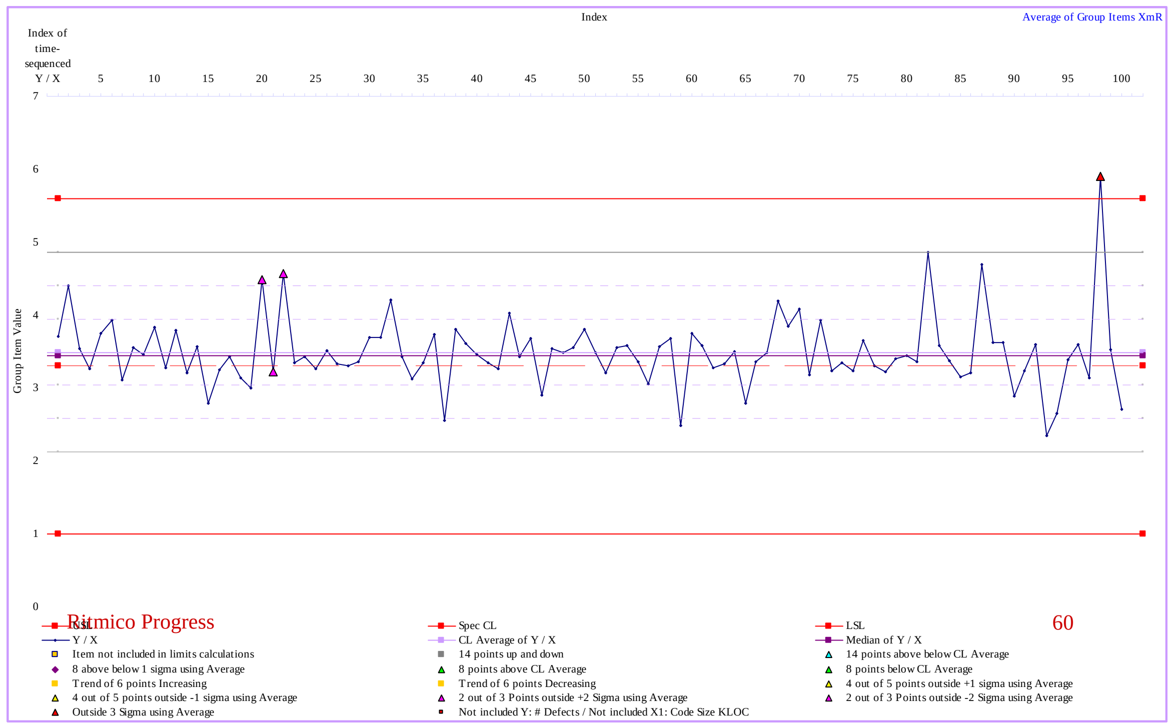
- Simulation of the following:
  - In reality, there will be module Module Size of < 6
  - X_1 ranges from 1 to 12 KLOC of Module Size
  - X_2 ranges from >= 1 Test Cases
    - (Max test cases simulated was up to 428, correlated with file size)
  - (X_3)^2 ranges from >=1 Testing Time
    - (Max testing time simulated was up to 3273 hrs^2, correlated with # test cases)
    - 3273 hrs^2 = (57.2 hrs)^2
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 92.55% >=LSL , 96.85% <= USL
    - 89.40% within LSL and USL





Final Decision

- Module size from 1 to 12 KLoc
- Test Cases variation : Calculated + - 50
- Testing time variation : Calculated + - 10 hrs
- Simulated PPB ctrl limits:
  - UCL = 4.86 defect density
  - LCL = 2.12
- PPB'
  - UCL = 5.601
  - LCL = 1.005



E.g. Module Size 9 KLoc

Size of module (KLoc): 9

QPPO USL	5.601	50.409	Expected defects to be found according to PPO and size of module
QPPO LSL	1.005	9.045	

