SAFe® and DevOps: What a Strange and Wonderful Trip It’s Been!

Scott Prugh, Chief Architect & SVP Software Engineering
CSG International
@ScottPrugh
CSG: Who Are We?

Industry Leading, Innovative, Configurable Business Support Solutions

REVENUE MANAGEMENT
CONTENT MONETIZATION
CUSTOMER COMMUNICATION MANAGEMENT
MANAGED SERVICES
ASCENDON NEXT GEN PLATFORM

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
State Of Your Industry??

Vision: Sustainable Pace; Customer and Employee Delight.
### Key Themes: Vision and Journey

<table>
<thead>
<tr>
<th>Mentors &amp; Support Community: Internal + External</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Learning / Evolution</td>
</tr>
<tr>
<td>Growth and experimentation mindset. Unlock the capability of your people.</td>
</tr>
<tr>
<td>Courage &amp; Transformational Leadership</td>
</tr>
<tr>
<td>“Success consists of going from failure to failure without loss of enthusiasm.”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Systems Thinking &amp; Lean Product Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queues</td>
</tr>
<tr>
<td>Work Visibility and Flow</td>
</tr>
<tr>
<td>Small Batches: Work &amp; Org Change</td>
</tr>
<tr>
<td>Structure Influences Behavior: Taylor/Conway</td>
</tr>
<tr>
<td>Customer Focus / Collaboration</td>
</tr>
<tr>
<td>Metrics &amp; Measurement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Engineering/Technical Excellence</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI/CD</td>
</tr>
<tr>
<td>Automated Testing</td>
</tr>
<tr>
<td>Infrastructure Automation</td>
</tr>
<tr>
<td>Architecture / Deployment Decoupling</td>
</tr>
</tbody>
</table>

### Timeline

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Prelude</td>
<td>Early Agile</td>
<td>Engineering Excellence</td>
<td>Architecture to Leadership to Agility</td>
<td>Early DevOps</td>
<td>Full DevOps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Prelude
- Early Agile
- Engineering Excellence
- Architecture to Leadership to Agility

### Full DevOps
- Early DevOps
- Full DevOps
Prelude
In Memory Of Mauricio “Mo” Zamora

Mauricio “Mo” Zamora

Mauricio helped start Scaled Agile, Inc., and was a principal developer of SAFe, but tragically, he passed away on Thanksgiving, 2011. His work lives on inside SAFe, where it improves the lives of practitioners every day; that was Mo’s personal mission. We think about Mauricio most every day, and his professionalism, knowledge, passion and integrity still set the standard we all try to adhere to.
1998-2006: Telution

• Mentor: Mauricio
• Foundation of Engineering Excellence & Agility
  • Customer Focus & Collaboration
  • Small/Frequent Releases: 3x/year → 6x/year
  • Hourly CI: Visual Build(Kinook)
• Version Control All The Things
  • Database from Version Control
• Developers are the DBAs
• Automated Regression: Embedded Test Engineering
  • 3x/year releases to 6x/year
Today:

Tomorrow:
2006: The Acquisition

Agile

Waterfall
2006: The Acquisition

Agile

Waterfall
Early Agile & Engineering Excellence
2007: Release Trains and Partially Agile

- Dean 1.0

Leffingwell’s Agile Release Train
2007: Release Trains and Partially Agile

• “Unified Agile Methodology”
• BiModal SDLC → No progress on enterprise agility

Program Management

- Project Definition
  - Request validated
  - BRD created
- Requirements
  - PDO writes requirements
- Analysis
  - High level design and estimate
- Implementation
  - Design completed
  - Coding occurs
- Testing
  - Testing and validation
- Operations
  - Code
  - Tested Code

Yay Agile! 15% of time here

430 days Lead Time
2007-2018: Bright Spot → CSG Engineering Excellence

- Continuous Integration Systems
  - Cruise Control then Jenkins
  - Types: Continuous, Daily, Stable, Patch
  - Versioned Database Scripting and Builds
- Core Frameworks
  - Middleware, Logging, Data
- Test Automation

