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®.
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

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
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”

**Impressive Growth (Number of Employees)**

<table>
<thead>
<tr>
<th>Year</th>
<th>Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>25</td>
</tr>
<tr>
<td>1996</td>
<td>42</td>
</tr>
<tr>
<td>1997</td>
<td>52</td>
</tr>
<tr>
<td>1998</td>
<td>80</td>
</tr>
<tr>
<td>1999</td>
<td>110</td>
</tr>
<tr>
<td>2000</td>
<td>147</td>
</tr>
<tr>
<td>2001</td>
<td>200</td>
</tr>
<tr>
<td>2002</td>
<td>370</td>
</tr>
<tr>
<td>2003</td>
<td>540</td>
</tr>
<tr>
<td>2004</td>
<td>650</td>
</tr>
<tr>
<td>2005</td>
<td>1,200</td>
</tr>
<tr>
<td>2006</td>
<td>2,300</td>
</tr>
<tr>
<td>2007</td>
<td>3,700</td>
</tr>
<tr>
<td>2008</td>
<td>5,000</td>
</tr>
</tbody>
</table>

**Publicly Listed on NYSE**
- December 12, 2007

**Blue-Chip Customers**

- TIBCO
- Oracle
- PeopleSoft
- ABB
- Cisco
- IBM
- ADIDAS
- 3M
- Citibank
- NRI
- The People’s Bank of China
- Sony

**Best-of-Breed Partners**

- Oracle PartnerNetwork
- Siebel Certified Partner
- Microsoft Certified Partner
- IBM
- HP
Facts and History
- Perficient’s Global Delivery Center 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

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPB</td>
<td><strong>Process-Performance Baselines</strong>&lt;br&gt;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.</td>
</tr>
<tr>
<td>PPO</td>
<td><strong>Quality and Process-Performance Objectives</strong>&lt;br&gt;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.</td>
</tr>
<tr>
<td>PPM</td>
<td><strong>Process-Performance Models</strong>&lt;br&gt;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.</td>
</tr>
</tbody>
</table>
### 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 |
Purpose of all improvements are derived from the Business Goals Strategy (BGS).
**VOP-MAR**

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

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Vision</strong></td>
<td>Realizing and understanding the vision, breaking the vision down into its constituent parts.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Objectives</strong></td>
<td>Developing and prioritizing the goals and objectives that must be achieved to fulfill each part of the vision.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Problems</strong></td>
<td>Identifying and analyzing the problems and root causes that are preventing us from reaching the goals, objectives, and vision.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Measures</strong></td>
<td>Determining the measures to understand the extent of the problems and target measures to meet the objectives.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Actions</strong></td>
<td>Developing the actions for resolving the problems and reaching the goals. Improvements are aligned towards the objectives, vision and goals.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Risks</strong></td>
<td>Considering the side effects and costs of the actions in order to mitigate risks and side effects caused by the actions.</td>
</tr>
</tbody>
</table>
BGS at High Maturity

Process Performance Baselines

Process Performance Objectives

Process Performance Models

Critical Process
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
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?

Exception is found

Is it a problem in
the process?

Yes
This is an exception.
Apply Preventive
Corrective Actions.

No
Is it a problem in
the product?

Yes
Is this a common
problem in the product?

Yes
Do not remove exception if product problem cannot be resolved. May require redesign in some modules.

No
This is an exception. Resolve problem in the product.

No
5M, 1E?

Yes
This is an exception. Apply preventive corrective actions. May require Training.

No

?Need more research?
PPB → PPB’

For each exception or set of exceptions, perform a problem solving process to consider improvements to prevent them.
PPB $\Rightarrow$ 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

![Average of Group Items XmR](image)
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 realistic objective.
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.

Average of Group Items XmR

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 see and feel.
- Output Y: #Defects found in a module during UT
- Input X: Module Size KLoc
- Tool needs to be interactive.