Y: # Defects	X_1: Code Size KLOC	X_2: # UT Test Cases (Ideal)	X_3: UT Testing Time Hrs^2 (Ideal)	X_3: UT Testing Time Hrs (Ideal)
30.63356	9	250.8809	283.2419	16.82979

UT Test Cases = (26.85* module code size + 9.23)

UT Testing Time^2 =(0.07* Test Cases - 0.79)^2

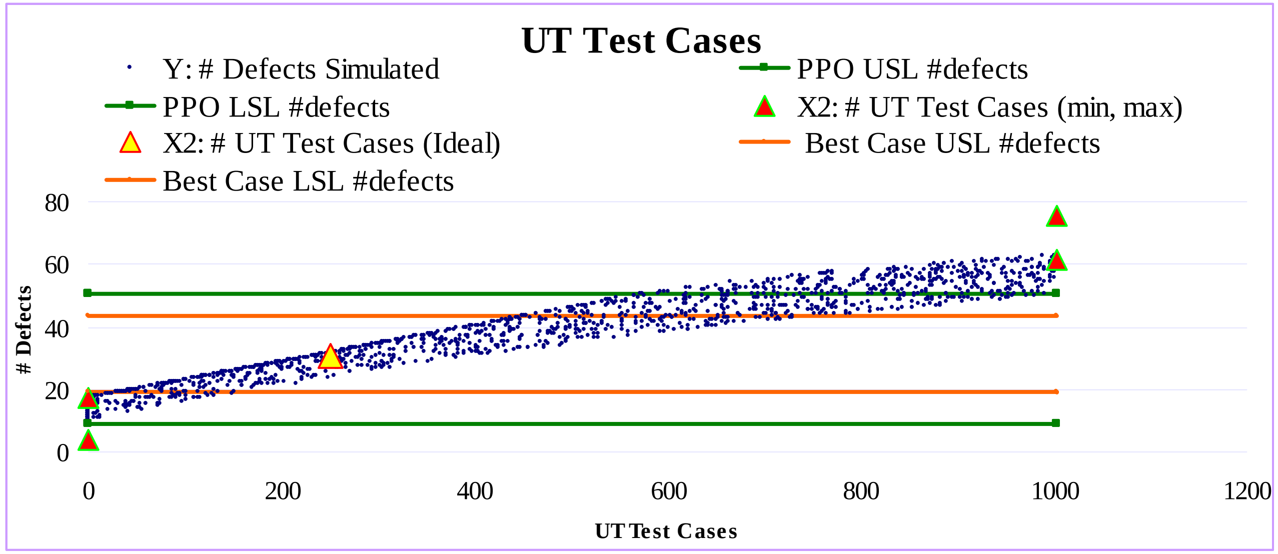
$y = m_1x_1 + m_2x_2 + m_3x_3 + b$

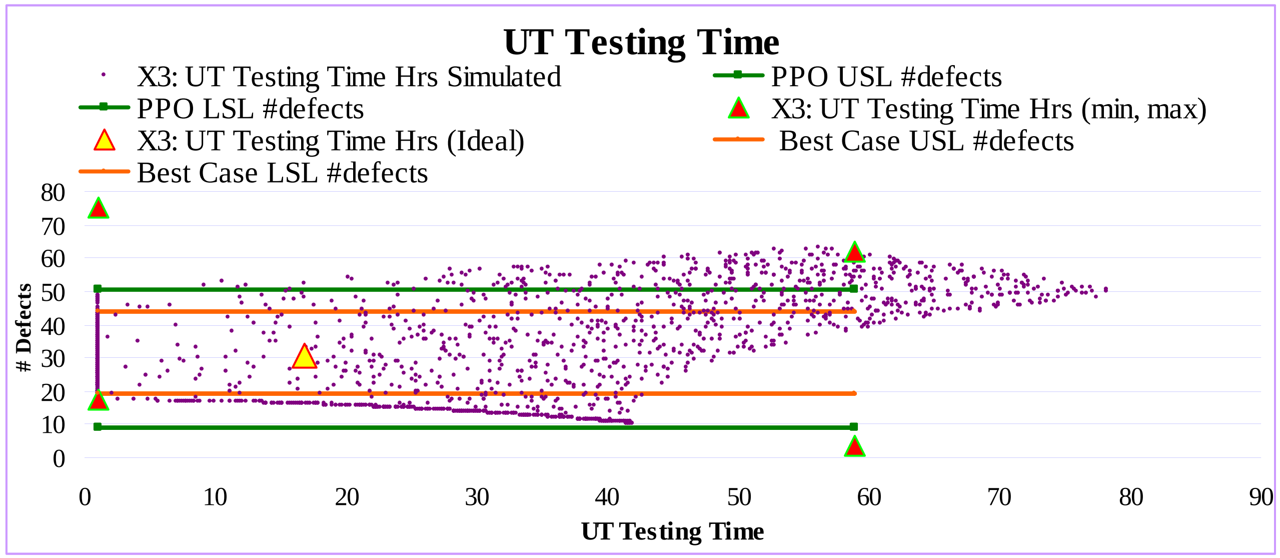
	m_1	m_2	m_3	b
Coefficients	1.912166	0.057942	-0.00393	0

