Experiments in Scaled Planning

7/11/2024

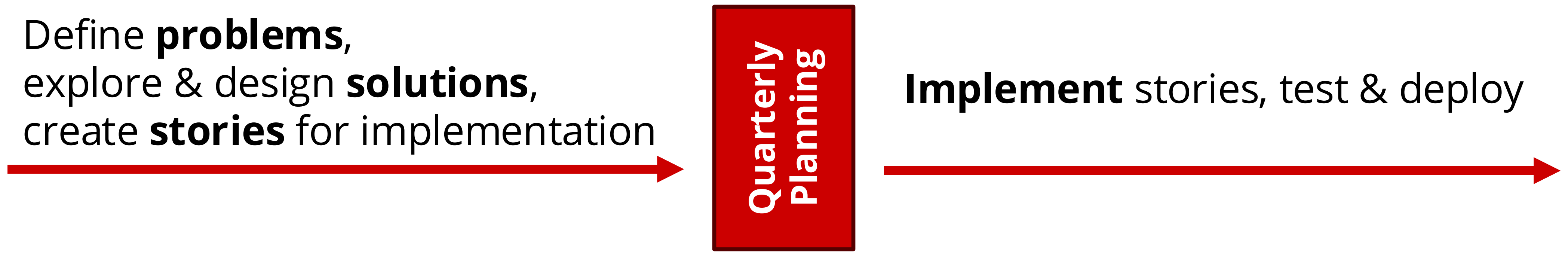
Heidi Nibouar

Agile Education

powered by scruminc.

The Context: Setting the Scene

- NWEA, an education non-profit
- 19 Scrum teams across 4 Value Streams in Product Development
- Using a framework for scaled Agile planning for 3 years, alongside an established end-to-end workflow



Define **problems**,
explore & design **solutions**,
create **stories** for implementation

Quarterly Planning

Implement stories, test & deploy

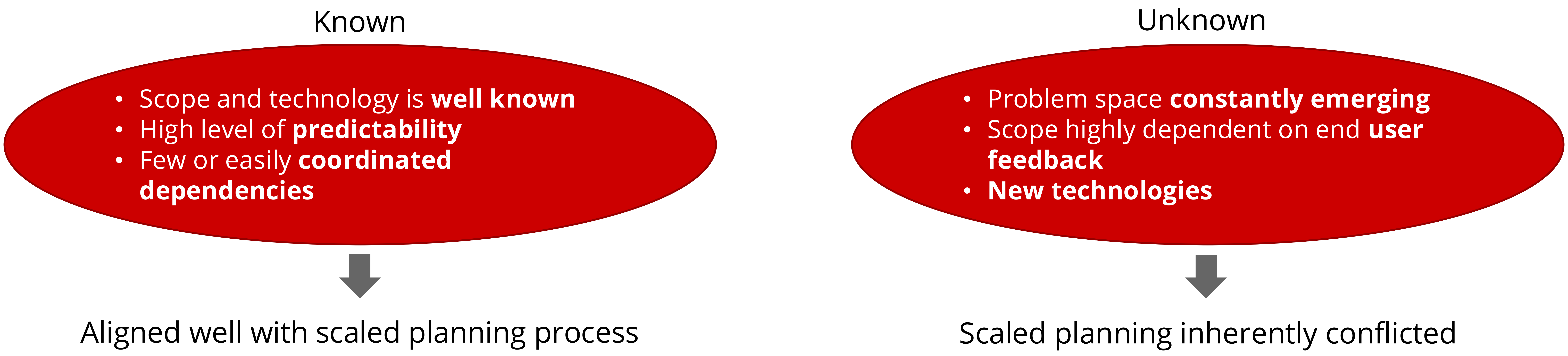
Established the new Product Value Workflow (PVW) to increase:

- iteration based on end user feedback
- development team ownership
- frequency of value delivery

Case Study By: **Heidi Nibouar**

The Problem: PVW and Scaled Planning

Two primary types of efforts go through the PVW:



Known

- Scope and technology is **well known**
- High level of **predictability**
- Few or easily **coordinated dependencies**

Aligned well with scaled planning process

Unknown

- Problem space **constantly emerging**
- Scope highly dependent on end **user feedback**
- **New technologies**

Scaled planning inherently conflicted

The problem: our scaled Agile planning framework conflicted with the PVW for “unknown” work.

Case Study By: Heidi Nibouar

The Problem: PVW and Scaled Planning

Through retros, 1:1 conversations and surveys, we identified the following issues with unknown work:

1. Low morale

   >> Teams felt **forced to commit** when they didn't feel comfortable and required **heroic efforts** to complete on time

   >> With the focus on delivery plans only, Scrum teams felt disconnected from **Product Mgmt, Design, and other critical team members**

