SAFE, ITIL AND DEVOPS

SCOTT PRUGH
CHIEF ARCHITECT AND SVP ENGINEERING
@SCOTTPRUGH

SAFE SUMMIT
10/03/2018
Industry Leading, Innovative, Configurable Business Support Solutions

**CSG: Who Are We?**

- **3,300 Employees**
  - Around the Globe

- **35+ Years**
  - Supporting the World's most respected CSPs

**CSG in North America**

SaaS Based Customer Care and Billing
- ~62M Subscribers
- 150k Call Center Seats
- ~6B External Transactions/month
- 40 Dev. Teams & 1000 Practitioners
- ACP: ~20 Technology Stacks: JS to HLASM
  - Integrated Suite of 50+ applications

Challenges: Time to Market, Quality: Software & Operations
— OUR JOB AS IT LEADERS

Build Great Organizations That Engineer Great Products That Delight Our Customers and Employees
PROBLEMS

- Misaligned Incentives and Adversarial Relationships
- Queues and Batches Inhibit Flow
  - Queues & Tickets Make IT Work Miserable
  - Large Batches Increase Failure and Rework
- Focus on Process vs Customer Value
- Operational Quality Continues to be a problem
OPPOSING FORCES

Development (SAFE/SDLC)

Speed!
Better/faster environments!
Better SR tools!
We hate CRQs!

Change Management
Release Management
Production Operations
PMD

Operations (ITIL)

Stability!
Better code!
Better Agile tools!
We hate CRQs!

Customers:
High Quality Features Quickly!

Speed! Stability!
Better code! Better faster environments!
Better Agile tools! We hate CRQs!
QUEUES & TICKETS MAKE IT WORK MISERABLE

WAITTIME=%BUSY/%IDLE

50/50=1
80/20=4
90/10=9

What's your number?
The de-facto technique in Software and Operations is to organize by role/function and place large amounts of WIP in queues as tickets between groups.

Queues don’t behave as we expect:

As utilization increases, cycle time increases exponentially → WaitTime=%Busy/%Idle

<table>
<thead>
<tr>
<th>Queues Create</th>
<th>Queues Don’t Scale or Learn!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longer Cycle Time</td>
<td>Admitted</td>
</tr>
<tr>
<td>Increased Risk</td>
<td></td>
</tr>
<tr>
<td>More Variability</td>
<td></td>
</tr>
<tr>
<td>More Overhead</td>
<td></td>
</tr>
<tr>
<td>Lower Quality</td>
<td></td>
</tr>
<tr>
<td>Less Motivation</td>
<td></td>
</tr>
<tr>
<td>Accountability Loss</td>
<td></td>
</tr>
<tr>
<td>Knowledge Loss</td>
<td></td>
</tr>
<tr>
<td>Delayed Learning</td>
<td></td>
</tr>
</tbody>
</table>

Adapted from Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development
OUR TOOLS

Foundational
- Lean and Systems Thinking
  - Strive for Flow, Continuous Learning and Adaptation
- Courage & Transformational Leadership
- Change Incrementally and Small Batches

Principles
- Build/Run Teams & Service Owner Value Streams
- Ownership & Engineering Excellence: CI, CD, Everything is Code
- Operations is an Engineering Problem

Practices
- Single Backlog
- Localized Change
- Incident Swarming
- Impact Scoring and Post Incident Reviews
- Improve/Consolidate Work Management Systems
BACKGROUND & FOUNDATIONAL THINKING
**Functional Archetype:** Role specific organizations “locally optimize” around their structure and role instead of globally optimizing the flow of business value.

Queues

WaitTime= (%Busy)/(%Idle)

Queues delay feedback and learning

Queues help enforce I-Shaped resources

---

*CSG Pre-2012*
**Agile Archetype:** Cross functional, self-empowered agile teams optimize the SDLC value-stream and remove impediments to do so.

Big product/feature a client wants
## DEVOPS JOURNEY IN METRICS

<table>
<thead>
<tr>
<th></th>
<th>Begin</th>
<th>End</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel. Impact</td>
<td>507</td>
<td>85</td>
<td>-83%</td>
</tr>
</tbody>
</table>

Agile: 2012-2016
Facts
1. 98% of incidents outside release
2. 92% of incidents fixed by ops
## DEVOPS JOURNEY IN METRICS

<table>
<thead>
<tr>
<th></th>
<th>Begin</th>
<th>End</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel. Impact</td>
<td>507</td>
<td>85</td>
<td>-83%</td>
</tr>
<tr>
<td>Inc./mo.</td>
<td>1640</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Agile: 2012-2016

DevOps: 2016-*
LET’S COMPETE ON THE SAME TEAM

Let’s be fast, stable and secure!
Let’s compete on the same team!

