Agile Journeys
The CareerBuilder Story
Journey Preview

The CB Culture

Overview our Continuous Integration & Delivery

Evolving our practices with more discipline

- 2007: Introduction to Scrum
- 2008-2012: Scrumification
- 2012: Defining Success
- 2012-2013: Verticalization of Teams
- 2013: Role & Process Clarity
- 2013-2014: Disciplined Grooming
- 2014: Team Performance Impact

Key Learning's & Challenges
Company Culture

**Candor**

**Agility**

**Passion**

*Easy to do business with*

**Disciplined Freedom**

- Always had a “**lean**” mentality
- Small changes – get **feedback**!
- Differentiator – “**The need for speed**” & 80/20 rule
In the Early Days – Late 90’s & Early 2000’s

Environment

- < 20 web servers
- 6 Application Servers
- 1 DB, ASP Front End
- VB backend
- Team Size < 15 initially (total in technology)

Life was good ....

- Folks could deploy in seconds! No compiled binaries.
- Self managed test environment, no QA
- Developers tested their own stuff
- On Demand Deployments!
- We had control … sort of 😊
Growing Pains

Getting harder to deploy often

- Increase in staffing and environment
- moved to .NET (ie. compiled binaries)

The Poppendieck Effect

- Deploy less? … Our leader said NO!
- Focus on daily automated deployments
- source control = good code
- ‘Flagging’ to control if code was on/off
Automated Deployment

Developed in-house tooling for automated deployment:

- Once a day
- Based on previous day code *snapshot (✔ compiled ❓ Tested)
- 10% Frontends and monitor for issues

Too many Hotfixes, Rollbacks & Aborts - A Testing strategy needed!
Automated Testing

Tested Code!

- Discover issues before they get into production
- Continuous integration (every hour to test)
- Deployments based on code quality (attempted daily)
- Teams began building out tests (Initially core tests only)

**NUnit** is an open source unit testing framework for Microsoft .NET

**Selenium** is a portable software testing framework for web applications. It automates browser interactions through a web driver.

**CruiseControl** is a Java-based framework for a continuous build process.
Evolution of Testing & Deployment

Challenges

- Test organization & management – lists of tests by area, then team
- Test runs took long – serial runs
- False failures (ie. contention w/tests)
- Investigation of failures – multiple changes associated

Learning’s & Improvements

- Selenium Grid – True Parallelization – increased feedback!
- Retry mechanism to reduce false failures
- Auto-detection of new tests
- Gated Check-Ins
- Last Good Snapshot (LGS) = “Blessed” snapshot to deploy
- 2-A-Day deployments – reduced time to completely deploy
- Stopping the line & lock downs (mass failure, team specific)
Environment Today

Environment
- ~500 web/application servers
- ~70 DB servers
- Team Size ~325 (total in Technology)

Core/native to our culture
- Pseudo DevOps Culture – developers own code thru production
- Iteration Control - flagging
- Rollback Strategy
- Real-time monitoring/alerts
- Implementing lean principles & evolving agile practices
EVOLVING OUR PRACTICES

Adding more Discipline
2007: Introduction to Scrum

How we worked
- Projects typically 2 weeks to 3 months
- Project prioritization meetings with competing customers
- Teams largely grouped based on related skillsets
- Teams, for the most part, set their own dates for projects

Challenges
- Setting customer expectations – customers suffered!

What we did
- *Agile Development Practices* conference – *Scrum can help!*
- Piloted 1st Scrum team in IT

95% accuracy on project completion date on 3 month projects
2008-2012: **Scrumification**

The Scrum Movement

- More teams moved to Scrum
- Team composition was lean (TL wore multiple hats – SM/PO)

Challenges & Concerns

- Victim to process – not understanding “The Why”
- Time and Focus towards
  - Dealing with Roadblocks
  - Driving product value

What we did

- Agile Coaching (me)
- Dedicated Scrum Masters & Product Owners
- **Key:** Finding that *first follower*
2012: Defining Success

How were we doing?

- Product Owner and/or Scrum Master for some teams
- Acknowledged teams worked differently
- Identifying appropriate measures
  - For teams – within the team
  - For leadership - across teams

Challenges & Concerns

- Effectiveness of agile roles?
- Working as a team

What we did

- Identified *Agile Guardrails* for teams
- Established core success metrics
IT Agile Guardrails

- Well defined cadence
- Transparency
- Commitment
Agile Metrics: Measuring Success

The Goal…

Customer Value & Quality =

How we measure success …

… Evaluating, and understanding our sprint based performance metrics in these four areas: customer value, quality, productivity, team/individual.

… Adjusting the constraints of cost, schedule, scope if needed to deliver the right value and quality at that time.
The Oz Principle

More than just guardrails… *We’re a Team!*

Living above the line…

*Individual & Joint Accountability*
The “Line”

**Above the Line®**

- Do It®
- Solve It®
- Own It®
- See It®

**Below the Line®**

- Cover Your Tail
- Finger Pointing
- Ignore/Deny
- It's Not My Job
- Confusion/Tell Me What To Do

The Line

© 2011 Partners In Leadership. All Rights Reserved.
Accountability

Don’t let it drop!
Building our Agile Environment

- Agile workspaces
- Education on core metrics
- Metrics on monitors – Transparency!

