Information Technology & Data Analytics

Raptor Extinction Leads to the Rise of the Empires

Presented by Matt House, Carl Rudroff, & Tonya Munger

Where We Started…

- One **large** software project comprised of several small development teams

- Teams lacked **cohesion** and **identity**

- Program management office **controlled** processes and schedules

- **Overwhelming** and **ever-changing** integrated master schedule

You Bred Raptors?

- Frustrations led one of the development teams to break from the existing processes and implement an agile approach to software development

- Took one team through an agile boot camp outside of Boeing

- Stood-down 3 weeks for a "sprint 0"
  - Established a team operating rhythm
  - Established norms for the team
    - What is an acceptable requirement?
    - Definition of done?
  - Engaged internal customers to promote two-way communication
- Empowered the team by giving them ownership of the process
- Team performance significantly improved

The "Velocity Raptors" were Born

Don't Do Agile…Be Agile

- Cannot just be a process change, must be a culture change too

- Teams must have autonomy and be empowered with ownership of their processes

- Moved from a system of individual assignments to one of team-level commitments

  - No team member was assigned work. Team members chose work from the WIP board themselves.

  - Team members started picking user stories to work on together. This led to increased quality and cross-training.

- Most importantly, the team bonded to such a degree that they owned any success or failure as a team.
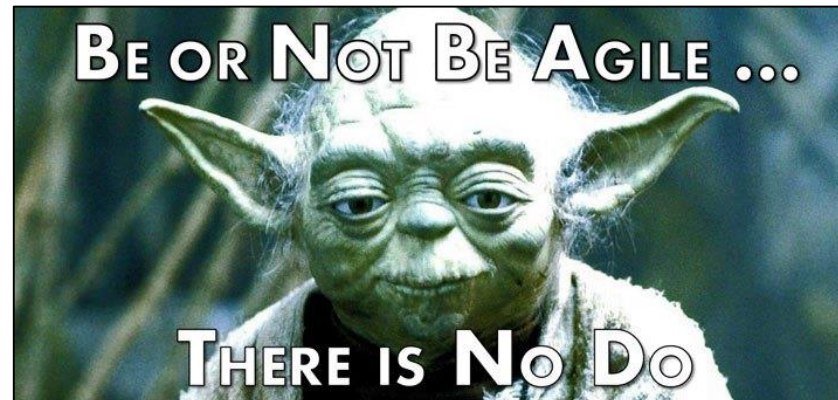
What You're Doing Is Great…Now Please Don't Do It Anymore

- Program management didn't like the Raptors' autonomy

  - Frustrated with lack of control

  - Wanted standardized development process with a "one size fits all" formula

- Program management constraints led to other teams lacking ownership of the process

- Program management had created a culture that didn't inspire improvement, ownership, or accountability

A New Opportunity

- In contrast to program management, executive leadership took note of the Raptor's success and realized the program needed to change

- Leadership asked the Velocity Raptors to find a way to propagate that success throughout the program

- Goal was to lead the charge in re-organizing the project to implement a scaled-agile approach to manage the program

- The challenge was to do this in a 5 week window that would still accommodate existing program commitments

A Push for Change

- All team members went through the same boot camp, from SIUE, as the Raptors

- Teams were re-organized around software capabilities

- Velocity Raptors were dispersed across the project to help propagate processes and culture

- Subject matter experts were assigned to each of the teams in a product owner role

- Empowerment was key – teams were given control of sprint plans, team norms, and team name

- Adopted the concept of a "release train" comprised of small teams that operate together towards a common release

The Rise of the Empires…

Backlog Prep and Sprint 0

- All existing requirements were decomposed and prioritized
- For most team members, this was their first experience breaking requirements into user stories
- Scrum masters worked with program leadership to determine where processes would fall, "federal" or "state"
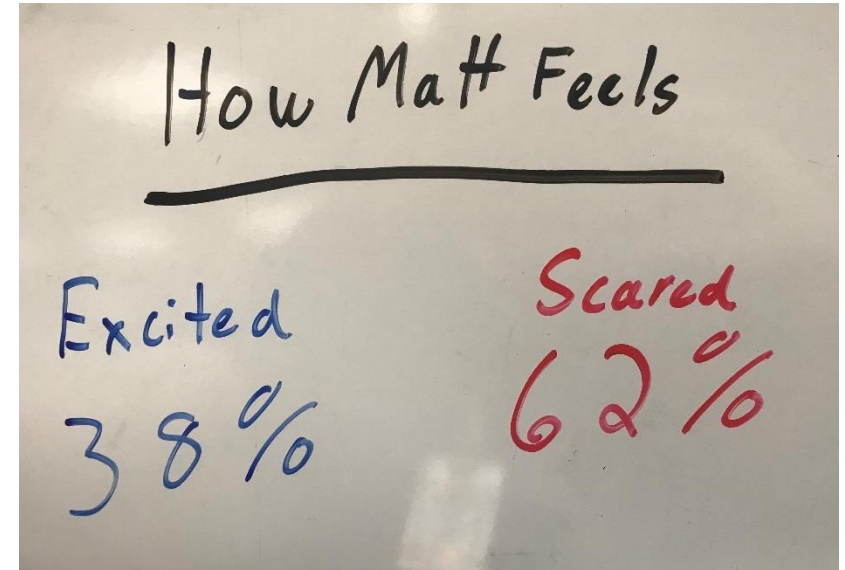
The Program Increment

