Six Steps Towards Self-Learning
Teams and Organizations
Agile2019
Washington, D.C.

Andy Cleff, CSP, ICP-ACC, ICP-ATF, CSM, CSPO
A story about a photo I love....
We learn rapidly by experimenting frequently. We make our experiments “safe to fail” so we are not afraid to conduct more experiments.
The three hardest words to say

In too many organizations too many people are doing a second job no one is paying them for...
Conditions for growth

Personal evolution, human flourishing, becoming a better person — these are at the center of the culture of a deliberately developmental organization (DDO)

agileuprising.libsyn.com
“All life is an experiment. The more experiments you make the better.”

- Ralph Waldo Emerson
What your role in all of this?
Overview: Six Steps

1. Build an Inventory
2. Construct a Team Level Matrix
3. Populate a Learning Backlog
4. Get Your Guilds Going
5. Measure Stuff
6. Make Everything Visible
Step One: Build an Inventory
Skill inventory

Prompt:

• What do we already know?
• What do we want to learn?
• What skills/competencies do you think we need to be most successful, now and in the foreseeable future…
Skill inventory

Example Categories:

- Languages & Technologies
- Processes & Practices
- Soft Skills
<table>
<thead>
<tr>
<th>List of Tools / Languages / Technologies / Platform Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android AutoValue</td>
</tr>
<tr>
<td>Android &amp; Claims</td>
</tr>
<tr>
<td>Android In-app billing</td>
</tr>
<tr>
<td>Android CI/CD - Circle CI</td>
</tr>
<tr>
<td>Android Proguard</td>
</tr>
<tr>
<td>Android Lint</td>
</tr>
<tr>
<td>Android Chromecast Receiver</td>
</tr>
<tr>
<td>Android Chromecast Sender</td>
</tr>
<tr>
<td>Android multi-windows</td>
</tr>
<tr>
<td>Android Facebook Integration</td>
</tr>
<tr>
<td>Android Banner Ads - DFP</td>
</tr>
<tr>
<td>Android Video Ads - Freewheel</td>
</tr>
<tr>
<td>Android Widevine - Streaming &amp; Offline Viewing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List of Processes / Practices / Platform Agnostic</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVB testing</td>
</tr>
<tr>
<td>long-term strategy</td>
</tr>
<tr>
<td>understand growth KPI's</td>
</tr>
<tr>
<td>accounting</td>
</tr>
<tr>
<td>code review</td>
</tr>
<tr>
<td>debugging</td>
</tr>
<tr>
<td>learning to learn</td>
</tr>
<tr>
<td>abstraction</td>
</tr>
<tr>
<td>identity and access management</td>
</tr>
<tr>
<td>cqrs</td>
</tr>
<tr>
<td>refactoring</td>
</tr>
<tr>
<td>continuous integration</td>
</tr>
<tr>
<td>continuous deployment</td>
</tr>
<tr>
<td>code profiling</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List of Soft Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>self-management</td>
</tr>
<tr>
<td>speaking</td>
</tr>
<tr>
<td>adaptability</td>
</tr>
<tr>
<td>researching</td>
</tr>
<tr>
<td>communication</td>
</tr>
<tr>
<td>networking (people)</td>
</tr>
<tr>
<td>interviewing</td>
</tr>
<tr>
<td>facilitation</td>
</tr>
<tr>
<td>coaching</td>
</tr>
<tr>
<td>improv</td>
</tr>
<tr>
<td>giving and receiving feedback</td>
</tr>
<tr>
<td>writing for publication</td>
</tr>
<tr>
<td>conflict resolution</td>
</tr>
<tr>
<td>collaboration</td>
</tr>
</tbody>
</table>
Discussion / Q & A

1. Build an Inventory
2. Construct a Team Level Matrix
3. Populate a Learning Backlog
4. Get Your Guilds Going
5. Measure Stuff
6. Make Everything Visible

Is anything confusing so far?
What are you doing already that's working?
What's not working for you?
What buckets are you thinking might work for initial brainstorming sessions?
**Step Two:**
**Build a Team Level Matrix**

<table>
<thead>
<tr>
<th>Technical/Platform Specific</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>.net</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ajax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alamofire</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>android (Java)</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>angular.js</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autolayout (iOS)</td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS</td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backbone.js</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>Brightscript</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>c#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>C++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>cassandra</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>charles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>chef</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comscore</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>consul</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cucumber</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: *"N" = now, "F" = future*
Team matrix of competencies

