SRE For Enterprise-Scaled

How To... Part 2

Dinesh Sekar
SRE Evangelist & Transformation

 Sekardinesh
Enterprise-scaled organizations; the older the organization is, the older the systems, process, culture are...

“Change is the only constant”

- Enterprise Need for SRE
- SRE Transformation Framework
- Engaging SRE across lifecycle

Dinesh Kumar Sekar

SRE Transformation & Competency Development

- Core SRE Transformation Management
- Driving SRE adopting across Organization
- Up-skill, re-skill and re-purpose teams aligning to SREs practices
- Lead SRE Academy
- Co-Lead SRE Community of Practices
Many org. are NOT Google (recap from 2020)

“It is a truth universally acknowledged that systems do not run themselves. How, then, should a system—particularly a complex computing system that operates at a large scale—be run?”

“It’s all well and good to be inspired by core technology serviced organizations!!!

Many enterprise-scaled organizations are NOT GOOGLE!

“Hope is not a strategy”
Few Challenges Faced By Enterprise-scaled

- Legacy environments and/or systems.
- Majority of systems have low DevOps CI/CD maturity.
- Workforce, Silo organization structure and silo culture (“you build it, you run it”)
- Workforce and upskilling/repurposing (e.g. operations adopting engineering)
- Workforce and more workforce to handle more incidents
- Legacy processes and controls (e.g. ops cannot touch codes)
- Buy-in from stakeholders on ‘Error-Budget’
What then..?

- Modernizing critical legacy applications is hard...
- Adopting SRE principles and practices is hard...

**But...**

- Glassdoor on SRE - > 54,000 open positions (8th June 2020). Both from tech firms (GitHub, Spotify, Apple) and non-tech companies (GE, Chase, Walmart, etc., )

**WHY ?**
The Need for SRE

1. When was the last time you / your team worked on a service incident?
   - This week > 54%

2. How many service incidents do you work / handle per week?
   - 5-10 incidents > 41%

3. Based on your experience what do you think is the main cause of these incidents?
   - Poor quality of changes to code, data, workload, infrastructure etc > 46%

Sample Size of approx. 1000 Participants across Asia
The Big Question…

How could an enterprise-scaled organization adopt SRE?

How could we integrate the SRE discipline into existing enterprise-scaled?
Guiding Principles for SRE (adopted from ITIL4)
SRE Transformation Framework

**Assessment**
- Assess current state (*e.g.* Enterprise ADOC)
- Develop SRE Champions & SRE Transformation / Change Ambassadors
- SRE roadmap (*Observability, Operability, Reliability, Scalability & People*)

**Initiation**
- Identify & mapping critical-user-journey
- Defining SLOs & SLIs
- Defining SRE Operating & Engagement Model

**Execution**
- Execute observability, operability, reliability and scalability action items
- Finalize and implement SREs ways of working
- SLOs, SLIs & Error Budget data in Dashboard
- Value realization, # of incidents, MTTD, MTTR, MTBF, Incidents that needs manual interventions and etc.
- Re-assess SRE Maturity

**Realization**
### Engaging SRE Across Lifecycle

<table>
<thead>
<tr>
<th>Design</th>
<th>Build</th>
<th>Release</th>
<th>Post Release</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Engineering / Product Team</strong></td>
<td><strong>SRE</strong></td>
<td><strong>Benefits SRE Provide at each stage</strong></td>
<td><strong>Benefits of engaging SRE</strong></td>
</tr>
<tr>
<td>• Build Requirement</td>
<td>• Provide input for scalable, resilient system</td>
<td>• Minimize future disputes over design choice post-production</td>
<td></td>
</tr>
<tr>
<td>• Decide on Scope</td>
<td>• Establish SLO / SLI and SRE by design</td>
<td>• Software engineers designing IT operations with rigorously focus on error budget and SLOs</td>
<td>• Establishes clear ownership of production resiliency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Build product / solution</td>
<td>• Confirm metrics, instrumentation can be captured accurately</td>
<td>• High quality of release &amp; deployment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Define error budget</td>
<td>• Low operational burden</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Go-to-market strategy</td>
<td>• Release Automation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Validate the product</td>
<td>• Monitor SLIs and delivery SLOs, plan capacity, emergency response, etc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Manage error budget</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Prioritize, define and iterate over feedback</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
THANK YOU!

Meet me in the Network Chat Lounge for questions