- Chose a central theme all teams could unite under: Historical Empires
- All teams participated in a program increment planning event, which involved over 100 software developers, scrum masters, analysts, and product owners
- Planned each team's work into a series of seven sprints that would begin and end at the same time with a coordinated release
  - Due to technical constraints, we could not perform a production release after each sprint
- Identified dependencies that existed between teams
- Actively managed risks, issues, and opportunities as a team

The Empires of IPDM

Program Increment Planning Event

Increment 1 Begins

- All the teams operated with the same sprint cadence of seven, two-week sprints

- After the 14 weeks of development, we entered a 4 week testing and hardening period

- Current deployment process was quite cumbersome, so we could only get deployments to our development environment once per week

- Tracking and Metrics

  - Due to a lack of time with the backlog, we could not put story point estimates on all the work before the start

  - Because of this, we tracked a burn down of total stories

IPDM User Story Dev Complete
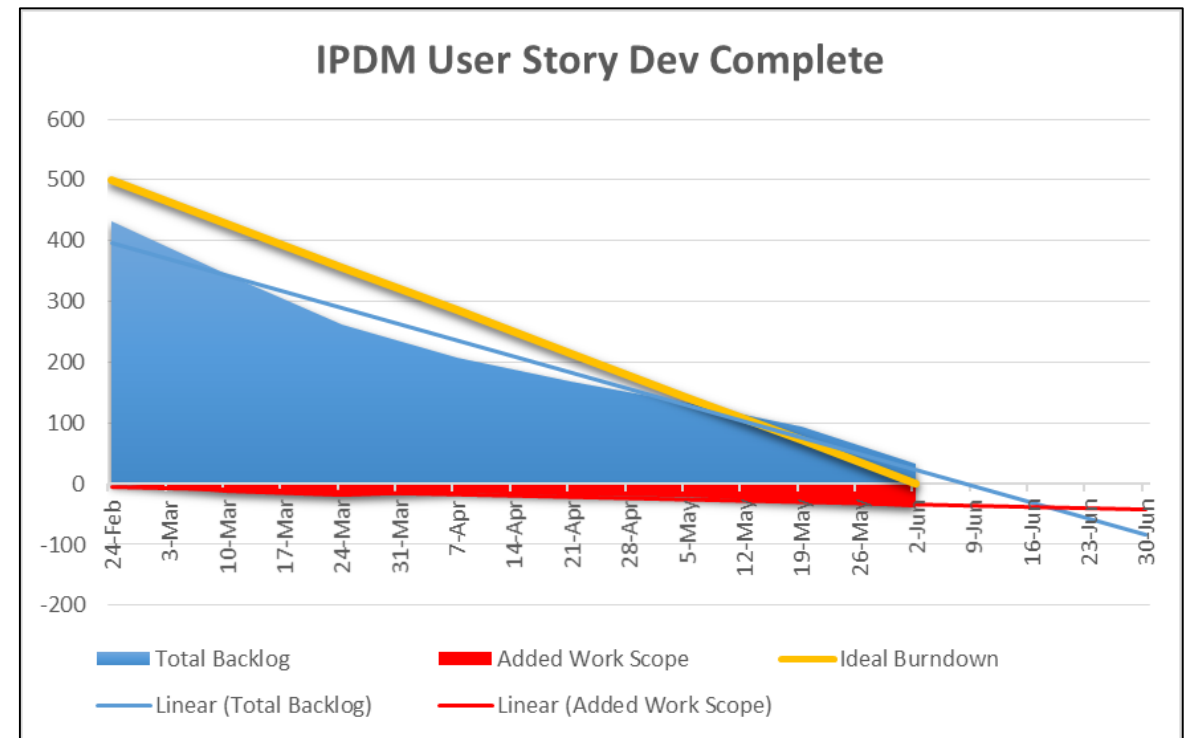
Increment 1 Retrospective

- Held a program-level retrospective to share what worked well and what didn't

- Took time to reward teams and individuals who contributed to success

- Lessons learned:

  - Do not spend hundreds of hours building a schedule that will probably change

  - It is difficult to predict what work can get finished by the end of the increment

  - While new work will inevitably emerge, product owners and customers are reluctant to remove scope from the increment to allow emergent issues to be addressed…they want it ALL

  - Disconnect still exists between product owners and customer leadership

    - E.g. severed fingers

  - The agile model is not the right solution for everything

Increment 2 – Why Aren't We Better?

Planning Event

- Had to be virtual due to budget constraints
- While teams were more familiar with agile processes the second time around; they had less time with the requirements before the planning event started

Increment 2 Begins

- Only had 4 sprints
- Overlapping deployment activities, while starting a new increment, caused issues

Lessons Learned and Moving Forward

- Increment 2 was rushed in terms of digesting the SOW and only having 4 sprints for development
- We got a lot done, but it was very stressful
- We allowed some critical development to continue – don't do this!
- Needed to do a better job of planning in the deployment support work as a part of increment planning

Where do we go from here?

- Even though the change has been hard, it's better than where we were

- We do not want to go back to the old way

- Communication has improved, but we can still make it better

- We will continue to leverage the scaled agile process, and make improvements as we go

- Culture has significantly improved –

    - Bad news is delivered more easily and people are speaking up

    - We are more actively managing risks

    - People aren't afraid to ask for help

Key Takeaways and Final Thoughts

- Executive support is critical to success
- Change needs to happen from both the bottom and the top
- The traditional project management approach needs to be assessed, many of those processes won't apply
- Stay focused on the discipline that's required to effectively manage an agile project
- Change the "status meeting" conversations to focus on the work being performed instead of providing status on a schedule
- Empower your scrum masters to hold the team accountable to following process
- It's too easy to fall back to the cultural norms when everyone around you is still behaving the "old way"
- Culture and process are equally important