Developing Risk Models Requires Mathematics, Domain Knowledge and Common Sense, although not Necessarily in that order

Brendan Murphy
Microsoft Research
Talk Content

Are People a problem?

Models

Metrics and Measurements

Interpretation

EDCC Valencia 2010
Tandem Systems availability 1985 – 1990 (Jim Gray)

- Fault Tolerant Systems
- Highly trained System Managers and Service Engineers
- Systems sent failures to Tandem
- Findings
  - “Operations is a significant cause of outages, second only to software”
  - “Operators are people, they will not be less faulty in the future”
Monitoring DEC Server Systems

• Reality did not match the theory
  – System reality is impact by product quality
    therefore static

• System Reality is periodic
  – Daily
  – Weekly
  – Monthly
  – Quarterly
The underlying trend was operators increasingly causing crashes. System and Network complexity made matters worse.

Murphy Gent 1995 Q&RE

Crashes only represent 10% of all Outages! Availability driven by controlled shutdowns!
Measuring OS on DEC Servers

System reliability increases dramatically in the days following installation Reliability continues to improve over the next 5 months

Murphy Gent 1995 Q&RE
Reliability of Microsoft Software

Software reliability always improve in the months following installation
Improvement not due to software patches
Improvement due to changes in usage profile
Do humans only impact failures due to usage.

• The reliability of the released system and software are heavily dependent on the usage profile.

• Is the underlying software quality based purely on development process?
  – Do human factors have a significant impact?
  – If so how do you feed this into a model?
Metrics and Measurements

“The people who cast votes decide nothing. The people who count the votes decide everything” Joseph Stalin
Measurement Objectives

• Tracking the Project
  – Project Managers can interpret most data

• Identify Exceptions
  – Cross correlation of data to identify exceptions or gaming

• Predictions
  – Completion dates and release quality
  – Verify on past projects.
Problems with tracking Metrics

• Metrics collected through software tools
  – Metrics often a By-Product
  – Tools evolved as do the metrics
    • Making historical comparisons difficult

• People / organizations adapt
  – Peoples behaviour changes based on the problem being addressed
  – People/ Organizations learn from past mistakes
Software Churn
Initial “Gold Standard” Metric

- Lehman and Belady identified its importance in 1960’s
- Measure code rather than Binary churn
- Key attributes
  - Churn frequency
  - Amount of churn
  - Frequency of repetitive churn
  - Late Churn
Churn Correlates to Failures

Use of Relative Code churn measures to predict System Defect Density, ICSE 2005
Nagappan, Ball (Microsoft)

EDCC Valencia 2010
Metrics Monitored in Windows

- **Test Coverage**
  - Arc / Block coverage reflecting testing
  - Testing focuses on problem areas so may be symptomatic rather than a predictor of quality!

- **Bugs**
  - Identified through in-house testing and Beta feedback
  - Difficulty in identifying bug severity!
  - Beta testers not necessarily reflecting user base

- **Code Complexity**
  - OO and non OO metrics
  - Measures often reflect ease of testing!
Metrics Monitored in Windows

• Dependencies
  – Direct and indirect dependencies reflect impact of change
  – Binaries cluster into three categories
• Architectural Layering
  – Does not distinguish hardware interfaces
• Code Velocity
  – Time code exists in the system before being checked into the main branch
  – Process rather than quality measure
• Legacy Code
  – Legacy code is either very good or a potential time bomb!
Organizational Structure Metrics

• Propose eight measures that quantify organizational complexity capturing issues such as
  – Organizational distance of the developers
  – The number of developers working on a component
  – Component changes within the context of an organization

• Organizational structure not taken literally
  – Structure reflects logical rather than actual structure
"The ability to foretell what is going to happen tomorrow, next week, next month, next year. And to have the ability afterwards to explain why it didn’t happen" 

*Winston Churchill on politicians*
Building the Models

- Various methods have been used to build the models
  - Bayesian, Step-wise regression
    - Technique applied does not make that big a difference
- Train and verify the model on a past product, apply to future products
- Initial focus was developing models for Vista
  - Pre usage of People data
Initial results of the Risk Model

<table>
<thead>
<tr>
<th></th>
<th>Win 2003</th>
<th>Win XP SP1</th>
<th>Win XP SP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win 2003</td>
<td>73%</td>
<td>60%</td>
<td>67%</td>
</tr>
<tr>
<td>Win XP SP1</td>
<td>64%</td>
<td>76%</td>
<td>64%</td>
</tr>
<tr>
<td>Win XP SP2</td>
<td>71%</td>
<td><strong>20%</strong></td>
<td>89%</td>
</tr>
<tr>
<td>Win 2003 SP1 RC</td>
<td>78%</td>
<td>96%</td>
<td>70%</td>
</tr>
</tbody>
</table>

EDCC Valencia 2010
Initial Interpretation of Results

• Variation in the objectives of releases
  – Main Releases are feature focused
    • New features create usage issues
  – Service Packs are risk adverse

• Variations between client and server software
  – Management, usage profile and hardware

• Ignoring vital areas
  – Engineers
Developing Models using Vista

• Developed models for predicting product status
  – Achieved accuracy late in the development cycle

• Developed Organizational Metrics
  – Focus is to enhance Churn Metrics

• Verify the predictability of the Organizational Metrics
  – Predict the post release failure rate based on single metrics
Accuracy of Metrics as Predictors

Each attribute characterized by a set of metric
All metrics correlated against failures

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization Structure</td>
<td>86.2%</td>
<td>84.0%</td>
</tr>
<tr>
<td>Churn</td>
<td>78.6%</td>
<td>79.9%</td>
</tr>
<tr>
<td>Complexity</td>
<td>79.3%</td>
<td>66.0%</td>
</tr>
<tr>
<td>Dependencies</td>
<td>74.4%</td>
<td>69.9%</td>
</tr>
<tr>
<td>Coverage</td>
<td>83.8%</td>
<td>54.4%</td>
</tr>
<tr>
<td>Pre-Release Bugs</td>
<td>73.8%</td>
<td>62.9%</td>
</tr>
</tbody>
</table>

EDCC Valencia 2010
“I was gratified to be able to answer promptly and I did. I said I didn’t know” Mark Twain, Life on the Mississippi.
Applied Models to Windows 7

• Tracking Project Status
  – Knowledge gained from cross correlating metrics

• Providing Real Time Data
  – No point identifying historical risk!

• Risk Assessment
  – Adapt to changes in Org structure and project management

• Verification of models once failure profile is known
Problem in Building Risk Models

• Predicting the Future
• Telling good Engineers something they don’t already know
  – Known Problematic area
    • Areas interfacing with hardware
      – Win 7 must work with existing hardware and not all hardware follows specs!
    • New complicated areas
    • Areas with a track record of problems
Summary

• Humans impact reliability
• Building knowledge is more important than models
  – Getting papers into conf/ Journals is far easier than getting engineers to value your results.
• Developing accurate risk models is difficult
  – Ensuring they provide useful and timely data is the real problem
• Writing complex software is difficult
  – So its highly unlikely that a simple model will capture that complexity!
QUESTIONS?