Simple Model of Competency

- Apprentice
- Journeyperson
- Master
Team matrix of competencies

Dreyfus Model of Skills Development

• Novice
• Advanced beginner
• Competent
• Proficient
• Expert
Team matrix of competencies
<table>
<thead>
<tr>
<th>Technical/Platform Specific</th>
<th>Apprentice (Um... dabbled there once or twice)</th>
<th>Journeyperson (Yeah, I can do it.)</th>
<th>Semi-Master (Heck, I can teach it)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Taher</td>
<td>RyanC</td>
<td>Dan</td>
</tr>
<tr>
<td>.net</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ajax</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alamofire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>android (Java)</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>angular.js</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autolayout (iOS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backbone.js</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brightscript</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cassandra</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>charles</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>chef</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comscore</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>consul</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cucumber</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: "N" = now, "F" = future, leave blank if does not apply.
Dealing with fear
Congruence

• People feel they can speak up about problems, issues, and mistakes
• Mistakes aren’t held against people; they are viewed as a potential source of learning
• People can talk about disagreements and differences of opinion
• People are open to new ways of getting work done, and new ideas are valued

“…an alignment between a person’s interior world of thoughts and feelings and the exterior work of their social and physical reality. Balancing the needs and capabilities of self, others, and context.”

Esther Derby
7 Rules for Positive, Productive Change
Discussion / Q & A

1. Build an Inventory
2. Construct a Team Level Matrix
3. Populate a Learning Backlog
4. Get Your Guilds Going
5. Measure Stuff
6. Make Everything Visible
Step Three: Learning Backlog + Slack & Flow
Taking time to sharpen the axe...
Discussion / Q & A

1. Build an Inventory
2. Construct a Team Level Matrix
3. Populate a Learning Backlog
4. Get Your Guilds Going
5. Measure Stuff
6. Make Everything Visible
Step Four: Get Your Guilds Going
Share knowledge, skills, and capabilities

- Special Interest Groups (SIGs)
- Centers of Excellence (CoE)
- Communities of Practice (CoP)
- Birds of a Feather (BoF)
- Tech Clubs
- Guilds
- Tribes
- ! = Committee
Communities of practice

- A community of people
- A domain of knowledge
- A shared practice
Community design possibilities

- Bootlegged
- Legitimized
- Supported
- Institutionalized
“When people work this way, barriers and boundaries between people and what they do are often insubstantial or irrelevant, since a collective endeavor holds people together.”

Seely Brown, *The Interaction of Complexity and Management*
Outcomes of CoP’s

• Sharing of Information, insights, problem solving

• Pondering of common issues, exploration of ideas, acting as sounding boards

• Creation of tools, standards, design patterns

• Hosting knowledge-based activities, like lessons learned, Q&A sessions, and on-boarding newcomers
Discussion / Q & A

1. Build an Inventory
2. Construct a Team Level Matrix
3. Populate a Learning Backlog
4. Get Your Guilds Going
5. Measure Stuff
6. Make Everything Visible
Step Five: Measure
“…Analysis without numbers is only an opinion.”

Atkins Law #1

“…If all we have are opinions, let’s go with mine.”

Jim Barksdale
Measure many things

Process Health
Release
Product Development
Technical / Code
People / Teams
Process health metrics

Assess day-to-day delivery team activities and evaluate process changes:

- Cycle time
- Cumulative flow diagrams
- Control charts
- Number of experiments performed
- Number of improvements you've made to your process over time.
- Flow efficiency
- Story/epic lead time
- Successful iteration completion
- Escaped defect resolution time
- Time thievery
The five thieves of time

- Too much WIP
- Unknown dependencies
- Unplanned work
- Conflicted priorities
- Neglected work
Release metrics

Focus on identifying impediments to continuous delivery:

- Time since last release
- Release success rate
- Release time
- Cost per release
- Escaped defects
- Release net promoter score
- Release adoption / install rate
Product development metrics

Help measure alignment of product features to user needs:

- Customer value delivered
- Risk burn down
- Value stream mapping
- Sales velocity
- Product net promoter score (NPS)
- User analytics / DAU / MAU
- Backlog health index