Linear Model \( y = mx + b \)

\[
y = 3.0399x + 2.8944 \\
R^2 = 0.8222
\]
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

Linear Model $y = mx + b$

- $y = 0.1106x + 2.7877$
- $R^2 = 0.8155$
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

Linear Model \( y = mx + b \)

- \( m = 1.3858 \)
- \( b = 8.5538 \)
- \( R^2 = 0.7073 \)
Correlation

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

Linear Model \( y = mx + b \)

- \( y = 26.85x + 9.2348 \)
- \( R^2 = 0.9614 \)
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

**Linear Model** \( y = mx + b \)

- \( m = 0.0702 \)
- \( b = -0.7941 \)
- \( R^2 = 0.894 \)

Graph showing a linear regression with data points and a trend line.
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

Linear Model $y = mx + b$

$y = 0.1616x - 0.6357$

$R^2 = 0.6121$
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

Linear Model $y = mx + b$

- $y = 0.1616x - 0.6357$
- $R^2 = 0.6121$

Overall Average
- 1 standard deviation around Overall Avg
- Average predicted Y, linear line
- Confidence Interval
- Prediction Interval
- Pred Y value
- #Defects / UT Testing Time / #UT Test Cases / UT Testing Time
- Not included #Defects / UT Testing Time / Not included #UT Test Cases / UT Testing Time
- Actual Y value
- Linear (Defects / UT Testing Time / UT Test Cases / UT Testing Time)
- Linear Regression
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
When the practitioners are comfortable with the correlations, develop the multiple regression model using the $X_n$ variables.

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>X₁: Code Size KLOC</th>
<th>X₂: # UT Test Cases</th>
<th>X₃: UT Testing Time Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>15.6</td>
<td>455</td>
<td>22.8</td>
</tr>
<tr>
<td>57</td>
<td>27.8</td>
<td>605</td>
<td>54</td>
</tr>
<tr>
<td>54</td>
<td>20.4</td>
<td>593</td>
<td>39.6</td>
</tr>
<tr>
<td>77</td>
<td>18.2</td>
<td>398</td>
<td>29.4</td>
</tr>
<tr>
<td>84</td>
<td>24</td>
<td>697</td>
<td>46.2</td>
</tr>
<tr>
<td>18</td>
<td>7.6</td>
<td>209</td>
<td>16.2</td>
</tr>
<tr>
<td>56</td>
<td>18.4</td>
<td>403</td>
<td>23.4</td>
</tr>
<tr>
<td>95</td>
<td>25</td>
<td>734</td>
<td>47.4</td>
</tr>
<tr>
<td>20</td>
<td>10.78</td>
<td>294</td>
<td>21</td>
</tr>
<tr>
<td>32</td>
<td>7.8</td>
<td>225</td>
<td>17.4</td>
</tr>
</tbody>
</table>

Data shown are just part of the complete set.
Y = 1.958602086 \times X_1 + 0.059436937 \times X_2 - 0.270573847 \times X_3 + 2.251835318

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>X_1: Code Size KLOC</th>
<th>X_2: # UT Test Cases</th>
<th>X_3: UT Testing Time Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence Level</td>
<td>95.00%</td>
<td>0.05</td>
<td>Alpha</td>
</tr>
<tr>
<td>Constant b set to zero?</td>
<td>Non Zero</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y = m_1 x_1 + m_2 x_2 + m_3 x_3 + ... + b</td>
<td>m_1</td>
<td>m_2</td>
<td>m_3</td>
</tr>
<tr>
<td>Coefficients</td>
<td>1.958602</td>
<td>0.059437</td>
<td>-0.2706</td>
</tr>
<tr>
<td>Standard Errors for m_n</td>
<td>0.74684</td>
<td>0.029757</td>
<td>0.22157</td>
</tr>
<tr>
<td>Upper 95.00%</td>
<td>3.44233</td>
<td>0.118555</td>
<td>0.169613</td>
</tr>
<tr>
<td>Lower 95.00%</td>
<td>0.474874</td>
<td>0.000319</td>
<td>-0.71076</td>
</tr>
<tr>
<td>R^2</td>
<td>0.830087</td>
<td>10.55925</td>
<td></td>
</tr>
<tr>
<td>F Statistics</td>
<td>146.561</td>
<td>90</td>
<td>df</td>
</tr>
<tr>
<td>ssreg</td>
<td>49023.80</td>
<td>10034.8</td>
<td>ssresid</td>
</tr>
<tr>
<td>t-observed values</td>
<td>2.62252</td>
<td>1.9974</td>
<td>1.22116</td>
</tr>
<tr>
<td>P-values</td>
<td>0.01025</td>
<td>0.048802</td>
<td>0.225211</td>
</tr>
</tbody>
</table>
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