Key differentiators in increasing quality and adopting true agility later. Engineering Excellence is more than just the functional code. Change behavior to change culture.
2011-2018: Technical Excellence & CI Build System Growth
From Architecture to Leadership to Agility
2009-10: Lean, TPS and Product Development Flow

**Thinking Tools**
- System/Lean Thinking
- Queuing Theory
- Org Tools
- Cross Functional Feature Teams
- Lean Primer

**Current Management View**
- Continuous Improvement
- Leverage Ingenuity
- Improvement Katas

**Economic View**
- Managing Queues
- Exploiting Variability
- Reducing Batch Size
- WIP Constraints
- Fast Feedback
- Decentralized Control

Books mentioned:
- Scaling Lean & Agile Development
- TOYOTA KATA
- The Principles of Product Development FLOW
- Agile Software Requirements

2009: Award for Excellence & Leadership (AFEL)

Can Good Code Fix A Bad System?

Can Good Code Fix A Bad System?

AFEL is a CSG Leadership Program
Recognition + opportunity to present to Executive Leadership

Key turning point in how we needed to influence the system and thinking
2009: AFEL Case Study Recommendations

Recommendations

3. Leverage Agility
   - Reduce Waste / Minimize Handoffs / Reduce Cycle Time
   - Favor Collaboration over Documentation
   - Increase Interaction Frequency
   - Increase Requirements Fidelity
   - Build a Team / Give Them a Chance to Succeed
     - *Esprit de Corps*
     - “Get Closer to the Customer”
     - “Watch the Baton, not the Runners!”
   - Deliver Incrementally
   - Minimize Risky and Large Changes

Looks like an Agile Team...
2009: AFEL Case Study Bonus Info

Lean Primer

TABLE OF CONTENTS

1. Introduction
2. Lean Thinking for Pros
3. Background
4. Lean Hierarchy of Needs
5. Lean Tools for Deliver Value Fast
6. Lean Production Lean Thinking Manager Numbers
7. What's Your Impact on People
8. Office/Facilities/Information
9. 11 Principles of
10. Lean Precipitate Development

Building High Performance Companies

Netflix Freedom and Responsibility Guide
http://www.slideshare.net/reed2001/culture-1798664
Leadership Series: Lean Thinking Mistakes

We think really hard about how to deliver features within our current structure. We don’t apply the same awareness and cunning to increasing the flow of value to our customers.

How do we rapidly deliver value at the lowest cost possible?

- **Sharpen our thinking tools:**
  - Systems Thinking
  - Mental Model Awareness
  - Lean Thinking

- **Thinking mistakes:**
  - Reliance on “Common Sense”
  - WIP
  - Resource pooling
  - Queuing

Final Review: 11/23/11
2011: Bringing Lean and Leadership Influence Together

Why Queues Are Bad
How Structure Influences Behavior

12 Days Reqs. → 430 Cal Days

Yay Agile!
2012: Inverse Taylor Maneuver & Great Reorg v1.0

Structure & responsibility will enforce behavior and prevents learning.

Which structure incents learning and removes queues?
Organize teams to optimize the entire flow of value.

Shook’s Model of Change: Influence Behavior to Influence Culture
2006: The Acquisition
2012: The Agile Reorg
Early DevOps
2013: The Start of DevOps

• Hitting the Operational Wall
  • Operations Frustrations and Failures
  • Forward Thinking Leaders and Shared Operations Teams

Steve Barr
“The Ops Guy”
2013: The Start of DevOps: Shared Operations Teams

Separate deployment and operations processes via infrequent handoffs create large batch transfers and high-failure rates.

- Handoff 4x/year
  - Agile DBT Team (7+/2)
  - Agile DBT Team (7+/2)
  - Agile DBT Team (7+/2)

Ops. Optimize

- Deploy 4x/year
  - “The Practice Team”
  - “The Game Team”

- Production Environment

Do you play the “Game” with a different team than you “Practice” with?

Make environments as close to production as possible and have the same team practice daily.