2. Poor user experience

   >> **No ability to iterate** on and improve the product based on user feedback

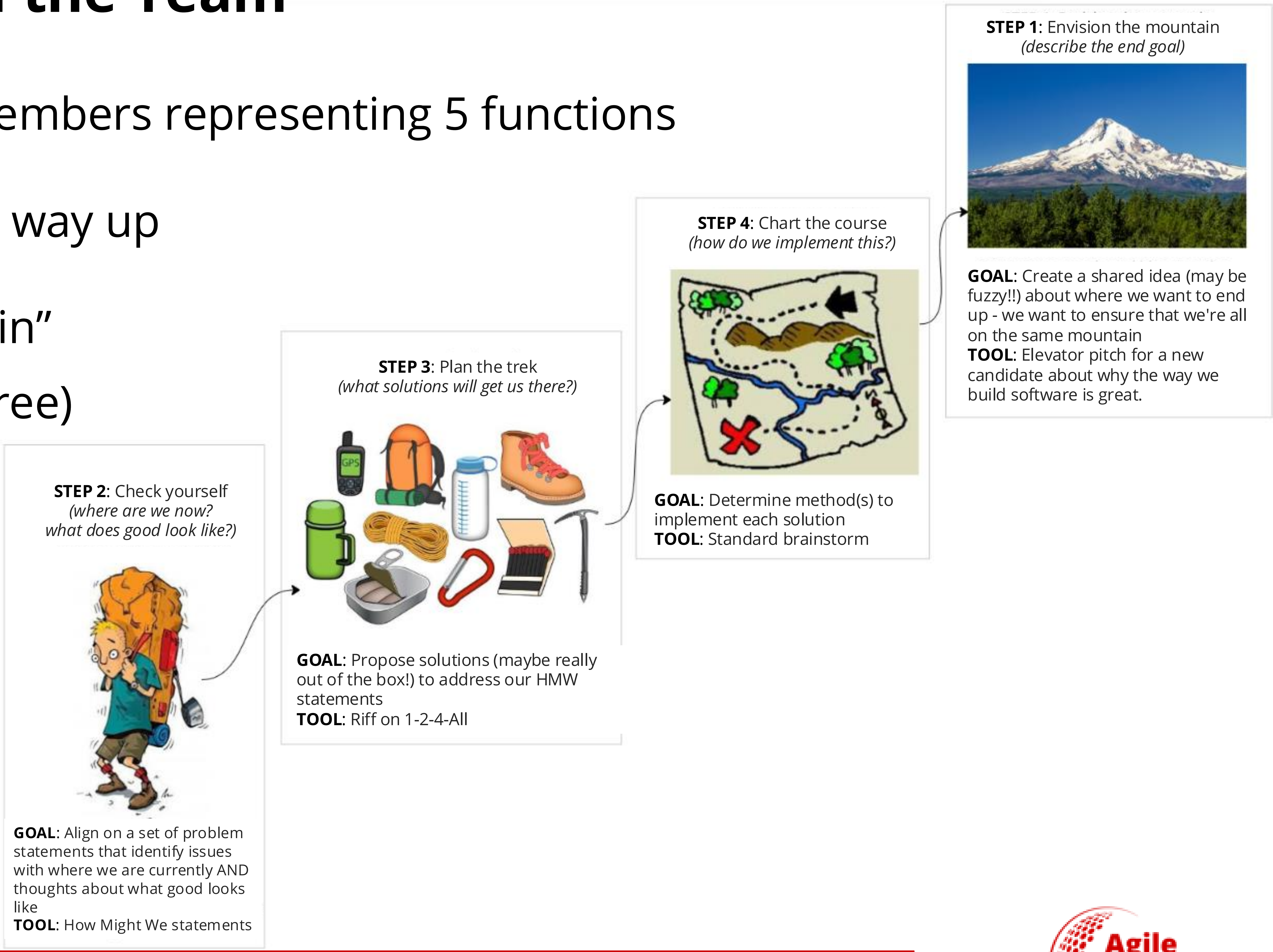
3. Missed milestones

   >> Unknown dependencies resulted in churn, **not delivering on commitments**, and impact to other teams' commitments

Case Study By: **Heidi Nibouar**

The Solution: Workshop WITH the Team

- Solutioning workshop with 12 team members representing 5 functions
- Created an end vision and worked our way up
- Resulting proposal: “Increment Check-in”
  - One month planning horizon (not three)
  - Focus on one distinct effort at a time
  - Integrate all team members (Design, Product, etc) rather than solely development teams



STEP 1: Envision the mountain
(describe the end goal)

GOAL: Create a shared idea (may be fuzzy!!) about where we want to end up - we want to ensure that we're all on the same mountain
TOOL: Elevator pitch for a new candidate about why the way we build software is great.

STEP 2: Check yourself
(where are we now? what does good look like?)

GOAL: Align on a set of problem statements that identify issues with where we are currently AND thoughts about what good looks like
TOOL: How Might We statements

STEP 3: Plan the trek
(what solutions will get us there?)

GOAL: Propose solutions (maybe really out of the box!) to address our HMW statements
TOOL: Riff on 1-2-4-All

STEP 4: Chart the course
(how do we implement this?)

GOAL: Determine method(s) to implement each solution
TOOL: Standard brainstorm

Experiment: Let’s try this out with one Initiative (and one Scrum team)

Case Study By: Heidi Nibouar

The Solution: Increment Check-in Agenda

1. Business & Product context setting and changes

2. Product, Research, & Design:
   - What was completed
   - What is planned

3. Delivery:
   - What was completed
   - What is planned

4. Quick Retro

Case Study By: Heidi Nibouar

Agile Education powered by scruminc.