Y: # Defects	X_1: Code Size KLOC	X_2: # UT Test Cases (min, max)	X_3: UT Testing Time Hrs^2 (min, max)	X_3: UT Testing Time Hrs (min, max)
17.26351	9	1	1	1
3.638934	9	1	3469.713	58.90427
75.35195	9	1003.524	1	1
61.72737	9	1003.524	3469.713	58.90427

E.g. Module Size 9 KLoc

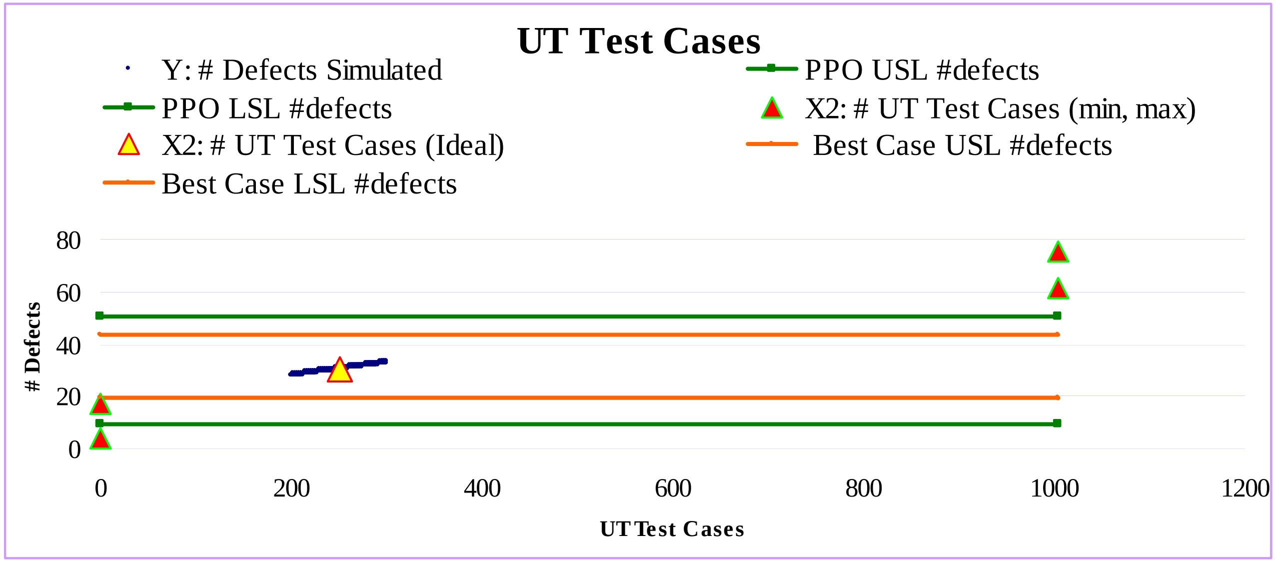
- 81.95% USL — LSL
- Module code size
  - 9 KLoc
- UT Test Cases
  - 1 – 1003
- UT Testing Time
  - 1 – 81 hrs
- PPO is too wide
  - Common problems