Polynomial X^2 Model: \( y = m_2x^2 + m_1x + b \)

Output Y: #Defects found in a module during UT

Input X: Module Size KLoc

Polynomial X^2 Regression

\[ y = -0.0448x^2 + 4.3063x - 3.4798 \]
\[ R^2 = 0.8315 \]

Linear Model: \( y = mx + b \)

\[ y = 3.0399x + 2.8944 \]
\[ R^2 = 0.8222 \]
Modeling improved

- Greatest gradient is at 9 KLoc

<table>
<thead>
<tr>
<th>Defects</th>
<th>Code Size</th>
<th>Defect Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.781777055</td>
<td>1</td>
<td>0.781777055</td>
</tr>
<tr>
<td>4.953784418</td>
<td>2</td>
<td>2.476892209</td>
</tr>
<tr>
<td>9.036253663</td>
<td>3</td>
<td>3.012084554</td>
</tr>
<tr>
<td>13.02918479</td>
<td>4</td>
<td>3.257296197</td>
</tr>
<tr>
<td>16.93257779</td>
<td>5</td>
<td>3.386515559</td>
</tr>
<tr>
<td>20.74643268</td>
<td>6</td>
<td>3.45773878</td>
</tr>
<tr>
<td>24.47074945</td>
<td>7</td>
<td>3.495821349</td>
</tr>
<tr>
<td>28.10552809</td>
<td>8</td>
<td>3.513191012</td>
</tr>
<tr>
<td>31.65076862</td>
<td>9</td>
<td>3.516752069</td>
</tr>
<tr>
<td>35.10647103</td>
<td>10</td>
<td>3.510647103</td>
</tr>
<tr>
<td>38.47263532</td>
<td>11</td>
<td>3.497512302</td>
</tr>
<tr>
<td>41.74926149</td>
<td>12</td>
<td>3.479105124</td>
</tr>
<tr>
<td>44.93634954</td>
<td>13</td>
<td>3.456642273</td>
</tr>
<tr>
<td>48.03389947</td>
<td>14</td>
<td>3.43099282</td>
</tr>
<tr>
<td>51.04191129</td>
<td>15</td>
<td>3.402794086</td>
</tr>
<tr>
<td>53.96038498</td>
<td>16</td>
<td>3.372524061</td>
</tr>
<tr>
<td>56.78932055</td>
<td>17</td>
<td>3.340548268</td>
</tr>
<tr>
<td>59.52817801</td>
<td>18</td>
<td>3.307151</td>
</tr>
<tr>
<td>62.17857734</td>
<td>19</td>
<td>3.27256702</td>
</tr>
<tr>
<td>64.73889856</td>
<td>20</td>
<td>3.236944928</td>
</tr>
<tr>
<td>67.20968166</td>
<td>21</td>
<td>3.200461031</td>
</tr>
<tr>
<td>69.5902663</td>
<td>22</td>
<td>3.163223938</td>
</tr>
<tr>
<td>71.88263349</td>
<td>23</td>
<td>3.125331891</td>
</tr>
<tr>
<td>74.08480223</td>
<td>24</td>
<td>3.08686676</td>
</tr>
<tr>
<td>76.19743285</td>
<td>25</td>
<td>3.047897314</td>
</tr>
<tr>
<td>78.22052535</td>
<td>26</td>
<td>3.008481744</td>
</tr>
</tbody>
</table>
Modeling improved