- Number of validated business-level hypotheses developed
- Number of times a week you talk to an actual customer
- Ratio of implemented to non-implemented customer-driven changes
Technical & code metrics

Help determine quality of implementation and architecture:

- Automated test coverage
- Number of tests written before coding
- Unit / regression test coverage
- Build time
- Defect density
- Code churn
- Code ownership

- Code complexity
- Coding standards adherence
- Crash rate
- Build breaks
- Technical drag
- Ratio of fixing work vs feature work
People & teams’ health and wellbeing

A strong constitution is necessary for teams to overcome adversity, maintain resilience and nurture a culture rich in collaboration and innovation.
“We’re trying to create systems which learn, and ‘happy’ systems don’t learn, they’re complacent.”

Prof. Dave Snowden
“Resiliency practices help you keep going, find meaning in the often long and political process of creating change, and see the good in [organizations] – even on days that can feel like you’re lost in a bureaucratic hairball.”

Lois Kelly

Resilience

Dictionary: *The capacity to adapt to stress or loss healthfully*

- Positivity
- Focus
- Flexibility
- Organization
- Proactivity
Visualizing health, wellbeing and resilience

Gallup Q12
Mood App
TeamMood
eNPS
Comparative Agility Survey
Crisp Happiness Index
Team Barometer

Spotify Health Check
Lean Agile Intelligence
Team Morale Metric
H.I.P. Survey
OfficeVibe
15Five
Happiness Wall
Happiness, Innovation, and Productivity Survey (Nomad8)

- Purpose
- Focus
- Autonomy
- Relationships
- Mastery
- Creativity
Human beings are smart and able to game any system. To prevent gaming, be transparent about values, intentions, and the metrics everyone is using.
What this is not

- A competition
- A comparison
- A weapon
What this is not

- An incentivization model
What this is not

- The absolute truth
Measure early and often

• That which is measured will improve, at a cost. (Hawthorne)

• When a measure becomes a target, it ceases to be a good measure. (Goodhart)

• Correlation is not causation, but it sure is a hint. (Friedman)

• Use multiple viewpoints - technical as well as human - to get a holistic perspective
Measure many things

- **Time spent mobbing** + cycle times + escaped defects
- **Unit test coverage** + release metrics + technical drag
- **Thieves of time** + H.I.P + Flow efficiency
Try something new

• The environment changes all the time
Discussion / Q & A

1. Build an Inventory
2. Construct a Team Level Matrix
3. Populate a Learning Backlog
4. Get Your Guilds Going
5. Measure Stuff
6. Make Everything Visible
Body Thinking
Step Six: Make Everything Visible
Working out loud

...an approach to building relationships that makes you more effective, gives you access to more possibilities...
Who this framework is for

There are two groups of stakeholders:

- The individuals & teams
- The people & systems supporting the teams
From the team perspective

- Builds up awareness
- Reveals working and what’s not
- Expands perspectives
- Provides a balanced view
From the supporting role perspective

- Provides high level summary
- Reveals patterns across teams
- Shows opportunities to change the system
Visibility and transparency

• Build momentum
• Drive interest
• Remove constraints
• Enable collaboration across boundaries
• Share Knowledge
• Lean coffee
• Celebrate experimentation
Recognize and celebrate

You are here

The Stretch Zone
Where all change, growth & transformation happens

The Panic Zone
Recognize and celebrate

“Failure is the key to success. Each mistake teaches us something.”

Morihei Ueshiba
Recognize and celebrate
Recognize and celebrate

Geese
“Simple, public recognition is one of the most effective and most underutilized management tools.”

Laszlo Bock
From the top & then improv

• Build an Inventory
• Construct a Team Level Matrix
• Populate a Learning Backlog
• Get Your Guilds Going
• Measure Stuff
• Make Everything Visible
Ponder the following

• When you’re learning at your best, you’re like what?

• Time is like what?

• When something’s gone wrong and you’ve got over it, what’s that like?
Principle #12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Where might you go?

• Creativity and learning takes a mix of disciplines

• Free flowing collision of ideas through conversations and connections makes a difference

• Exchange of ideas and technologies speeds innovation
Let’s keep the conversation going…

Andy Cleff

andycleff@icloud.com
andycleff.com
linkedin.com/in/andycleff
@JustSitThere
agileuprising.com