HUNTING THE DRAGON-KING: MULTI-DIMENSIONAL TESTING

Robert V. Binder
Co-Director, UIC Internet Assurance Lab

Midwest Verification Day - UIC
September 21, 2013
Overview

• Is software eating the world?
• Bugs
• Black Swans
• Dragon Kings
• Multi-dimensional Testing
• Challenges
"Software is Eating the World"

-Marc Andreessen

Why Software Is Eating The World

By MARC ANDREESSEN

This week, Hewlett-Packard (where I am on the board) announced that it is exploring PC business in favor of investing more heavily in software, where it sees better profit margins. Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves are surprising. But both moves are also in line with a trend I’ve observed, one that makes me optimistic about the future of the American and world economies, despite the recent turmoil in the stock market.

In short, software is eating the world.

More than 10 years after the personal computer, a dozen or so new Internet companies - Amazon, eBay, Twitter, Webvan and Pets.com are still finding ways to shake up the rapidly growing private market for software. They are doing it in ways that even the occasional successful IPO. With the success of Google and LinkedIn, I am optimistic that the model of underpricing software is sustainable, and that the market is not a bubble. But I am also optimistic that the market is not a bubble, and that software is eating the world.

In an interview with WSJ's Kevin Delaney, Groupon and LinkedIn investor Marc Andreessen insists that the recent popularity of tech companies does not constitute a bubble. He also stressed that both Apple and Google are undervalued and that "the market doesn't like tech."
First Generation of Re-Imagination - After 125 Years, Landlines Were Surpassed by Mobiles in 2002

Global Fixed Telephone Lines vs. Mobile Subscriptions, 1994 - 2009

- Fixed Telephone Lines
- Mobile Subscriptions

2002: Inflection Point
Mobiles > Landlines

2006: Landlines Peaked at 1.3B

2009: 4.7B Mobile Subscribers

Embedded and Connected

David Perlmutter, Intel. 2009
By 2015, mobile application development (AD) projects targeting smartphones and tablets will outnumber native PC projects by a ratio of 4-to-1.

Through 2016, the financial impact of cybercrime will grow 10% per year, due to the continuing discovery of new vulnerabilities.

By 2016, 40% of enterprises will make proof of independent security testing a precondition for using any type of cloud service.

Is *Software* really eating the world?

**Methods/Tools since Y2K**
- 2nd gen IDE
- MBT, MDD
- Open Source
- UML 2.0
- Cont. Integration
- Web 2.0
- Agile
- ISO 12027
- LAMP Stack
- Many protocols
- HTML 5
- iOS/Android
- Big Data
- Software Def Netwks


Relative rank from ~2M data points
Is **Software** really eating the world?

<table>
<thead>
<tr>
<th>Source of In-House Developed Software Code for Current Project</th>
<th>Auto</th>
<th>Mil/Aero</th>
<th>Medical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Coded for the Current Project (e.g., C, C++, Assembly, Java, etc.)</td>
<td>56.0%</td>
<td>49.1%</td>
<td>62.8%</td>
</tr>
<tr>
<td>Generated Using Modeling Tools (e.g., UML/SysML, HMI, etc.)</td>
<td>15.5%</td>
<td>16.7%</td>
<td>9.5%</td>
</tr>
<tr>
<td>Leveraged from Previous In-House Projects/Designs</td>
<td>28.5%</td>
<td>34.2%</td>
<td>27.7%</td>
</tr>
</tbody>
</table>

*Engineering Change to Accelerate Product and Systems Development.*
Christopher Rommel, VDC Research. August 2012
**Hardware** is eating the world

### Everything Computes & Connects

- 10x to 100x
  - ten years
- 100x to 1000x
  - twenty years

- Cheaper, better, faster
  - *Disrupt or die*

### Software Methods and Tools

- ?X Productivity
- ?X Quality
- Last ten years?
- Last twenty years?