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

Polynomial $X^3$ Model

$$y = m_3x^3 + m_2x^2 + m_1x + b$$

Polynomial $X^3$ Regression

$$y = -0.0033x^3 + 0.1634x^2 - 2.4375x + 12.683$$

$R^2 = 0.6837$

Linear Model

$$y = mx + b$$

Linear Regression

$$y = 0.1616x - 0.6357$$

$R^2 = 0.6121$
Modeling improved

Greatest gradient range:

- 15 – 23 test cases per hour.

<table>
<thead>
<tr>
<th>Defects / Testing Time</th>
<th>UT Test cases / Testing Time</th>
<th>Defects / UT Test Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.347993798</td>
<td>10</td>
<td>0.13479938</td>
</tr>
<tr>
<td>1.24901571</td>
<td>11</td>
<td>0.11354683</td>
</tr>
<tr>
<td>1.258854477</td>
<td>12</td>
<td>0.10490454</td>
</tr>
<tr>
<td>1.357688464</td>
<td>13</td>
<td>0.10443754</td>
</tr>
<tr>
<td>1.525696034</td>
<td>14</td>
<td>0.108978288</td>
</tr>
<tr>
<td>1.743055554</td>
<td>15</td>
<td>0.116203704</td>
</tr>
<tr>
<td>1.989945388</td>
<td>16</td>
<td>0.124371587</td>
</tr>
<tr>
<td>2.246543901</td>
<td>17</td>
<td>0.132149641</td>
</tr>
<tr>
<td>2.493029458</td>
<td>18</td>
<td>0.138501637</td>
</tr>
<tr>
<td>2.709580423</td>
<td>19</td>
<td>0.142609496</td>
</tr>
<tr>
<td>2.876375161</td>
<td>20</td>
<td>0.143818758</td>
</tr>
<tr>
<td>2.973592038</td>
<td>21</td>
<td>0.14159621</td>
</tr>
<tr>
<td>2.981409418</td>
<td>22</td>
<td>0.13551861</td>
</tr>
<tr>
<td>2.880005665</td>
<td>23</td>
<td>0.125217538</td>
</tr>
<tr>
<td>2.649559146</td>
<td>24</td>
<td>0.110398298</td>
</tr>
<tr>
<td>2.270248224</td>
<td>25</td>
<td>0.090809929</td>
</tr>
<tr>
<td>1.722251265</td>
<td>26</td>
<td>0.066240433</td>
</tr>
<tr>
<td>0.985746634</td>
<td>27</td>
<td>0.036509135</td>
</tr>
</tbody>
</table>
Modeling Improved

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

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

Average of Group Items XmR

![Index of time-sequenced Polynomial XmR Control Chart]

Index

Index of time-sequenced Polynomial

X2 13 26 39 52 65 78 91

Group Item Value

-40 -30 -20 -10 0 10 20 30 40

November 19, 2008
The preferred regression formula is used in the multiple regression:

\[
Y = -0.0448X_1^2 + 4.3063X_1 - 3.4798
\]

<table>
<thead>
<tr>
<th># Defects</th>
<th>Code Size KLOC</th>
<th># UT Test Cases</th>
<th>UT Testing Time Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>52.80374</td>
<td>455</td>
<td>22.8</td>
</tr>
<tr>
<td>57</td>
<td>81.63646</td>
<td>605</td>
<td>54</td>
</tr>
<tr>
<td>54</td>
<td>65.73796</td>
<td>593</td>
<td>39.6</td>
</tr>
<tr>
<td>77</td>
<td>60.06585</td>
<td>398</td>
<td>29.4</td>
</tr>
<tr>
<td>84</td>
<td>74.0848</td>
<td>697</td>
<td>46.2</td>
</tr>
<tr>
<td>18</td>
<td>26.66236</td>
<td>209</td>
<td>16.2</td>
</tr>
<tr>
<td>56</td>
<td>60.59941</td>
<td>403</td>
<td>23.4</td>
</tr>
<tr>
<td>95</td>
<td>76.19743</td>
<td>734</td>
<td>47.4</td>
</tr>
<tr>
<td>20</td>
<td>37.73976</td>
<td>294</td>
<td>21</td>
</tr>
<tr>
<td>32</td>
<td>27.38574</td>
<td>225</td>
<td>17.4</td>
</tr>
</tbody>
</table>
### PPM

