The Sprint 3 Revolt
How our transformation nearly went off the rails

Agile 2015
Jason Alexander
Raj Mudhar
Jason Alexander

Financial Services Product Guy
Lapsed swimmer, proponent of Slow Management, aversion to hugs

Raj Mudhar

Transformation Consulting Guy
Wannabe farmer and cook, proponent of leading by example
What we want to do

1. Share the story of a large scale transformation
2. Get to the point where it all (almost) went wrong
3. Discuss as a group alternative causes and solutions
4. Go through what actually happened
5. Answer questions to help others on a similar journey
A bit about the organization

A four year young leading edge division of a mature Global Financial Services Company operating in 90+ countries

1k+ Employees in the division

50 Times zones in which IT operates

5.5M Transactions each month

60M Customers served

67k+ Employees in the company

50 The number of Scrum teams
What was achieved is nothing less than awesome

- The completion of 9 major initiatives in 2014 versus only 2 in 2013
- The improvement of customer satisfaction ratings from under 20% to over 40% through product enhancements delivered
- A quality increase represented by the reduction of defects in live customer environments from 245 to 30 (an 88% reduction)
- An increase to 40 software releases in 2014 (up from 5 major and 2 minor releases in 2013)
- A normalized scrum team output per sprint increase of 120% from the start of the transformation through the end of 2014
- A scoring in the 70th percentile in requirements throughput and the 80th in predictability of output against an industry leading work management system client-base benchmark
Agile@Scale in Action – We chose to make some important investments

1000+ product and engineering division within a 10,000+ Financial Services IT organization

Enabled an end-to-end agile process
- Prioritization, Intake, Program, Product, Software and Release Processes are all agile
- Enable releases to become a non-event and occur every two weeks
- Rapid experimentation of testable product innovation and features

Refactored their platforms with componentization in mind
- Platform refactored into a discrete set of products and platform services
- Agile is enabling teams to focus on component stewardships and technical debt reduction to drive quality improvements

Re-aligned the organizational structure and delivery teams
- Product and technology is reorganized into ~50 cross-functional scrum teams
- Each scrum team has ownership of a platform service / application and is responsible for a set of platform KPI’s

Adopted new agile management and automation tools
- Agile ALM tooling for managing programs and product backlogs
- Continuous integration for automating continuous builds, testing and deployment
Investment: Leadership buy in

PRODUCT

TECHNOLOGY
Investment: Training and Coaching in place
Investment: Organization ready for change
Investment: Expectation that this will be messy
Investment: Teams embracing the mission
But...

THE BOSS...

Are we going to regain our development momentum? Seems to have slowed greatly....
We spent 5 months getting ready

Pilots
We explored the outliers, wanting to know where we could get quick wins and where the failure points were

SDLC Blueprint
We sketched a process from intake to release to help us understand how we were going to work – we assumed we were always wrong

Re-architecture
We undertook work to carve up the technology stack into individual, releasable “products”

Team/org design
We built teams around the desired architecture, not what was there, to take advantage of Conway’s Law

Training Teams
We trained 500+ employees using a customized curriculum that covered Agile as well as contextualized organizational content

Coaching
We deployed a team of coaches to support the Agile teams, all aligned to a coaching playbook to help maintain consistency, while allowing for local customization

Program Coaching
We coached Program leads and teams on how to build and manage their backlog of MMFS / MVPs, including how to work effectively with POs

CoE
We launched a Center of Excellence, staffed by dedicated employees to help drive change, CoPs, continuous improvement
We had a solid plan to further our maturity

- An incremental approach to change was used combined with a waved onboarding approach
- A high-level plan existed as a guide, details were elaborated just-in-time
- Each change was used to gather feedback and pivot/pursue decisions were made

Validated and refined as you go
January 14, 2014, we started sprinting

January 2014

<table>
<thead>
<tr>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>New Year's Day</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td></td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M L King Day</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td></td>
</tr>
</tbody>
</table>
Sprint 1 – we jumped right in

Teams assembled
- Team members were informed by their leaders to show up at the appointed morning
- Team members were expecting this

Introductions
- They met their coaches and POs
- An overview of the team, members, and some just in time training was delivered

Be cool
- Relax, we know this will be tough, but let’s get going and work the kinks out as we go
- Start work on the backlog, which was primarily populated with architectural decoupling work
Sprint 2 – It was getting tense

Risky Release

- Code quality was rickety and the release process was not fully in place
- The Release team and the Scrum teams were not on the same page

Sprint 3 around the corner

- Code in S3 > S2 – we were getting better
- We are building more code than we thought we could... Is this bad?

Executive expectations

- Our justification and business plan for agile required us to improve
- But we also needed to slow down and fix our release issues. The teams were fed up with the delivery problems and less then good quality code
Something was wrong, and the symptoms were...

- We were a “no go” for R2 deployment and barely fixed
- The leadership team was relaxed
- The queue of demand was growing
- Fond memories of waterfall
- Coaches hitting brick walls
- Lots of pissed off people
- “Everything is fine”
The town hall
The moment of truth
How would you solve this “problem”?

• Let’s do the 5 WHYs
• Breakout at your tables and come back with three solutions
Our root cause

No autonomy for the teams
Our root cause – no autonomy for the teams
What we did

- Moved ownership of code clean to the scrum teams
- Stop all new work to remediate
- Changed backlog focus to “True Grit”
- Bi-weekly global priorities focused on 3 sprints not the next one
- “If you have a conflict…”
- Management training

Note: None of the above were popular
Conditions for Success and Lessons Learned

- Executive buy-in and support is critical for making agile work @ scale
- Think organizational change instead of adopting agile in “projects”
- Align teams, process and architecture together
- The organization owns its “destiny” and is proactively problem solving to make it work
- Engage all levels of the organization
- Plan, but respond to feedback and pivot
- Take an incremental approach and learn along the way
- Avoid “installing” a method – contextualize the solution
- Be prepared to be bold– sometimes the right decision is the hardest to make.

Agile @ Scale transformation is chemistry and experimentation
There are common patterns that can help guide the transformation plan depending on where you are

- We are just starting... some projects are agile but we need to figure this whole thing out
  - Helping to solve the “burning platform”
  - Business and IT collaboration
  - Faster feedback and decision making cycles

- We are doing agile but it’s not working so well
  - Optimization opportunities
  - Expanding out the agile model
  - Agile engineering

- We are definitely agile but we want to take this to the next level
  - Innovation speed
  - Release speed
  - Self-directed teams

Typical Urgencies

- Target State Alignment
- Basic Team Agile Adoption
- Agile Leadership Capability

Change Streams

- Architecture and Team Re-alignment
- Agile Governance
- Continuous Integration
- E2E Agile Adoption
- Pull based System

- Componentization
- Experimentation
- DevOps
- Beyond Budgeting
- Agile Talent and Performance Management

However, none of these can just be reused but they can help your mental model for where to focus the transformation
If all else fails...
The responsive bagel beckons your questions
Jason Alexander
VP Product, Chase
Jason@slow-management.org
@jasonpalexander

Raj Mudhar
Leader, Deloitte LEAN
rmudhar@deloitte.ca
@rmudhar