Culture and the Games
People Play

Roy Rapoport

rsr@netflix.com  @royrapoport
November 18, 2015
SHALL WE PLAY A GAME?
What We Want
(And How We Get It)
What We Want
(And How We Get It)

Outcomes

Actions

Decisions

What environment says

What environment does
What We Want
(And How We Get It)
What We Want
(And How We Don’t Get It)
What We Want
(And How We Don’t Get It)
Test #1
Attendance Award
A Word About Netflix …

Culture

- Clear Priorities
  1. Innovation
  2. Availability
  3. Cost
- Hire smart, experienced, people
- Get out of the way
- Anti-process bias
In Practice ...
The Before Time

Dozens of SSL Certificates

Decentralized

Kept Expiring

Hilarity would ensue

Amazon Resources

“No Preset Limit”

You know when you hit it

Hilarity would ensue
The Before Time

Well-developed Developer Ecosystem

Service Discovery

DB Client

Credentials Management

Memory Object Cache

Server Infrastructure

Telemetry

You wanted that for Java, right?
The Before Time

Just moved from IT/Ops

Formally tasked with SSL cert issue as quarterly goal

Limits issue “tacked” on

“Effective” in Python

Didn’t know Java
No Problem!

Ported necessary libraries to Python
Boss was dubious. **Really** dubious.
Ran into security problem
Introducing Jay
Democratized Innovation

What would you say you do around here?

Story Time: Shark Tank
Conceived by Reliability Engineer
Remote Telemetry Network
Teams involved:
  Reliability Engineering
  Insight Engineering
  Performance Engineering
  Some others …

“Proof-of-concept work on Ansible configuration management for Gulo and Hammerhead.”
Avoid Zero-Sum Games
Stack ranking
Fixed bonus / raise pools
No ranking/quantifying
Reviews != raises
Decentralize collaboration
Align goal

I want:
Collaboration and Selflessness
Act In Netflix’s Best Interests
Test #2
Early Birds, Late Worms
I want:

Decentralized Innovation
Autonomy and Independence

Bets and Risk Tolerance:
a Story of Failures
## Losing Bets

### 18 month report card (estimated)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Monkey</td>
<td>Success</td>
</tr>
<tr>
<td>Howler Monkey</td>
<td>Success</td>
</tr>
<tr>
<td>Exploit Monkey</td>
<td>Failure</td>
</tr>
<tr>
<td>Python</td>
<td>Success</td>
</tr>
<tr>
<td>Service SLA Dashboard</td>
<td>Failure</td>
</tr>
<tr>
<td>Alert Outsourcing</td>
<td>Success</td>
</tr>
<tr>
<td>Alert Response Analytics</td>
<td>Failure</td>
</tr>
<tr>
<td>Alert Gateway</td>
<td>Success</td>
</tr>
<tr>
<td>Alerting GUI</td>
<td>Success</td>
</tr>
<tr>
<td>Latency Monkey Adoption</td>
<td>Fizzle</td>
</tr>
<tr>
<td>Stateful Alerting</td>
<td>Failure</td>
</tr>
<tr>
<td>Open Application Alerting</td>
<td>Failure</td>
</tr>
</tbody>
</table>

**50% Failure Rate**
I want:

Decentralized Innovation
Autonomy and Independence

An Engineering Manager Walks Into an Override Bar …
The Override Bar

Asgard: Full-fledged cloud orchestration
GUI-driven
Region-and-account specific
The Override Bar

Four regions
Eight accounts
Hundreds of clusters
A Bold Proposal
Totally duplicates functionality
Customized fit
Failed the override bar:
  Am I sure this is the wrong thing?
  If I’m right, will this be very expensive for us?
The Override Bar

Accomplished predicted results
Massively simplified operational processes
Improved resiliency and velocity
Unpredictable results
Used by other teams
Inspiration
Will retire
I want:

Decentralized Innovation
Autonomy and Independence

Spheres of Autonomy:
Staying DRI
Concentric Spheres of Autonomy
Spheres of Autonomy: A New Model

- Fang’s Sphere of Autonomy
- Roy’s Sphere of Autonomy
- Josh’s Sphere of Autonomy
- Yury’s Sphere of Autonomy
- Neil’s Sphere of Autonomy
- Reed’s Sphere of Autonomy
Spheres of Autonomy: 
A New Model

Set context. Not control.
Spheres of Autonomy: A New Model

Keeping Peers DRI
Test #3
Lucy and the Ball
Literally* no downsides!

Predictability tradeoffs
Locality optimization
Duplication
Duplication

* For very non-literal definitions of the word “literally”
Agility vs Predictability

Neither is bad
Probably need some of both
Do you know how much you want?
Do you have it?
Optimize for agility
Constrain predictability

Some things are important to predict
- Public KPIs
- Big product plans

Fewer are important than you may think
Locality Optimization

Or lack thereof

If a Thing can be built anywhere
Not always in the best place
Extra work
Locality Optimization
Or lack thereof

Story Time: Scryer
Scryer: Start State

Real-Time Telemetry System
2 weeks of data
Scryer: Goal

Real-Time Telemetry System
2 weeks of data

Signal Predictions
Today

Product
Value-add Process
 Predictor
Scryer Architecture, v1

Real-Time Telemetry System
2 weeks of data

Telemetry Extractor

Telemetry Persistence
4 weeks of data

Signal Predictions
Today

Product

Value-add Process

Waste of Time

Pain the [REDACTED]
The Thing Is …

Real-Time Telemetry System
2 weeks of data

Cloud Storage
All telemetry, forever

ETL
Scryer Architecture, v2

Real-Time Telemetry System
2 weeks of data
Predicted Signal Today

Cloud Storage
All telemetry, forever

ETL

Product

Value-add Process

Predictor
Test #4
Making Friends $100 At a Time
"I only want to ride the wind and walk the waves, slay the big whales of the Eastern sea, clean up frontiers, and save the people from drowning. Why should I imitate others, bow my head, stoop over and be a slave?" - Lady Triệu
rsr@netflix.com
@royrapoport

Attributions:
https://www.flickr.com/photos/cseeman/
http://www.flickr.com/photos/watchsmart
http://www.flickr.com/photos/yaketyyakyak/
https://www.flickr.com/photos/gfreeman23/
https://www.flickr.com/photos/dotcode
https://www.flickr.com/photos/tlindfors

And the Rands Leadership Slack