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>Code Size KLOC</th>
<th>X: UT Test Cases</th>
<th>X: UT Testing Time Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0448X_1^2 + 4.3063X_1 - 3.4798</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Confidence Level: 95.00%  Alpha: 0.05  Constant b set to zero?: Non Zero

\[ y = m_1x_1 + m_2x_2 + m_3x_3 + \ldots + b \]

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( m_3 )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.065908</td>
<td>0.119684</td>
<td>-0.21865</td>
<td>2.054101583</td>
<td>Constant b</td>
</tr>
<tr>
<td>Standard Errors for ( m_n )</td>
<td>0.095014</td>
<td>0.018365</td>
<td>0.229078</td>
<td>2.40169675</td>
</tr>
<tr>
<td>Upper 95.00%</td>
<td>0.254669</td>
<td>0.15617</td>
<td>0.236454</td>
<td>6.825491266</td>
</tr>
<tr>
<td>Lower 95.00%</td>
<td>-0.12285</td>
<td>0.083197</td>
<td>-0.67375</td>
<td>-2.717288101</td>
</tr>
</tbody>
</table>

\[ R^2 = 0.818075672 \]

<table>
<thead>
<tr>
<th>F Statistics</th>
<th>134.9037284</th>
<th>90 df</th>
<th>ssreg</th>
<th>48314.40909</th>
<th>10744.2 ssresid</th>
</tr>
</thead>
</table>

| 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 \( X_1 \).
Y = 1.912166199*X₁ + 0.057942217*X₂ -0.003927848*(X₃)^2 + 0

Constant b (intercept) set to zero

90% confidence level. P-values have improved by using (X₃)^2.

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>X₁: Code Size KLOC</th>
<th>X₂: # UT Test Cases</th>
<th>X₃: (UT Testing Time Hrs)^2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence Level</strong></td>
<td>90.00%</td>
<td>0.1</td>
<td>Alpha</td>
</tr>
<tr>
<td><strong>Constant b set to zero?</strong></td>
<td>Zero</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y=m₁x₁+m₂x₂+m₃x₃+…+b</td>
<td>m₁</td>
<td>m₂</td>
<td>m₃</td>
</tr>
<tr>
<td><strong>Coefficients</strong></td>
<td>1.912166</td>
<td>0.057942</td>
<td>-0.00393</td>
</tr>
<tr>
<td><strong>Standard Errors for mₙ</strong></td>
<td>0.733273</td>
<td>0.027162</td>
<td>0.002075</td>
</tr>
<tr>
<td><strong>Upper 90.00%</strong></td>
<td>3.130698</td>
<td>0.103079</td>
<td>-0.00048</td>
</tr>
<tr>
<td><strong>Lower 90.00%</strong></td>
<td>0.693634</td>
<td>0.012805</td>
<td>-0.00738</td>
</tr>
<tr>
<td><strong>R²</strong></td>
<td>0.955483871</td>
<td>10.45901</td>
<td>Standard error for Y estimate</td>
</tr>
<tr>
<td><strong>F Statistics</strong></td>
<td>651.0676344</td>
<td>91</td>
<td>df</td>
</tr>
<tr>
<td><strong>ssreg</strong></td>
<td>213662.4368</td>
<td>9954.563</td>
<td>ssresid</td>
</tr>
<tr>
<td>t-observed</td>
<td>2.607713</td>
<td>2.133218</td>
<td>1.892992</td>
</tr>
<tr>
<td>P-values</td>
<td>0.010653</td>
<td>0.035598</td>
<td>0.061537</td>
</tr>
</tbody>
</table>