- Agile DBT Team (7+/2)
  - Agile DBT Team (7+/2)

- Dev/Test Environment
  - “The Game Team Practices”

- Production Environment
  - “Practice Make Perfect”

We let our teams practice at least 70 times before “Game Day”...

https://www.scaledagileframework.com/guidance-continuous-delivery/
2013: The Start of DevOps

• The Phoenix Project
  • Gene Kim
  • Parallels Between Parts Unlimited & CSG
IT Is Not Manufacturing...

But there are some good lessons to pay attention to!

How do you intake work and manage your portfolio?

“You probably don’t even see when work is committed to your organization. And if you can’t see it, you can’t manage it.”

-Dr. Erik Reid
2013: The Start of DevOps

- The Phoenix Project
  - Gene Kim
- Parallels between Parts Unlimited & CSG
- Queues again: WaitTime=%Busy/%Idle
- Introduction to the DevOps Community
- Invitation to DOES 2014
DOES 2014: DevOps and Lean in Legacy Environments

1. Accelerate Learning & Lean Thinking
2. Inverse Taylor Maneuver
3. Inverse Conway Maneuver
4. Shared Service Continuous Delivery
5. Environment Congruency & Practice
6. Application Telemetry
7. Visualize Your Work
8. Work Release & WIP Limits
9. Cadence & Synchronization
10. Reduce Batch Size
DOES 2014: Lean Systems Thinking and Cross Training

People First: Build a culture of Learning and Self Improvement by embracing Lean & Systems Thinking.

Find a Lean Framework

Encourage Cross Training

[Diagram showing Lean Framework and Cross Training models]
2015: Release Success

Operations Engineers watching over 15.1 deployment
71 features released to 50M subscribers
2015: The Importance of CI and Test Automation

Leverage test automation (ATDD) and CI with legacy code to reduce re-work and increase feature development time.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Planning</td>
<td>5.00%</td>
<td>5.00%</td>
</tr>
<tr>
<td>Environment Setup</td>
<td>10.00%</td>
<td>5.00%</td>
</tr>
<tr>
<td>Code Builds</td>
<td>10.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Production Support</td>
<td>25.00%</td>
<td>10.00%</td>
</tr>
<tr>
<td>Testing - Current Features</td>
<td>15.00% (Manual)</td>
<td>20.00% (Automated)</td>
</tr>
<tr>
<td>Testing - Regression</td>
<td>20.00%</td>
<td>5.00%</td>
</tr>
<tr>
<td>Feature Development</td>
<td>15.00%</td>
<td>55.00%</td>
</tr>
</tbody>
</table>

https://itrevolution.com/devops-resource-legacy-code/
2015: The Importance of Telemetry and Observability

Leverage telemetry as a learning platform to increase cross-team understanding of system behavior. Evolve your telemetry to accelerate feedback.
2015: Feeling Good; But What Those About Opposing Forces?

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

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

Development (SAFE/SDLC)
Change Management
Release Management
Production Operations
PMD

Operations (ITIL)
Customers:
High Quality Features Quickly!

2015: Operations Reality

Facts
1. 98% of incidents outside release
2. 92% of incidents fixed by ops
2016: Great Reorg v2.0

Before: Traditional Dev and Ops Structure

Observations/Hypotheses
- Release quality optimized but system quality not
- Manual processes around hard to run software
- Dev lacks understanding around operations
  - Lack of feedback into development
  - Produced hard to run software
- Collaboration was unnatural and scripted
  - Lack of “Esprit De Corps”
  - No shared mission
  - Misaligned org goals
- Context switching and work management chaos
- Handoffs caused elongated lead times
- Lack of ops engineering skills created a duct tape culture of fix “after the fact”
2016: Great Reorg v2.0

Before: Traditional Dev and Ops Structure

After: Cross Functional DevOps Structure

The future is here

IAC vs IAT
2012: The Agile Reorg

Agile

Waterfall
2016: The DevOps Reorg

DevOps isn’t just for Unicorns!

DevOps is Not a BiModal Choice!

