# Presentation of

## “Predictors of Perceived Software Quality”

Lucas Panjer

November 16, 2006

# Motivation

- Can we model and predict perceived software quality from a customer POV?
- Determining influence of a factor on perceived software quality
  - Make targeted enhancements
  - Optimally allocate resources

# Factors / Themes

- Deployment Issues
- Usage Patterns
- Platform
- Hardware configuration
- Target System

# Target System

- Avaya switch application
  - 7 MLOC C/C++
  - Several decades of history
- Databases
  - Trouble ticket DB
  - Change history DB

# Data

- ~2M tickets from 2003
- 100K installed systems
  - 44% have configuration information

# Models of CPSQ

- Rare, high impact events
  - Equipment service
  - Malfunctions w/ software fixes
- Frequent low impact events
  - Technician dispatches
  - Customer calls
  - Alarm reports
- Only measures first three months

# Predictors of CPSQ

- System factors
  - System size
  - Operating System (Propietary, Linux, MSWindows)
  - Ports
  - Deployment time
  - SoftwareUpgrades
- Nuisance factors
  - Geographic location
  - Service contract
  - Missing configuration information (44% of 100K)

# Technique

- Transform some measures
  - log(nPort)
  - log(rtime)
  - LARGE (Binary)
- Regression

# Modeling Software Failures

- Correlation tests indicate independence in all variable, with few predicted exceptions
- Total deployment time is one of the most important predictors.
  - Never upgrade to a dot zero release

| | Estimate | Std. Err. | z-value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | −5.26 | 0.64 | −8.18 | $3 * 10^{-16}$ |
| $\log(rtime)$ | −0.30 | 0.03 | −8.85 | $< 2 * 10^{-16}$ |
| Upgr | 1.38 | 0.15 | 9.01 | $< 2 * 10^{-16}$ |
| OX | −1.18 | 0.17 | −6.75 | $2 * 10^{-11}$ |
| WIN | 1.01 | 0.34 | 2.98 | 0.003 |
| $\log(nPort)$ | 0.36 | 0.08 | 4.37 | $10^{-5}$ |
| $nPortNA$ | 2.03 | 0.58 | 3.49 | $5 * 10^{-4}$ |
| LARGE | 0.52 | 0.20 | 2.67 | 0.01 |
| Svc | 0.57 | 0.18 | 3.11 | .002 |
| US | 0.52 | 0.27 | 1.92 | 0.05 |

**Table 1: Software failure regression results.**

# Other measures

- Support calls, System outages, Technician dispatches, Alarms
  - All show that deployment time improves and is significant
  - Upgrades degrade all measures
  - Larger systems are worse
  - Larger numbers of ports are worse
  - Linux beats embedded OS

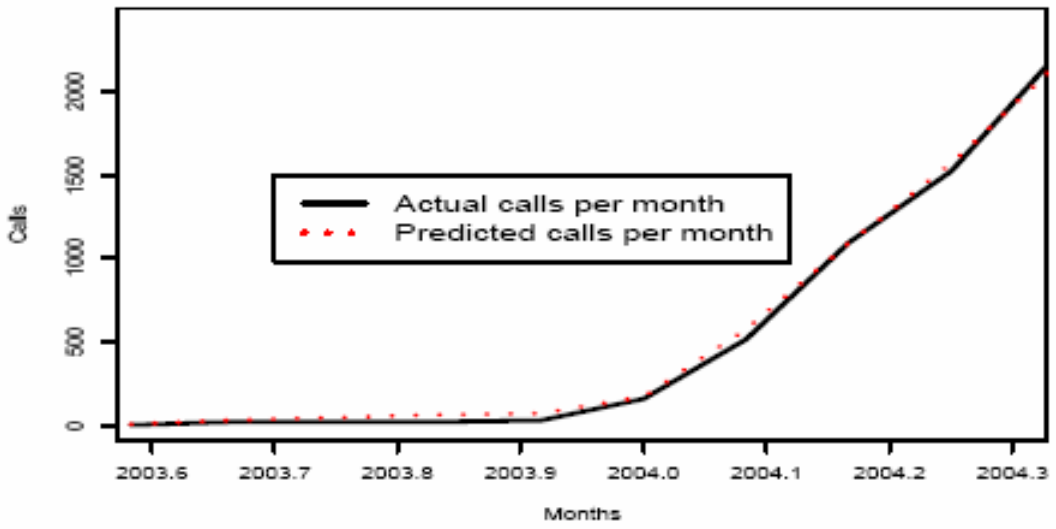


**Figure 2: Prediction of monthly call traffic.**

The two trends are very close to each other indicating that the flow of calls can be predicted fairly accurately. Due to space limitations we do not present full details of predicting the inflow of calls for new and existing systems.

# Contributions

- Technique for modeling software quality
- Application of statistical techniques to software engineering domain
- …

# Positive

- Simple technique, only regression and statistical tests
- Transferable to other data sources, generalizable technique
- Appears to be very accurate

# Negative

- Many potential biases in data
- Difficult to choose predictors
- Validation, description of fit?