R² = 0.955483871

F Statistics = 651.0676344

df = 91

ssreg = 213662.4368

ssresid = 9954.563

F Distribution = 1.21531E-54

t-critical = 1.661771156
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 $\geq 1$ Test Cases
    - (Max test cases simulated was up to 1448, correlated with file size)
  - $(X_3)^2$ ranges from $\geq 1$ Testing Time
    - (Max testing time simulated was up to 12624 hrs$^2$, correlated with # test cases)
  - $12624 \text{ hrs}^2 = (112.35 \text{ hrs})^2$
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 97.4% $\geq$ LSL , 98.85% $\leq$ USL
  - 96.25% within LSL and USL

Data shown are of one instance of the simulation.
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

\[
Y = 1.912166199X_1 + 0.057942217X_2 - 0.003927848(X_3)^2
\]

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>X1: Code Size KLOC</th>
<th>X2: # UT Test Cases</th>
<th>X3: UT Testing Time Hrs^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.47587698</td>
<td>1</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>88.8097803</td>
<td>1</td>
<td>1500</td>
<td>4</td>
</tr>
<tr>
<td>49.5470133</td>
<td>1</td>
<td>1500</td>
<td>10000</td>
</tr>
<tr>
<td>96.1720207</td>
<td>50</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>56.9092537</td>
<td>50</td>
<td>1500</td>
<td>10000</td>
</tr>
<tr>
<td>182.505924</td>
<td>50</td>
<td>1500</td>
<td>4</td>
</tr>
<tr>
<td>143.243157</td>
<td>50</td>
<td>1500</td>
<td>10000</td>
</tr>
</tbody>
</table>

Remove -ve Y
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 $\leq 12$ KLoc during planning of the modules’ WBS (work breakdown structure)
  - The higher the gradient, usually the higher the productivity

\[
y = 18.59 \ln(x) + 46.94 \\
x = \exp\left((y - 46.94) / 18.59\right)
\]
Optimum range of $X_1$: Code Size

<table>
<thead>
<tr>
<th>Defects</th>
<th>Code Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.83397868</td>
<td>13.8528494</td>
</tr>
<tr>
<td>95.02715121</td>
<td>13.26478748</td>
</tr>
<tr>
<td>94.22032373</td>
<td>12.70168915</td>
</tr>
<tr>
<td>93.50146522</td>
<td>12.22015383</td>
</tr>
<tr>
<td>93.48747921</td>
<td>12.21096848</td>
</tr>
<tr>
<td>91.55774891</td>
<td>11.00759279</td>
</tr>
<tr>
<td>90.42074775</td>
<td>10.35485998</td>
</tr>
<tr>
<td>89.61403259</td>
<td>9.915349726</td>
</tr>
<tr>
<td>88.9882957</td>
<td>9.58732721</td>
</tr>
<tr>
<td>88.66999985</td>
<td>9.42465782</td>
</tr>
<tr>
<td>88.47703144</td>
<td>9.327385204</td>
</tr>
<tr>
<td>86.95887644</td>
<td>8.596311257</td>
</tr>
<tr>
<td>86.15204896</td>
<td>8.231392594</td>
</tr>
<tr>
<td>85.34522148</td>
<td>7.88196495</td>
</tr>
<tr>
<td>84.53839401</td>
<td>7.547370724</td>
</tr>
<tr>
<td>83.73156653</td>
<td>7.226980226</td>
</tr>
<tr>
<td>82.92473905</td>
<td>6.920190501</td>
</tr>
<tr>
<td>82.11791157</td>
<td>6.626424188</td>
</tr>
<tr>
<td>81.3110841</td>
<td>6.345128433</td>
</tr>
<tr>
<td>80.50425662</td>
<td>6.075773856</td>
</tr>
<tr>
<td>79.69742914</td>
<td>5.817853544</td>
</tr>
<tr>
<td>78.89060167</td>
<td>5.570882108</td>
</tr>
</tbody>
</table>

