SMASHING SILOS: CHANGING CULTURE IN ORGANIZATIONS

A BIRDS OF A FEATHER SESSION

- MATT MICENE, RED HAT
An unconference approach

Write topic choices → Vote → Discuss!
Key takeaways on starting a project

When your change affects the roles of a number of different teams

Who in the room ran into obstacles?
Who in the room overcame them?
What’d we say:

- Get executive buy in for air cover and authority
- Executive buy in on it's own isn't enough
- "Stakeholders" all speak different languages and have their own ideas about value
- Most of the people you need to talk to are not on your team, you need to broaden your audience
What’d we say:

- Measuring progress and success helps people who’s roles are changing
- Process change can be iterative just like the development process
  - Can you build a Minimum Viable Process?
  - Can you show smaller changes that make progress?
  - If not, can you implement on a per team basis instead?
- Sometimes changing the meaning of success has to happen (esp if the needs of the products are different than the team has worked on in other roles)
- Agreeing on the right things to measure keeps everyone in sync
Not my team, not my problem

When the Business creates silos where none existed before

How to keep teams talking?
How to keep teams focused on the right things?
What’d we say:

- Keep lines of communication open between teams, especially where there are dependencies
  - Even if these have to be unofficial
- Understand the perceived value in the new silo
- "One team, One goal" and that's the success of the business not the output of a single team
- if the business fails, you won't be applauded for having hit your project plans
What’d we say:

- Is it a personnel problem?
  - Is there one problem manager who needs to be worked around?
  - Is there a trusted chain of command that issue can be raised through?
- Matrix teams can help with management silo mentality while allowing dependent work to be done cross team
- Resources can be a problem, if you ask for time be prepared to offer it in return
20 scrum teams, 20 sets of libraries. This is a problem

When your choices affect other teams

How can we get some standards across projects?

How can we reuse instead of rebuilding?
What’d we say:

- How do you publish and discover what other teams are working on?
- Innersourcing? An internal "Linux Foundation" that registers projects?
- Is an internal repository enough? Will we need to maintain more tools?
- Do you have an internal "Not Invented Here" problem with a team?
- Documenting the internal usage is important
  - What’s current, what’s outdated
What’d we say:

- Building vs consuming from the outside, which is right?
- Supporting internal usage can become a full time job
- Knowing what outside libraries are safe and healthy can be a challenge
- Service discovery is OK, but need more than just APIs
- Don't fork data if you unless really need to, and even then, don't
Thanks for participating!