Gene Kim visits CSG

SAFe does DevOps

Gene Kim visits CSG
### Ops is Really Hard
- Invisible Work
- Competing Priorities
- Heavy WIP
- Pager Fatigue

### Change is Voluminous and Scary

### Lack of Visibility

### Brent is Everywhere!

### Manual Configuration
- Unsafe Environments

### Lack of Ops Understanding From Dev
2016-2017: When Ops Swallowed Dev

<table>
<thead>
<tr>
<th>Ops is Really Hard</th>
<th>Countermeasure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invisible Work</td>
<td>Cross Functional Build/Run Teams</td>
</tr>
<tr>
<td>Competing Priorities</td>
<td>Make Work Visible via Shared/Single Backlog: FTR, SR, CRQ, INC</td>
</tr>
<tr>
<td>Heavy WIP</td>
<td>Collapse ITIL Functions into SDLC: Localize Change; Unwind CAB</td>
</tr>
<tr>
<td>Pager Fatigue</td>
<td>Incident Swarming [\rightarrow] Collapse L2, L3 support</td>
</tr>
<tr>
<td>Change is Voluminous and Scary</td>
<td>Automation: Infrastructure &amp; Audit Validation (20k) hours [\rightarrow] 4k</td>
</tr>
<tr>
<td></td>
<td>Self Service Operations</td>
</tr>
<tr>
<td></td>
<td>Automated Validation / Continuous Testing</td>
</tr>
<tr>
<td>Lack of Visibility</td>
<td>Pervasive Shared Telemetry</td>
</tr>
<tr>
<td></td>
<td>Everything in Version Control / Artifact Repository</td>
</tr>
<tr>
<td>Brent is Everywhere!</td>
<td>Cross Skilling; Version Control</td>
</tr>
<tr>
<td></td>
<td>Shared Runbooks; Shared Standups</td>
</tr>
<tr>
<td>Manual Configuration</td>
<td>Automation; Version Control</td>
</tr>
<tr>
<td>Unsafe Environments</td>
<td>Environment Congruency &amp; Practice</td>
</tr>
<tr>
<td>Lack of Ops Understanding From Dev</td>
<td>Shared standups; Shared On-call</td>
</tr>
<tr>
<td></td>
<td>Shared Metrics: Impact Scoring/MTTR</td>
</tr>
<tr>
<td></td>
<td>Post Incident Reviews: Local and Global Knowledge Sharing</td>
</tr>
</tbody>
</table>

Summary and Going Forward
### DevOps Journey In Metrics

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

Agile: 2012-2016
## 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>
</tbody>
</table>
## 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**

- **Subscribers:**
  - Begin: 48.9M
  - End: 62M
  - Change: 27%

- **TPS:**
  - Begin: 750
  - End: 4,000
  - Change: 433%

- **Release On Demand:**
  - Begin: <5%
  - End: 28%
  - Change: 460%
2017-2018: Moving Forward

**Spreading DevOps Across the Org.**
- Making All Work Visible
- Involving All Our People

- Work Life Balance → 15% in Team Driven Improvements
- Improve Release On Demand
- Education and Evangelism
- Extensive Infrastructure Automation
- Shifting Security Left
- Public Cloud
High Performance Behaviors That Lead to Better Outcomes*

**Technical/Software Practices**
- Deployment Automation
- Continuous Integration
- Version Control
- Continuous Testing
- Monitoring & Observability
- Database
- Security

**Software Delivery & Operational Performance**
- Software Delivery Performance
- Availability

**Continuous Delivery**

**Lean Product Dev/Mgt**
- Small Batches
- Make Flow of Work Visible
- Gather & Implement Customer Feedback
- Team Experimentation
- Production Feedback

**Org Performance**

**Transformational Leadership**
- Vision
- Intellectual Stimulation
- Inspirational Communication
- Supportive Leadership
- Personal Recognition

*Adapted from Accelerate & 2018 State of DevOps Report
Nicole Forsgren, Jez Humble, Gene Kim
Thanks: Community & Mentors

Mauricio Zamora
Ken Kennedy
Steve Barr
Erica Morrison

Dean Leffingwell
Gene Kim
Damon Edwards
John Willis

Dominica DeGrandis

Courtney Kissler
Ross Clanton
Jason Cox
Jeff Gallimore

Rosalind Radcliffe

Scott Nasello
Nicole Forsgren
IT Rev Staff