Agile Team Training with Ethan Foulkes

4:30pm - Tuesday 24th, 2017 NYCDA New York City, Prospect

Ethan Foulkes heads system development at Camerican (not affiliated with yours truly, github.com/camerican). Today we'll be talking about how agile processes work and what responsibilities of team members typically are within such teams.

The Responsive Team

Communication is Key

Krystal Kaplan (KK) - Ethan is an amazing partner in crime when it comes to working with our students, conducting workshops, responsive teams. His company has hired a couple graduates, most recently from WDI April and previous cohort from September 2015. Today what we're going to be doing is something a little bit new. At NYCDA we're at the process of revamping our curriculum, we're not only educators abut also learners as well. What we're planning on doing today is learning more about responsive teams and doing more interactive session. And I'll take it away and have Ethan do his thing.

Ethan Foulkes (EF) - Thank you everybody. This is the first time I've done this. Normally I've come in at the end. Give a little bit of info at the end. This time we're going to come in at the beginning, dive into something a little bit interactive. It'll be a little bit rough and that's okay, we'll iterate and it will get better. Fun to experiment. Me, I'm the director of IT/ Supply Chain for Gellert Global Group. 7 or 8 companies. Supply Chain for one of those companies.

Fun fact, prior to that my life was for about 8 years a professional poker player. Job transition for me. I did IT way back in the day and then I came over to the business side of things. The one thing I learned from being a professional poker player, don't get me wrong, I was pretty good. Had friends that were much better than me. I was usually more successful than they were. The reasons you were more successful, you had better bankroll management and better emotional control. Those are the hidden things. Everybody thanks whoever has the best read, can play the best strategic poker.

What makes the best programmers are communication. It's great if you can code but it you can't communicate with the stakeholders, it's not going to matter.

This is my day. A lot of people get overwhelmed when they look at my calendar. For me this is super productive, I get all my work done in meetings.

You have to structure a meeting right, talk about the right things and needs to move along in the proper fashion. Otherwise meetings are a waste. Recently, I found this interesting to included it in the presentation.

What Google Learned Form Its Quest to Build the Perfect Team

Project Aristotle

Charles Duhigg Feb 25,2016

Project Aristotle, NY Times

They wanted to know what makes an awesome team and what makes not an awesome team.

They were expecting it to be team composition. All Type As or a mix. All business or friends outside of the workplace. It's not what they found. That they found it was the team dynamics. The way they interacted and how they structured. A few of the key takeaways.

Norms ar the traditions, behavioral standards, and unwritten rules that govern how we function when we gather.

The right norms raise a group's collective intelligence.

The wrong norms hobble a team, even if, individually, all the members were exceptionally bright.

Good teams spoke in roughly the same proportion, a phenomenon the researchers referred to as "equality in distribution of conversational turn-taking."

Sometimes it could be a group of lots of interruptions. When they measured at the end of the day how much time everyone spent talking, there was an equality there.

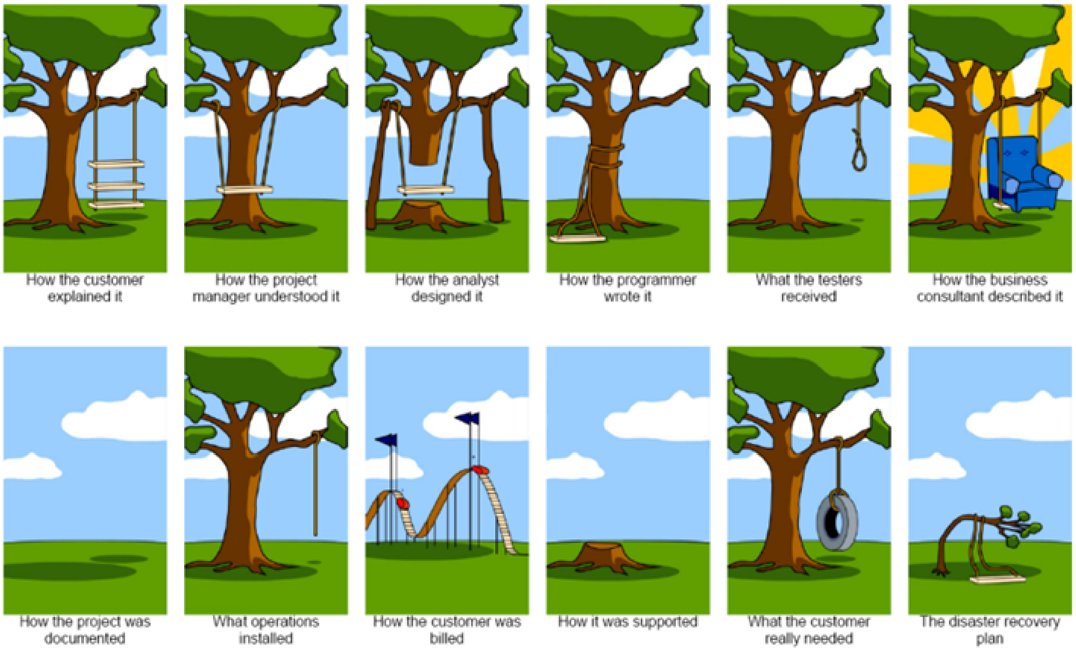
Good teams all had high "average social sensitivity" -- skilled at intuiting how others felt based on their tone of voice, their expressions and other nonverbal cues.

They seemed to know when someone was feeling upset or left out.

How do we manufacture these things?