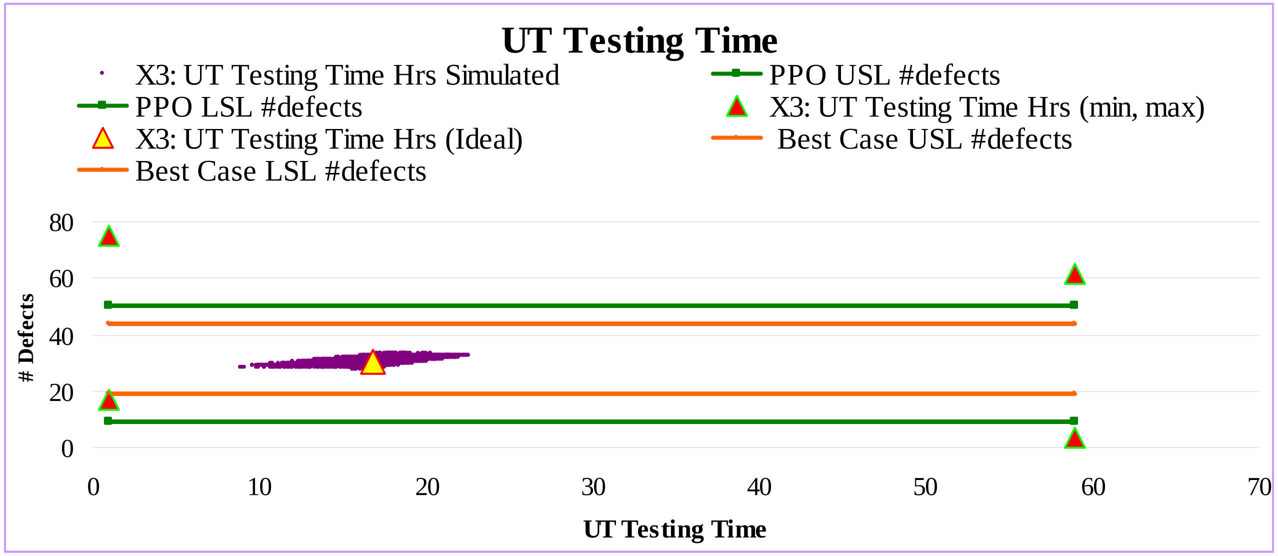




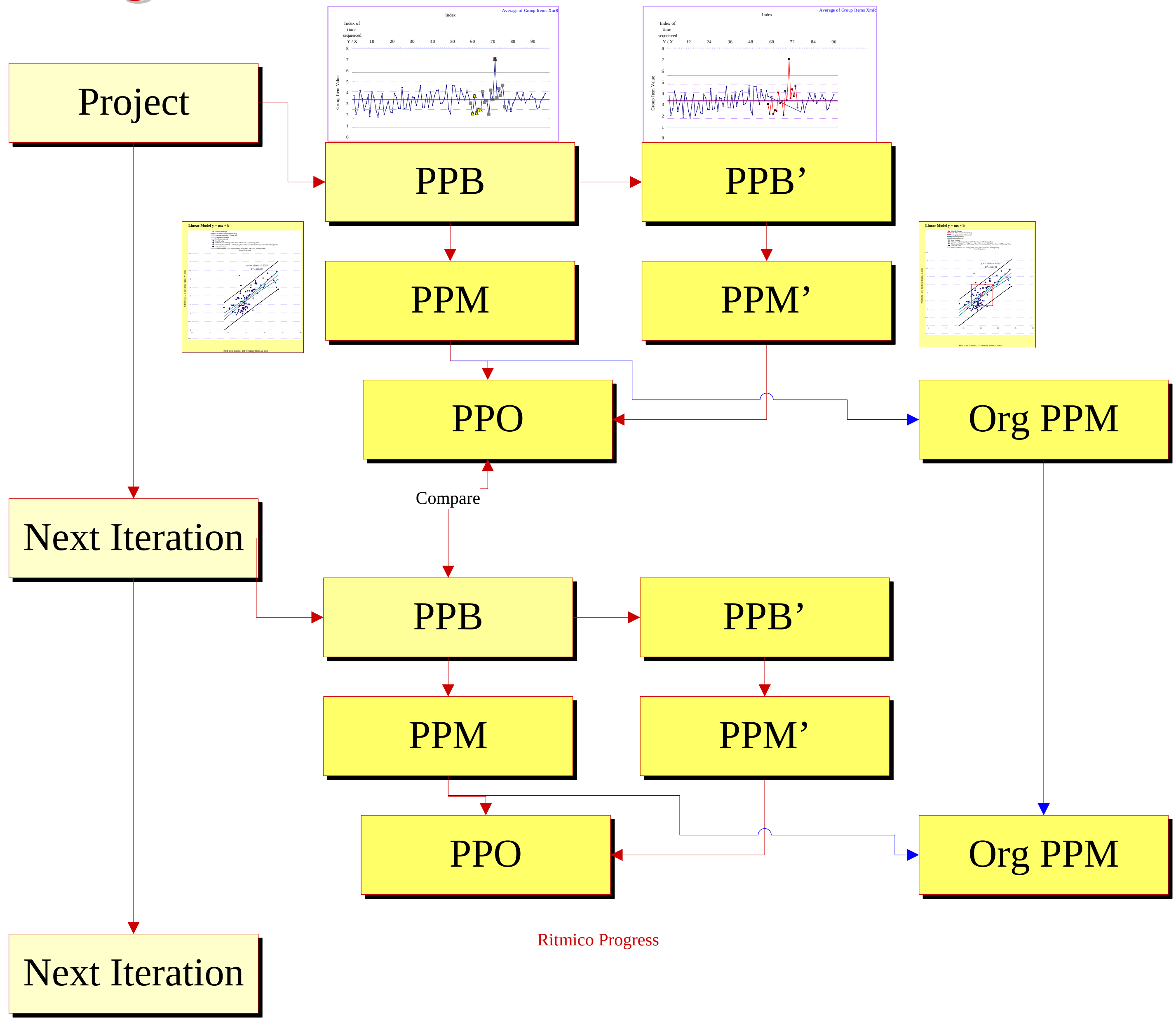
E.g. Module Size 9 KLoc

- 100% USL — LSL
- Module code size
  - 9 KLoc
- UT Test Cases
  - 200 – 300
- UT Testing Time
  - 9 – 22 hrs