The Results: Initial Concerns

- Initial anxieties…

How am I going to prepare other departments with only a month's worth of planning?

So now I have to pay attention to multiple planning sessions?

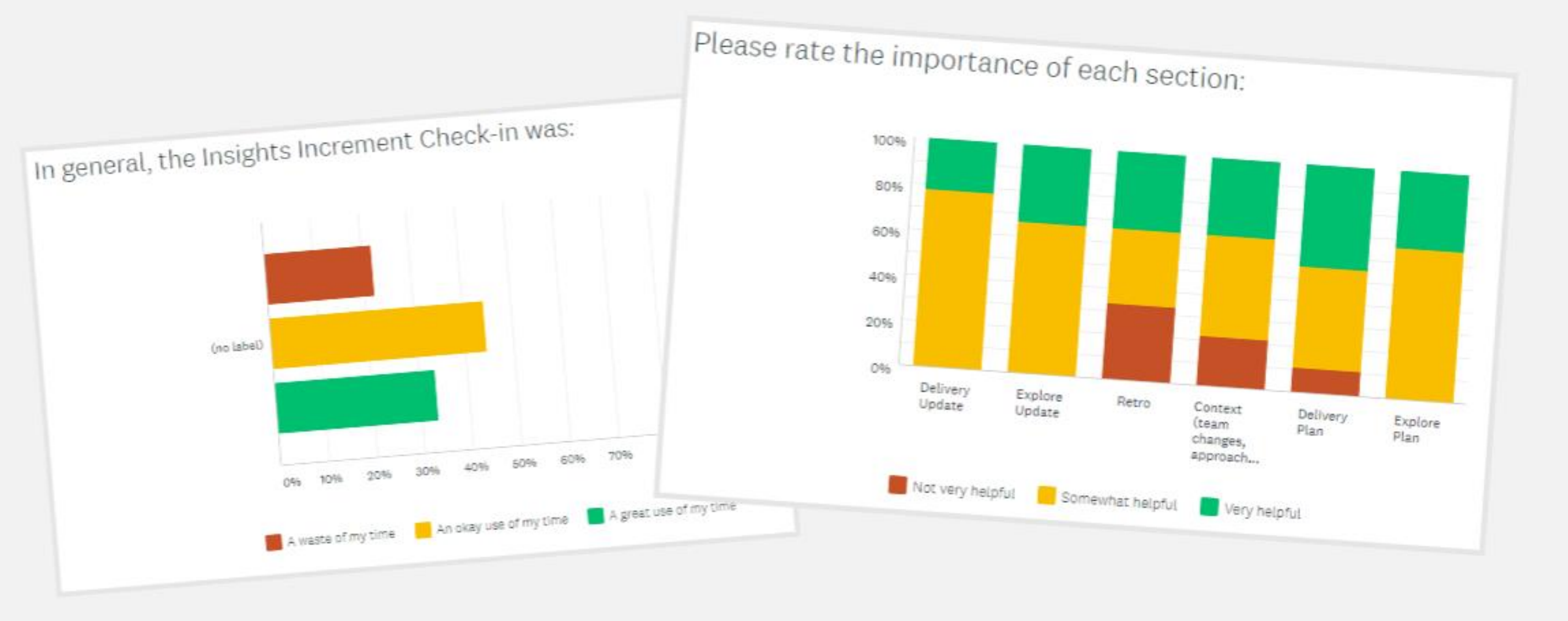
But what if a team is working on more than one initiative?

What happens when a new priority pops up or there's too much tech debt?

Case Study By: **Heidi Nibouar**

The Results: So What Happened?

… led to formation of a cross-functional team that focused on improvement for the first year, processing quantifiable and targeted retro feedback



Case Study By: Heidi Nibouar

©Jeff Sutherland & Scrum Inc 1993 – Present

Agile Education powered by scrumInc.

The Results: Show Me the Numbers

- 6 month poll revealed:
  - 70% of dev team members felt less pressured to commit to unknown work
  - 84% of team members reported that they were more able to change direction based on direct feedback
  - 84% of team members saw increased ability to inspect and adapt cross-team level processes
  - 65% saw an improvement in collaboration between upstream and downstream teams
- Since a second team was added to the Check-in (6 months after initial kickoff), 0 deliverables have been missed due to unidentified dependencies.

Employee engagement score on impacted teams rose from **-16% prior to 60% after** implementation

0 instances of MVP not meeting users needs as seen before

Case Study By: Heidi Nibouar

In Conclusion

- We experienced issues when trying to blend a fairly rigid scaled Agile planning framework with our new, more fluid, end-to-end workflow
- To get answers, we went to the impacted team to create an initial solution, which has been continually iterated on
- Over time, the changes to the planning process for the impacted teams have led to
  - Increased collaboration between dev teams and with their upstream counterparts
  - Increased ability to pivot, which ultimately provides higher value for the customer
  - Higher quality deployments
  - Higher morale across team members