Development  Operations

Let’s be fast, stable and secure!
Let’s compete on the same team!
BUILD/RUN(DEVOPS) TEAMS & SERVICE OWNER VALUE STREAMS
The Service Owner is the single Transformational Leader accountable for the: end to end construction, operation, SLAs, customer experience and stewardship of business value for a product or set of services.
**PRINCIPLE: BUILD/RUN(DEVOPS) TEAMS & SERVICE OWNER VALUE STREAMS**

- 1-N cross functional DBTR teams (9+/-2)
- Strong cross functional leadership
- Sprint cadence and Kanban
- Single backlog for all activities
- Stewardship and Engineering Excellence: CI, CD, Everything is Code

Measure: Deploy Freq, Lead Time for Change, MTTR, Change Failure Rate
- Swarm for Incident Response
- Operations is an Engineering Problem: Concerns continually worked into backlog
**PRACTICE: SINGLE BACKLOG**

---

**Traditional Dev and Ops**

- Dev Board
  - F: Bus
  - F: Bus
  - F: Bus
  - F: Bus

- Ops Board
  - F: Ops
  - Change
  - Incident
  - F: Sec

- Invest%

---

**Collapse Dev + Ops teams + process**

- DevOps Board
  - F: Bus
  - F: Ops
  - Change
  - Incident
  - F: Sec

---

**Product Management**

- **Lets build more features!**

---

**Product Management**

- **Why can’t I build more features?**

---

**Service Owner and team own bus. features, ops. features/concerns from one backlog.**

**Integrate ops design into construction instead of after deployment. CSI is BAU.**
PRACTICE: LOCALIZED CHANGE

Traditional Change Advisory/Approval

- Puts approval furthest from knowledge
- Removes accountability from responsibility
- Increases batch sizes and lead times
- Increases risk around change failure

Centralized Change Advisory/Approval

Decentralized Change Model

“Changes are features with low variability.”

“How are you making the system safer for change?”
**PRACTICE: SUPPORT SWARMING**

**Standard 3 Tier Support Model**

- Level 1: Help Desk
- Level 2: Prod Ops
- Level 3: Dev

**Tiered Support Model Disadvantages**
- Creates queues which
- Elongate TTR
- Prevent the sharing of knowledge
- Issue resolution furthest from the knowledge

**Major Incident Swarm Model**

- Tech Lead
- Oncall
- Oncall
- Customer Lead

**Order Service**

**Storage Service**

**Swarm Model Advantages**
- All parties with proper expertise swarm
- Removes queues and handoffs
- Reduces TTR/maximizes knowledge sharing
- “Swarm and solve problems / create new knowledge”
The After Action Summary (AAS) process:

1. Impact Scoring
2. AAS Causal Factors & Countermeasures
3. AAS Work Item Follow-ups
4. Improvements Prioritized & Completed

AAS = AFTER ACTION SUMMARY (AKA: POSTMORTEM, POST INCIDENT REVIEW)
# PRACTICE: IMPACT SCORING AND POST INCIDENT REVIEWS

<table>
<thead>
<tr>
<th>Timeframe</th>
<th>Incident Remediation</th>
<th>Team Retrospective</th>
<th>AAS Preparation</th>
<th>AAS Review</th>
<th>Improvement Follow-ups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Who</strong></td>
<td><strong>Who</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triage Team and HD</td>
<td>Team owning incident</td>
<td>Team owning incident</td>
<td>All interested stakeholders across the org</td>
<td>Teams with causal factor follow-ups</td>
<td></td>
</tr>
<tr>
<td><strong>What</strong></td>
<td><strong>What</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whiteboard and recording of incident. AAS document created in Jira</td>
<td>Jira AAS document started and team records findings. Draft ready.</td>
<td>Additional summary, cross team information, follow-ups</td>
<td>AAS document complete and ready for review and final publish.</td>
<td>Defects or features created to improve the system</td>
<td></td>
</tr>
<tr>
<td><strong>Why</strong></td>
<td><strong>Why</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triage team swarms incident to drive quick MTTR and information sharing. Help Desk records pertinent facts and information in whiteboard.</td>
<td>Team retro occurs within 24hr so events are fresh and can be shared at the team level.</td>
<td>Additional details provided and questions resolved with stakeholders. Preparing for more global review.</td>
<td>All stakeholders: sales, leaders, other teams, PM review findings and follow-ups.</td>
<td>AAS generates work-items that are merged into the product backlog. This generates feedback and improvement for a safer system.</td>
<td></td>
</tr>
</tbody>
</table>