Defects vs Code Size with lines for Linear and Logarithmic models. The models are:

- Linear model: $y = 18.59 \ln(x) + 46.94$
- Logarithmic model: $x = \exp((y - 46.94)/18.59)$
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 $\geq 1$ Test Cases
  - (Max test cases simulated was up to 428, correlated with file size)
- $(X_3)^2$ ranges from $\geq 1$ Testing Time
  - (Max testing time simulated was up to 3245 hrs$^2$, correlated with # test cases)
  - $3245\text{ hrs}^2 = (57\text{ hrs})^2$
- 100,000 simulations of 2,000 instances of UT
- USL=5.601, LSL=1.005
- Result: 99.95% $\geq$ LSL , 100% $\leq$ 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 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 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 $\geq 1$ Test Cases
  - (Max test cases simulated was up to 1444, correlated with file size)
- $(X_3)^2$ ranges from $\geq 1$ Testing Time
  - (Max testing time simulated was up to 11418 hrs$^2$, correlated with # test cases)
- 11418 hrs$^2 = (106 \text{ hrs})^2$
- 100,000 simulations of 2,000 instances of UT
- USL=5.601, LSL=1.005
- Result: 99.95% $\geq$ LSL , 100% $\leq$ USL
  - 99.95% within LSL and USL

\[ y = 18.59 \ln(x) + 46.94 \]
\[ x = \exp\left(\frac{(y - 46.94)}{18.59}\right) \]
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 $\geq 1$ Test Cases
  - (Max test cases simulated was up to 428, correlated with file size)
- $(X_3)^2$ ranges from $\geq 1$ Testing Time
  - (Max testing time simulated was up to 3273 hrs$^2$, correlated with # test cases)
- $3273 \text{ hrs}^2 = (57.2 \text{ hrs})^2$
- 100,000 simulations of 2,000 instances of UT
- USL=5.601, LSL=1.005
- Result: 92.55% $\geq$ LSL , 96.85% $\leq$ 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 $\geq 1$ Test Cases
    - (Max test cases simulated was up to 264, correlated with file size)
  - $(X_3)^2$ ranges from $\geq 1$ Testing Time
    - (Max testing time simulated was up to 2725 hrs$^2$, correlated with # test cases)
  - $2725 \text{ hrs}^2 = (52.2 \text{ hrs})^2$
  - 100,000 simulations of 2,000 instances of UT
  - USL=5.601, LSL=1.005
  - Result: 85.2\% $\geq$ LSL , 93.8\% $\leq$ 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.

<table>
<thead>
<tr>
<th>Module Size Range KLOC</th>
<th>1 to 6</th>
<th>1 to 12</th>
<th>1 to 50</th>
<th>6 to 12</th>
<th>6 to 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSL &gt;=</td>
<td>85.20%</td>
<td>92.55%</td>
<td>97.40%</td>
<td>99.95%</td>
<td>99.95%</td>
</tr>
<tr>
<td>&lt;= USL</td>
<td>93.80%</td>
<td>96.85%</td>
<td>98.85%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Within LSL and USL</td>
<td>79.00%</td>
<td>89.40%</td>
<td>96.25%</td>
<td>99.95%</td>
<td>99.95%</td>
</tr>
</tbody>
</table>

\[
y = 18.59 \ln(x) + 46.94
\]

\[
x = \exp\left(\frac{y - 46.94}{18.59}\right)
\]
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 $\geq 6$, 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 $\geq 1$ Test Cases
    - (Max test cases simulated was up to 428, correlated with file size)
  - $(X_3)^2$ ranges from $\geq 1$ 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% $\geq$ LSL, 96.85% $\leq$ USL
    - 89.40% within LSL and USL

