

Team 4 Organizational Impediments

In this exercise, your team's job is to figure out how a scrum master would diagnose and address an organizational impediment. You will do this by following the three C's approach:

- Complaint
- Cause
- Correction

Read Your Team's Scenario

Scenario Four

The core 'secret sauce' in one of the company's products was **written by Sue**, one of the founders. The code is Prolog, a relatively **obscure programming language**. The company has three products, mostly implemented in Java. **Each product has its own scrum team**. They all depend on that original **Prolog code**. Sue is the only person in the company familiar with that code.

The scrum teams are frequently blocked waiting on Prolog changes. The product owners are upset that new **features and fixes take so long**. The members of the **teams often get into squabbles over who should get attention from Sue next**. It's causing a sense of rivalry and even resentment between the teams. **Sue wants to retire, but feels obligated to stay**. She can't abandon the company that she helped start.

Complaint

Complaints are the symptoms, or pain points, that alert us to the presence of an impediment. In your scenario, what complaints and pain points are people experiencing?

<things are taking too much time to deliver
Teams complaining about Sue's availability
Sue is unable to retire>

Cause

The cause is the underlying reason the complaint exists. The causes of organizational impediments often include: organizational structure, policies, politics, process, powerful personalities, and/or shared technologies.

A complex organizational impediment may have more than one cause. A good starting place is to identify ways that the empirical scrum pillars of transparency, inspection, and adaptation are not working effectively. In this scenario, where do you see failings of transparency, inspection, and adaptation?

<Prolog code base is not well maintained
Sue is founder and code-owner, may be unwilling to change code or there is not enough time taken
Lack of shared knowledge in the code
Lack of ownership to a common library and persistent management
>

Now take your analysis deeper. One way to find root causes is by repeatedly asking ‘why?’ Start with a problem, then ask “Why is that so?” Take the answer you find and again ask “Why is that so?” In your scenario, what do you believe is the underlying root cause(s)?

<Your answer goes here>

3 teams with common dependency and frustrated with each other
Single point of failure is a bottleneck
The original codebase
Prioritization problems

Correction

The correction is an intervention to address the cause, and ultimately relieve the complaints. Try to come up with three possible solutions. For each possible solution, describe:

- What needs to happen
- Who should be involved
- How a scrum master would begin to implement the solution
- What problems might arise with this approach

<Shared ownership of codebase>

<KT sessions to spread knowledge>

<Change the source code of one or the other. Prolog to Java, or Java to Prolog, or a common third choice to cover both>

Which of your solutions do you think is best?

<Knowledge transfer/ cross training with assigned lead from each of the three product team>

Prepare To Present Your Findings

Your team will be sharing your scenario, complaints, cause(s), and your best solution with the rest of the group. Decide who will present each section. The time box for your presentation will be two minutes.

AGILE LEARNING LABS

© 2020 Agile Learning Labs

AGILE LEARNING LABS

© 2020 Agile Learning Labs