Challenges & Concerns

- Teams struggled with dependencies with other teams
- Focus for teams & consistency in terms of work

What we did

- Facilitate focus thru team alignment with the business/product
Creating Focus: Team Alignment

…to help provide a deeper focused team aligned with a particular area of the business providing complete technical ownership of its solutions…

…team consists of dedicated tech resources fixated on the prioritization of that business unit…

…Verticalization pools the necessary engineers together (regardless of skillset need) for the common business domain focus…
2013: Role & Process Clarity

Verticalization continues

- Continuing to revisit/identify core products/services

Challenges & Concerns

- Scrum didn't feel right for some teams?
- What does a Scrum Master do?
- Were our Product owners doctors? Or, waiters?
- How do all the roles on a team interact?
  - Team Lead, Manager, Scrum Master, Product Owner, Engineer

What we did

- Kanban for teams where it made more sense
- Clarified agile team roles, duties and responsibilities
- From Scrum Masters to Agile Coach
- Product Ownership community of practice
Role Clarity: Focusing on “The Why”

From Scrum Master to Agile Coach...

- Alignment with our Agile Guardrails
- Coaching to understand “The Why” & support “The How”
- Team Motivation/Drive = Autonomy + Mastery + Purpose
Autonomy

Agile coaches support the process that works for the team - *self direction is a natural inclination*. Agile coaches help teams to appreciate the depths of agile practices/principles.
Agile coaches help contribute to our inner drive by supporting teams to give them a *sense of progress*, not just *in our work*, but *our capabilities*.
Agile coaches help teams find purpose in their work to unlock the highest level of motivation.
A Study on “Purpose”

Adam Grant – Professor at Wharton School of Business

- Evaluating “Purpose” as an effective motivator
- Call Center fund raising organization

Test:

*infusing a task with purpose can motivate high performance*
“Purpose” an effective motivator?

3 Groups - Call center fundraising organization

- **Group 1 – Personal Benefit Condition**
  Read stories from employees describing perceived benefits of the job

- **Group 2 – Task Significance Condition**
  Read stories from beneficiaries of the fundraising (scholarships obtained & positive impact)

- **Group 3 – Control Condition**
  Did not read any stories

*Employees did not share what was read*
The Results...

Personal Benefit & Control Groups

Raised same number of pledges and money as before

Task Significance Group

Earned more than twice ...

the number of weekly pledges & donation money
Product Ownership community

More alignment across POs
- Consistency with Process (visualize cadence, Discovery)
- Truly owning the product (not just an order taker, ability to put product first)

Career Growth
- Mentorship/coaching
- Focused Goals
- Ensure decisions made for what's best for the company and not for whom they report to

Visibility across the whole
- Ensure no duplication of effort and identify synergies
- Right initiatives with right team
2013-2014: Disciplined Grooming

“Sprinting Ahead”
- Sprints started to look more like “Spikes”
- Sprint planning/estimation was painful

Challenges & Concerns
- **Commitment**
  - Not having “mature” stories
  - Unsure of what they were committing too
- **House Keeping** – Addressing Tech Debt

What we did
- Training with **Marty Cagan**
- Separate *Discovery* from *Delivery*
- Dedicated time towards Tech Debt
Creating Focus: Disciplined Grooming

2 Pipelines (Discovery & Delivery)
- Disciplined, concurrent grooming sessions for each pipeline
- Regular cadence increasing story maturity

Discovery
- Defining the “what” – identifying mature stories ready for Delivery track
- Forward Thinking Activities
  - Grooming
  - Technical Discovery - Experimenting/prototyping to understand what is feasible
  - Story Time - understand need & value; confirm story readiness for delivery

Delivery
- Team focused on delivering “mature” stories that meet business need
Creating Focus: Tech Debt Backlog

**Discipline with paying back the debt we’ve built up**

- Allocating 20% of delivery towards pay back
- Removes the “Us vs. Them” avoiding contention between PO and team
2014: Team Performance Impact

What?
- Team Impact on quarterly bonus

Why?
- Team based component fundamental in having properly aligned incentives and driving teamwork

2014 Quarterly bonus structure

<table>
<thead>
<tr>
<th>Moving from...</th>
<th>Moving to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>50% Company</td>
<td>50% Company</td>
</tr>
<tr>
<td>50% Individual</td>
<td>25% Individual</td>
</tr>
<tr>
<td></td>
<td>25% Team*</td>
</tr>
</tbody>
</table>

* Team portion currently equates to commitment

✔ Customer value/satisfaction
✔ KPIs focused on outcomes
Utopia – Just as we imagined

Challenges with our agile implementation

- Specialization
- Siloed approach
- “Everybody in” attitude
- Knowing and Engaging our Customers

Key Learnings

- Measure outcomes, not output
- Need for more T-Shaped People
- Still refining/defining clear products on the IT side
- Team alignment/structure is important
- One size does NOT fit all (Process & People)
Journey Review

✓ The CB Culture

✓ Overview our Continuous Integration & Delivery

✓ Evolving our practices with more discipline
  ▶ 2007: Introduction to Scrum
  ▶ 2008-2012: Scrumification
  ▶ 2012: Defining Success
  ▶ 2012-2013: Verticalization of Teams
  ▶ 2013: Role & Process Clarity
  ▶ 2013-2014: Disciplined Grooming
  ▶ 2014: Team Performance Impact

✓ Key Learning’s & Challenges
Thank You!

Manager of Agile Development
Andy.krupit@careerbuilder.com

www.linkedin.com/in/akrupit
Agile Metrics: Comparing Teams

Team Health - Information Systems

<table>
<thead>
<tr>
<th>Team Ranking</th>
<th>% Commitment Met</th>
<th>% Delivered Over Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 3

- % Backlog Fresh: 57
- % Backlog Estimated: 67
- Days Since Last Fun: 26