Index of time-sequenced Y / X

<table>
<thead>
<tr>
<th>Index of Group</th>
<th>Item not included in limits calculations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y / X</td>
<td>14 points up and down</td>
</tr>
<tr>
<td></td>
<td>8 points above $CL$ Average</td>
</tr>
<tr>
<td></td>
<td>8 points below $CL$ Average</td>
</tr>
<tr>
<td></td>
<td>Trend of 6 points Increasing</td>
</tr>
<tr>
<td></td>
<td>4 out of 5 points outside $+1$ sigma using Average</td>
</tr>
<tr>
<td></td>
<td>2 out of 3 Points outside $+2$ sigma using Average</td>
</tr>
<tr>
<td></td>
<td>Outside 3 Sigma using Average</td>
</tr>
</tbody>
</table>

Index of Group Item Value

- Y: # Defects / Not included X1: Code Size KLOC
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

<table>
<thead>
<tr>
<th></th>
<th>QPPO USL</th>
<th>QPPO LSL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.601</td>
<td>1.005</td>
</tr>
<tr>
<td></td>
<td>50.409</td>
<td>9.045</td>
</tr>
</tbody>
</table>

Expected defects to be found according to PPO and size of module

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>X₁: Code Size KLOC</th>
<th>X₂: # UT Test Cases (Ideal)</th>
<th>X₃: UT Testing Time Hrs^2 (Ideal)</th>
<th>X₃: UT Testing Time Hrs (Ideal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.63356</td>
<td>9</td>
<td>250.8809</td>
<td>283.2419</td>
<td>16.82979</td>
</tr>
</tbody>
</table>

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

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

**y = m₁x₁ + m₂x₂ + m₃x₃ + b**

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>m₁</th>
<th>m₂</th>
<th>m₃</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.912166</td>
<td>0.057942</td>
<td>-0.00393</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y: # Defects</th>
<th>X₁: Code Size KLOC</th>
<th>X₂: # UT Test Cases (min, max)</th>
<th>X₃: UT Testing Time Hrs^2 (min, max)</th>
<th>X₃: UT Testing Time Hrs (min, max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.26351</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3.638934</td>
<td>9</td>
<td>1</td>
<td>3469.713</td>
<td>58.90427</td>
</tr>
<tr>
<td>75.35195</td>
<td>9</td>
<td>1003.524</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>61.72737</td>
<td>9</td>
<td>1003.524</td>
<td>3469.713</td>
<td>58.90427</td>
</tr>
</tbody>
</table>
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

Project

PPB

PPM

PPO

Org PPM

Next Iteration

PPB

PPM

Org PPM

Next Iteration

Linear Model $y = mx + b$

#defects / UT Testing Time, Y-axis

#UT Test Cases / UT Testing Time, X-axis

Linear Regression

$y = 0.1616x - 0.6357$

$R^2 = 0.6121$

-0.5

0

0.5

1

1.5

2

2.5

3

3.5

4

4.5

0 5 10 15 20 25 30

Overall Average

0.00 Stdev around Overall Avg

Average predicted Y, linear line

Confidence Interval

Prediction Interval

Pred Y value

#defects / UT Testing Time / #UT Test Cases / UT Testing Time

Not included #defects / UT Testing Time / Not included #UT Test Cases / UT Testing Time

Actual Y value

Linear (#defects / UT Testing Time / #UT Test Cases / UT Testing Time)
Process Performance parameters Considerations

Standards

Product Parameters

Process Parameters

Process

Output

Interacting Process
Unit Testing Process possible parameters

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

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

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:

- $Y = \text{#Test Defects Found}$
- $X_1 = \text{Test Effort - controllable}$
- $X_2 = \text{#Test Cases – controllable during planning}$
- $X_3 = \text{Tested Frequency (# times tested) - controllable}$
- $X_4 = \text{Product module size – controllable during planning}$
- $X_5 = \text{Development Effort – need more consideration}$
Questions?

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