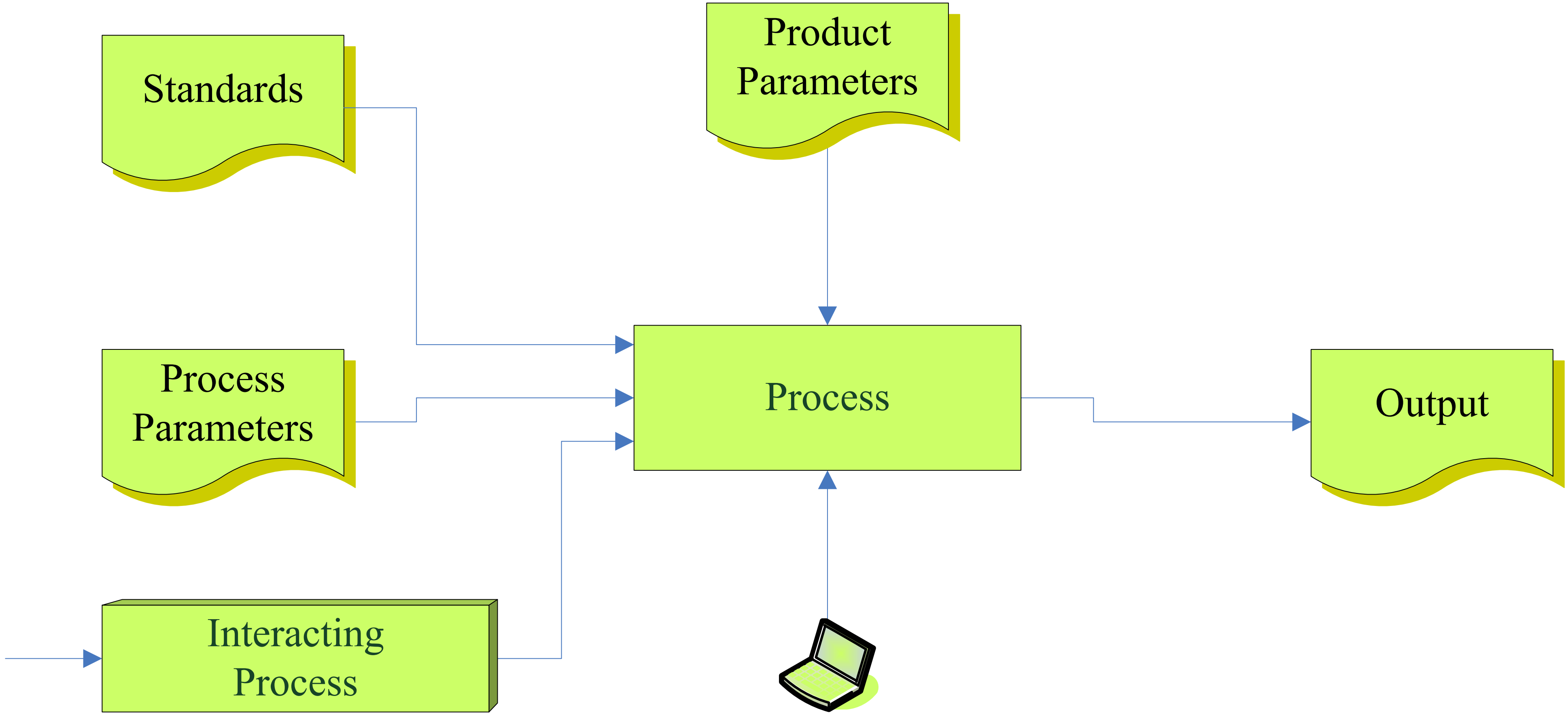




Composing the Defined Process



Process Performance parameters Considerations



Unit Testing Process possible parameters

Standards:
Test case density
Product standards
Test case guidelines

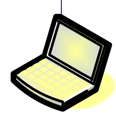
Product Parameters:
Product Module size
Programming language
Product Module complexity
#Changes LOC
Rework Effort
Author Experience
Module development effort

Process Parameters:
Test effort
#Test cases
Test case complexity
Tester training type, time, effectiveness
Tested Frequency
Tester Experience

Unit Testing Process

Output:
#Defects Found
Test Coverage %
Remaining #defects

Interacting Process:
#SQA audit on writing the test cases
#Non-compliances found by QA
Test case review time
#Issues found in test case review



Equipment, Tools, Environment:
Tools:
-WinRunner
-LoadRunner
Methods:
-White box testing
-Black box testing

- Take care of discrete variables which cannot be used in multiple regression.

UT Testing Process

- Selected parameters:
  - Y = #Test Defects Found
  - X_1 = Test Effort - controllable
  - X_2 = #Test Cases – controllable during planning
  - X_3 = Tested Frequency (# times tested) - controllable
  - X_4 = Product module size – controllable during planning
  - X_5 = Development Effort – need more consideration

Questions?

- Ritmico Progress, Rayney Wong
- Rayney@RitmicoProgress.com