We The Few

Critical Team Composition and Responsibilities For Day 2 Operations
Who Am I?

Keith Strini ...

Field Facing Solutions Architect that serves as a technology analyst for the US Department of Defense and Intelligence communities. I architect, develop and field information systems across the Joint Services both CONUS and OCONUS (Korea, Japan, Europe, and the Middle East) and NATO
End Vision
Not End Vision
Options Getting to End Vision
Release Engineering Stratification

App Operator

Noob Dev Team

App Profiles
Prod Manifest
Unit/Smoke Test Pipelines

Platform Product Manager

Developer Enablement

Release

Platform Reliability

Cadence Calendar

Self Service Deployment

Release Repository
unit/smoke security uat

pass

fail
Platform Reliability
Coordination Point For All Platform Environment Changes.

- Creates/Coordinates Cadence Meeting
- Continuously Develops Resiliency Probes based on Post Mortems
- Maintains Environment Parity
- Enforces Strict Runtime Version Control
- Communicates Environment Adversity
- Creates/Coordinate Resiliency Exercise
- Instruments Distributed Tracing in Ops

Release Engineering
Coordination Point For All New Releases.

- Attends Final Pre-Release Demo of Apps
- Verifies Release Artifacts
- Coordinates Initial Release Date
- Collaborates on Downstream Environment Triage

Developer Enablement
Coordination Point For All New Development Efforts.

- Creates/Coordinates Platform on boarding meeting
- Provides Latest Information about Platform Environments
- Provides tooling around Dev Services, CI/CD environment, governs code repositories, etc
Troubleshooting Complex System Failure
Simple System Failure (Daddy I want to build a car)
Simple Complexity
## Failure is Inevitable, Hope is Not a Strategy

<table>
<thead>
<tr>
<th>Developers</th>
<th>Dev Enablement</th>
<th>Release</th>
<th>Operations</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit/Smoke</td>
<td>Contract</td>
<td>Feature Flagging</td>
<td>Canary</td>
<td>Distributed Tracing</td>
</tr>
<tr>
<td>Been There, Done That, Got the swag!</td>
<td>Focuses Developers on Contract Based Testing For Integration</td>
<td>Decides on Feature Maturity From a Stability Perspective</td>
<td>Care And Feeding Of New Releases To Ensure Early Intervention</td>
<td>Deep Understanding of How New Efforts Deploy Into Operations</td>
</tr>
</tbody>
</table>
Decoupled Integration – Contract Testing

• If Speed Is What you Want, End-To-End Testing is not how you get there.
  • Getting feedback…this week?
• Are you mocking me?
  • Single Source of Truth…
  • Verifying the Goods
• Isolation testing of Single Services (Provider or Consumer)
• I do not think that means what you think it means (Semantic Testing)

• Complexity From Simplicity, Not Complexity From Complexity
• Test Data…let’s not ignore the elephant in the room
• Stability, Stability, Stability… we’re talking Operations not Science Experiments
• Ah Sunsets…
  • Paying off Technical debt by subtraction and addition
• You get me… you really get me
  • Consumer defined APIs
Maintaining Operational Velocity – Feature Flagging

- I dunno. You tell me what you want.
  - Non Techs getting in on the Action
- Ok so most of it works but I gotta send it back?
  - Beauty of context encapsulation
  - Waiting for a feature like you
- I see how we do 1 app but how do I manage 1000s?
- So what if we don’t know exactly what our users want?
- Ah Sunsets...
  - Paying off Technical debt by subtraction and addition
Predictive Fire Fighting In Operations – Canary/Distributed Tracing

- Almost trust you
- Canaries – Profiling the CPU, memory, disk usage, cache synch
  - Rollback/Roll Forward Strategies
  - Blue/Green Deployments
    - The case of stateless
    - The case of stateful (transactions, migrating data)
    - Infrastructure Isolation
- A/B testing

- It not you, its me
- Distributed Tracing
  - Yes we are talking scale here
  - But that’s a lot of instrumentation
  - Correlation is tough
  - Good definitions of SLO/SLIs
- Threshold tuning
Operations as the Caretaker of Code?

• Your baby is ugly, Our baby is cute
  • Platform as Product helps align our interests
  • Automation helps us be responsive as a team to our end users
• Lot’s of up front pain is better than chronic pain indefinitely
  • Success as defined by rhythm
Get Off My Lawn!

• Change Inherently Creates Failure
  • Alignment of Values
• One Team One Fight
  • Joint SLOs
    • Platform SLIs
    • Application SLIs
    • Instrumentation

• Growing up is hard to do
  • Graduating Product Teams to Self-Service Deployments

Growing up is hard to do
Growing Up Is Hard to Do

- Resiliency Exercises as the litmus test
- Communicating the attitude that Stability is a team sport

- Starting the Cycle Over
  - Capturing Lessons Learned from every class
  - Knowledge Transparency aides the greater team
In Conclusion

“When learning something new you have to practice going slow, if you want to eventually go fast forever”
Questions?