**AAS=** AFTER ACTION SUMMARY (AKA: POSTMORTEM, POST INCIDENT REVIEW)
Different teams pick different tools to track and manage work. Teams should pick tools that make them successful but “over-tooling” work management systems creates a cloud of confusion.
When collapsing to DevOps, teams are left with a wasteland of tools and swivel scenarios. Tool confusion fights against the ability to share knowledge, understand work capacity and improve workflows.
PROBLEM WITH MULTIPLE WORK MANAGEMENT SYSTEMS

- Tool Exhaustion
- Cognitive Overload
- Extra Status Reporting
- Missing/Incomplete Information
- Misaligned Understanding
- Lack of E2E Understanding

**Poor Work Visibility: Five Thieves**
- Unknown dependencies
- Too Much WIP
- Unplanned Work
- Conflicting Priorities
- Neglected Work
GUIDING PRINCIPLES TO IMPROVE WORK MANAGEMENT SYSTEMS

Make Work Visible

Adjust or Consolidate Work Management to Make Work Visible

Use Visibility to Improve System Behavior

System Behavior
• Flow of Work
• Handoffs
• Batch Size
• Work Process
• Nature of Work*
• Knowledge Sharing
• Common Awareness

*Note: Nature of Work refers to the nature of the work being performed and can impact the way work is managed and tracked.
## COUNTERMEASURES TO WORK MANAGEMENT SPRAWL

<table>
<thead>
<tr>
<th>Countermeasure</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off The Shelf Replication</td>
<td>Replicate entities across tools using off the shelf products such as TaskTop.</td>
</tr>
<tr>
<td>Consolidation, Replication, Strangulation</td>
<td>Choose a foundational system to replicate and consolidate work items to and strangle off non-strategic options.</td>
</tr>
<tr>
<td>Rip and Replace</td>
<td>Wholesale replace multiple work item systems and consolidate.</td>
</tr>
<tr>
<td>Swivel Chair</td>
<td>Manual entry from edge systems to central strategic systems.</td>
</tr>
</tbody>
</table>
Building Run Teams with Collapsed Toolchain: Real-World Example

- **DevOps Team**
- **Service**
  - EPIC
  - FEATURE
  - STORY
  - DEFECT
  - TC
  - CRQ
  - PIR
  - SR
  - INC
  - PR
- **Single work management tool:** Jira
- **L1 Support Helpdesk**
  - INC
  - PR
  - Help Desk
  - Remedy
- **Replicas**

*CSG CONFIDENTIAL AND PROPRIETARY INFORMATION | COPYRIGHT © 2018 CSG SYSTEMS INTERNATIONAL, INC. AND/OR ITS AFFILIATES ("CSG"). ALL RIGHTS RESERVED.*
FIND OUT MORE ABOUT INEFFICIENCIES IN WORK MANAGEMENT SYSTEMS

### SUMMARY: DEVOPS JOURNEY IN METRICS

<table>
<thead>
<tr>
<th></th>
<th>Begin</th>
<th>End</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel. Impact</td>
<td>507</td>
<td>85</td>
<td>-83%</td>
</tr>
<tr>
<td>Inc./mo.</td>
<td>1640</td>
<td>427</td>
<td>-74%</td>
</tr>
<tr>
<td>Impact Minutes</td>
<td>22,932</td>
<td>9,481*</td>
<td>-58%</td>
</tr>
<tr>
<td>Subscribers</td>
<td>48.9M</td>
<td>62M</td>
<td>27%</td>
</tr>
<tr>
<td>TPS</td>
<td>750</td>
<td>4,000</td>
<td>433%</td>
</tr>
<tr>
<td>Release On Demand</td>
<td>&lt;5%</td>
<td>28%</td>
<td>460%</td>
</tr>
</tbody>
</table>

**Agile: 2012-2016**

**DevOps: 2016-***

**Agile + DevOps 2012-2018**
SUMMARY: OUR JOB AND THE PROBLEM

Our Job As IT Leaders
Build Great Organizations That Engineer Great Products That Delight Our Customers and Employees

Problems
• Misaligned Incentives and Adversarial Relationships
• Queues Make IT Work Miserable
• Focus on Process vs Customer Value
• Operational Quality Continues to be a Problem
SUMMARY: OUR TOOLS TO ADDRESS THESE ISSUES

- Foundational
  - Lean and Systems Thinking
    - Strive for Flow, Continuous Learning and Adaptation
  - Courage & Transformational Leadership
  - Change Incrementally and Small Batches

- Principles
  - Build/Run Teams & Service Owner Value Streams
  - Ownership & Engineering Excellence: CI, CD, Everything is Code
  - Operations Is An Engineering Problem

- Practices
  - Single Backlog
  - Localized Change
  - Incident Swarming
  - Impact Scoring and Post Incident Reviews
  - Improve/Consolidate Work Management Systems