- Cheaper, better, faster
- “Choose two”
Old bugs, new bugs

- No fundamental or transformative improvements in software
- Bugs still as likely, more code

- End of the uni-stack
- Multi-stack, mobile, connected
- New failure modes

What do you think will happen?
Old Bugs

- Incorrect output
- Missing feature
- Abnormal end
- Missed deadline

• Bohr Bugs
• Mandel Bugs
• Heisen Bugs
Heisen Bugs
New Bugs

• Power Law Models
• Black Swans
• Dragon Kings
Power Law Models

- Many natural phenomena
- "Fat Tails"

Dragon-Kings, Black Swans and the Prediction of Crises. Didier Sornette

http://www.youtube.com/watch?v=FITSbzOvKZI
Reliability Growth

The Lognormal Distribution of Software Failure Rates: Application to Software Reliability Growth Modeling. Robert E. Mullen
Black Swans

- Very rare
- Surprise
- Cannot predict
- Power law outliers
Software Black Swans

- Total eclipse
  - Once every 360 years

- Multi-cycle race conditions in embedded systems
Dragon Kings

- Extreme Outlier
  - Beyond fat tail
- Not “random”
- Latent Positive Feedback
- Phase Shift
  - Earthquakes
  - Material Fractures
  - Epileptic Seizures
  - Market Crashes
Sornett calls these exceptional events *dragon kings* “to stress that we deal with a completely different kind of animal, beyond the normal, whose extraordinary characteristics [have] profound significance.”
Latent Positive Feedback = Dragon King
New Failure Modes: Amazon Cloud Outage

• Amazon web service Elastic Compute Cloud crashes (power failure)

• The Elastic Load Balancing (ELB) system frantically tries to assign workloads to available servers.

• As Amazon’s cloud rebooted, “a large number of ELBs came up in a state which triggered a bug we hadn’t seen before”

• ELB reaction: try to allocate more larger servers.

• Backlog in the “control plane” results.

• Demand from customers in unaffected availability zones continues

• System swamped and crashes again.
What kind of failure was this?

• Why didn’t test find it?

• What would we have to do to find these kind of bugs?
Hic Sunt Dracones
Here be Dragons

- Latent positive feedback
  - External disruptors
  - Malicious attack target
- Partial degradation
  - Provoked panic
- Low testability
  - Multi-stack
  - Field-infeasible
  - Uncontrollable Inputs
Looking under the Model-based Lampost?

Logic errors and incorrect assumptions are inversely proportional to a path’s execution probability.

Boris Beizer

- 9 in 10 Functionality only
- Climbs a tree, not the twigs
- Typically no load test
Multi-Dimensional Testing

**Behavior**
- Realistic variation
- Markov event/data
- Include
  - Fat finger
  - Abuse cases
  - Sneak paths
  - Panics

**Interfaces**
- Drive all interfaces
- Disrupt resources

**Load**
- Vary input rate wrt time

*Not possible without full automation*
Interleave Behavior, Load, Interfaces
### Hunting the Dragon King

<table>
<thead>
<tr>
<th>Method</th>
<th>Common Bugs</th>
<th>Black Swans</th>
<th>Dragon Kings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragile</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Massively Manual</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Test Driven Dev</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Exploratory</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Crowdsourced</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Functional MBT</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Multi-dimensional MBT</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

- 0 You must be joking
- 1 Write if you get work
- 2 Even a broken clock is right...
- 3 One Sigma
- 4 Michael Jordan free throw (82%)
- 5 Sargent Preston
What have we learned?

• *Information Technology* is eating the world

• Common bug rates will **not** decline

• **More** new systems will have dragon kings

• Functional/model coverage MBT **will not** find dragon kings

• Multi-dimensional MBT **can** find dragon kings

• Opportunity for test engineering to **lead**
May you live in interesting times …

Challenges for the MBT Community

Tool Providers
• Start shipping multi-dimensional solutions

MBT Test Engineers
• Learn to lead
• Start hunting your dragon kings
Thank you!
King Effect

- *More* extreme than power law outliers
- “King effect”
Stress versus Fracture Energy

the final rupture associated with a run-away crack releases much more energy

10^{10}

Dragon-Kings, Black Swans and the Prediction of Crises. Didier Sornette
Market Crashes

Dragon-Kings, Black Swans and the Prediction of Crises. Didier Sornette
MBT and Agile

• Don’t waste your time with Fragile

• Agile Pain Points
  • Regression testing gap
  • Superficial testing
    • TD Development
    • Acceptance Tests

• The power of MBT is in the models

• **Test-Model** Driven Development
  • Rapid modeling
  • Some, maybe a lot of help for the RT trap