NONDETERMINISTIC SOFTWARE FOR THE REST OF US

An exercise in frustration by Tomer Gabel
BuildStuff 2018, Lithuania

Follow along!
https://tinyurl.com/nondeterminism
Case Study #1

- Delver, circa 2007
- We built a search engine
- What’s expected?
  - Performant (<1 sec)
  - Reliable
  - Useful
Let me take you back...

- We applied good old fashioned *engineering*
- It was kind of great!
  - Reliability
  - Fast iteration
  - Built-in regression suite
Let me take you back...

- So yeah, we coded it
- And it worked... *sort of*
  - It was highly available
  - It responded within SLA
  - ... but with crap results
- Green tests aren’t everything!
Furthermore

- Not all software *can* be acceptance-tested
  - Qualitative/subjective (e.g. search, social feed)
Furthermore

• Not all software *can* be acceptance-tested
  – Qualitative/subjective (e.g. search, social feed)
  – *Huge input space* (e.g. machine vision)
Furthermore

- Not all software *can* be acceptance-tested
  - Qualitative/subjective (e.g. search, social feed)
  - Huge input space (e.g. machine vision)
  - **Resource-constrained** (e.g. Lyft or Uber)
Takeaway #1

“CORRECT” AND “GOOD” ARE SEPARATE DIMENSIONS
Getting Started

• For any product of any scale, always ask:
  – What does success look like?
Getting Started

• For any product of any scale, always ask:
  – What does success look like?
  – How can I measure success?
Getting Started

• For *any* product of *any* scale, always ask:
  – What does success look like?
  – How can I measure success?

• You’re an *engineer*!
  – Intuition can’t replace data
  – QA can’t save your butt
What *should* you measure?

- (Un-) fortunately, you have *customers*
- Analyze their behavior
  - What do they *want*?
  - What influences your *quality of service*?
- For a search engine...
Takeaway #2

USERS ARE PART OF YOUR SYSTEM
What *should* you measure?

• (Un-) fortunately, you have *customers*
• Analyze their behavior
  – What do they *want*?
  – What influences your *quality of service*?
• For a search engine...
What *should* you measure?

- Query
- Skim
- Decide
- Follow

Paging
- “Not relevant enough”
What *should* you measure?

1. **Query**
2. **Skim**
3. **Decide**
4. **Follow**

**Paging**
- “Not relevant enough”

**Refinement**
- “Not what I meant”
What *should* you measure?

- **Query**
  - "Not relevant enough"

- **Skim**
  - "Not what I meant"

- **Decide**
  - "Bingo!"

- **Follow**

- **Paging**
  - "Not relevant enough"

- **Refinement**
  - "Not what I meant"

- **Clickthrough**
  - "Bingo!"
What should you measure?

- Paging
  - “Not relevant enough”
- Refinement
  - “Not what I meant”
- Clickthrough
  - “Bingo!”
- Bonus: Abandonment
  - “You suck”
Is this starting to look familiar?

It should.
Well now!

- We’ve been having this conversation for *years*
- Mostly with...
  - Product managers
  - Business analysis
  - Data engineers
- *Guess what?*
Well now!

• We’ve been having this conversation for *years*

• Mostly with...
  – Product managers
  – Business analysis
  – Data engineers

• *Guess what?*
What can we learn from BI?

• **Analysis**
  - Be *mindful* of your users
  - Talk to your *analysts*

• Experimentation

• Iteration
What can we learn from BI?

- Analysis
- **Experimentation**
  - Invest in A/B tests
  - *Prove* your improvements!
- Iteration
What can we learn from BI?

• Analysis
• Experimentation
• *Iteration*

- Establish your *baseline*
- Invest in metric collection and dashboards
Takeaway #3

SYSTEMS ARE NOT SNAPSHOTS. MEASURE CONTINUOUSLY
Hold on to your hats

... this isn’t about search engines
Case Study #2

- newBrandAnalytics, circa 2011
- A social listening platform
  - Finds *user-generated content* (e.g. reviews)
  - Provides operational analytics
Social Listening Platform

**Acquisition**
- 3rd party ingestion
- BizDev
- Web scraping

**Analysis**
- Manual tagging/training
- NLP/ML models

**Analytics**
- Dashboards
- Ad-hoc query/drilldown
- Reporting

- A three-stage pipeline
Social Listening Platform

• A three-stage pipeline
• My team focused on *data acquisition*
• Let’s discuss *web scraping*
  – Structured data extraction
  – At scale
  – Reliability is paramount

**Acquisition**
- 3rd party ingestion
- BizDev
- Web scraping

**Analysis**
- Manual tagging/training
- NLP/ML models

**Analytics**
- Dashboards
- Ad-hoc query/drilldown
- Reporting
Large-Scale Scraping

• A two-pronged problem
• Target sites...
  – Can change at the drop of a hat
  – Actively resist scraping!
• Both are external constraints
• Neither can be unit-tested
Optimizing for User Happiness

- Users *consume* reviews
- What do they want?
  - *Completeness* (no missed reviews)
  - *Correctness* (no duplicates/garbage)
  - *Timeliness* (near real-time)
Putting It Together

• How do we measure *completeness*?
• Manually
  – Costly, time consuming
  – Sampled (by definition)
Putting It Together

• How do we measure *completeness*?

• Manually
  – Costly, time consuming
  – Sampled (by definition)

• Automatically
  – Re-scrape a known subset
  – Produce *similarity score*
Putting It Together

• How do we measure completeness?
• Manually
  – Costly, time consuming
  – Sampled (by definition)
• Automatically
  – Re-scrape a known subset
  – Produce similarity score
• Same with correctness
Putting It Together

• Targets do not want to be scraped
• Major sites employ:
  – IP throttling
  – Traffic fingerprinting
• 3rd party proxies are expensive
Putting It Together

• What of *timeliness*?
• It’s an *optimization problem*
  – Polling frequency determines latency
  – But polling has a cost
  – “Good” is a *tradeoff*
Putting It Together

- So then, timeliness...?
- First, build a cost model
  - Review acquisition cost
  - Break it down by source
- Next, put together SLAs
  - Reflect cost in pricing!
  - Adjust scheduler by SLA
Recap

1. “Correct” and “Good” are *separate dimensions*

2. *Users* are part of your system

3. Systems are not snapshots. *Measure continuously*

*Image: Confused Monkey, Michael Keen (CC BY-NC-ND 2.0)*
QUESTIONS?

Thank you for listening