KK - I would just like to make a connection to our last workship with teamwork. A, B, C, D profiles. Everyone was in a group. Makin the connection for you.

EF - Life of project management and development.


How the customer explained it
How the project manager understood it
How the analyst designed it
How the programmer wrote it
What the testers received
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed
The disaster recovery plan

Good questions and bad questions

Volunteered from our class:

- What are some of your interests?
- What are you expectations?
- What are you thinking about this or that?
- What's your favorite way to vacation?
- Are you looking to relax or explore?
- Where do you want to go?
- What's your budget?

Bad questions:

- It's up to you.

From the slides:

Good questions

- What activities do you like?
- What time do you want to get up?
- How much are you comfortable spending?
- What's the best vacation you ever took?
- What's the worst vacation you ever took?
- Do you like to travel?
- Where should we stay?
- Should we ring the kids?

Bad

How many vacation days do you have? When should we go? Who's going to book the tickets? Who will pay for it?

Waterfall vs Agile

Waterfall is what we used to do Agile is what we do now, but we're not all there yet. Each person in agile will be on different areas of that spectrum.

Waterfall is making a plan and then saying there's the golden plan, now go forth and do it. When you're done, give me the finished results. Waterfall has its roots in manufacturing. You have a lot of fixed costs, not going to change things after it's done. That's the roots.

When we started software development, that was the methodology we had. This is the end result of waterfall, we're supposed to be here, but we're here. Any bandaid we need.

What is agile. This is parkour or free running. You can lose half a day on YouTube watching these guys . Controlled figuring it out as you go.

Agile Manifesto.

"Waterfall stuff sucks, let's make a manifesto because manifestos are badass"

Take aways

- Individuals and interactions over processes and tools
- Working software over Compressive Documentation
- Customer collaboration over Contract Negotiation
- Responding to change over following a plan

Agile Development

"Software is eating the world."

- Flatter organizations
- self-organizing, responsive teams
- short feedback loops
- inspect, adapt, and iterate
- continuous improvement
- high visibility + transparent communication

Responsive Teams

- Dynamic, ad hoc formation of teams and roles
- Easy and fast access to the right data and information
- Having the right tools
  - Communication
  - Project Management (visibility)
  - Project Documentation
- Fewer meetings
- Predictable delivery

Atlassian Tools

- HipChat (communication)
- JIRA (issues tracking)
- Confluence (documentation)

Scrum vs. Kanban

Scrum

- Sprint Planning
- Daily Standup
- Sprint Review
- Retrospective

Kanban

- Lean Manufacturing
- Constant Flow
- No Backlog

Sprint Planning

- Planning
  - Start of the sprint
  - Discussing tradeoffs
  - Time and energy is finite
  - Spring "Commitment" (Story Points)
  - Backlog Grooming
- Predicable delivery
- Mutual agreement of work
- Visible to everyone

Daily Standup

The Daily Standup

- What did you work on yesterday?
- What are you going to work on today?

- Is there anything blocking or slowing you down?

- Identify blockers

- 15 minutes only
- Creates accountability

Spring Review & Retrospective

- Review
    - Invite Stakeholders
    - Review Acceptance Criteria
    - Demo
- Retrospective
    - What worked well?
    - What didn't work well?
    - How can we improve?
    - Happiness (Happy/Unhappy)
- End of sprint

- Surfaces tensions and problems early
- Inspect, adapt, iterate
- Collaboration and experimentation

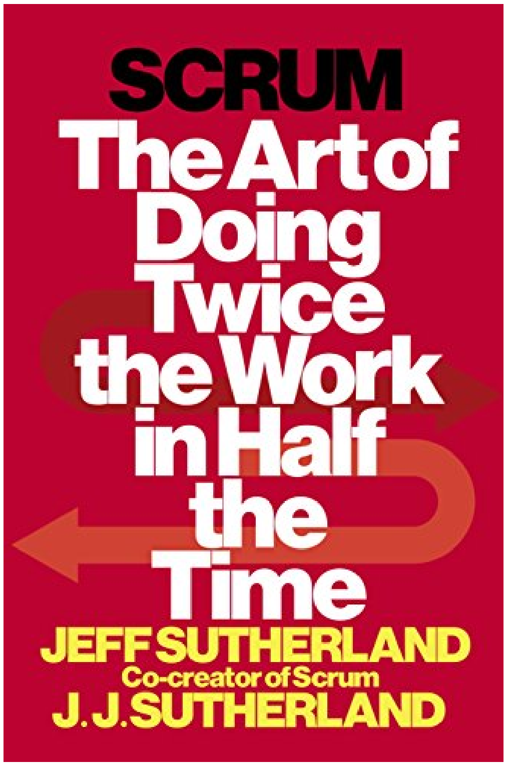
How will we know we're done? User will make a choice through a dropdown or through a multiselect. It will have green yellow and red colors for status. I will have my definition of done. Does it meet all these criteria That's our sprint review. We'd then kick out the stakeholders and do our internal review. Talk about what went well, what didn't go so well.

This is in Confluence and we'll make a page or each retrospective. Add everyone's name on the left, give a happiness score. Don't want my answer to influence you. Going to hold up fingers.

Burndown Chart

Track progress, completing items that are weighted by points. As we continue, progress should be a constant. Sudden movements in progress is a sign that blockers are not being identified.

Correlation between point reporting and happiness.



Scrum the art of doing twice the work in half the time by Jeff Sutherland

Collaborations for Resolving Blockers

Robert & Joel

Hak & James

Jarvis & Robert

Eric & Alfe