Automating Continuous Planning in SAFe

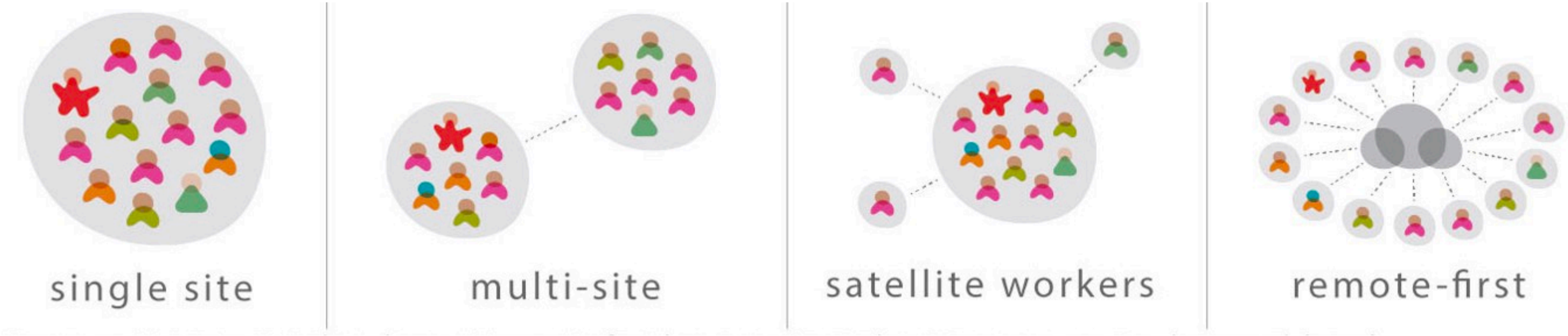
Darius Foo, Jonah Dela Cruz, Subashree Sekar, Asankhaya Sharma

Veracode

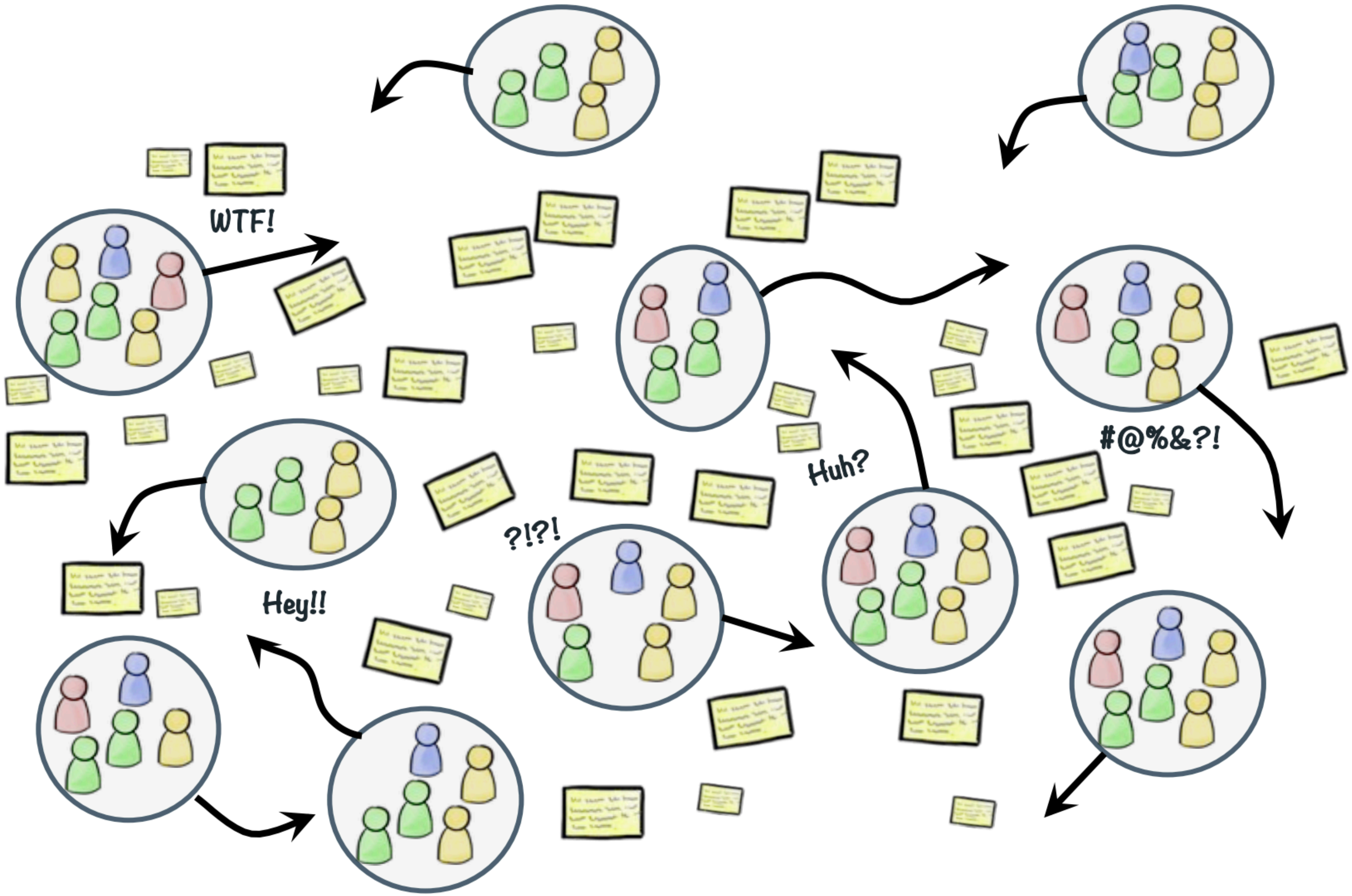
Agenda

- Motivation
- Sapling
- Demo
- Future work

Agile teams



Source: Martin Fowler, http://martinfowler.com/articles/remote-or-co-located.html


WTF!
?!?!
Hey!!
Huh?
#@%&?!

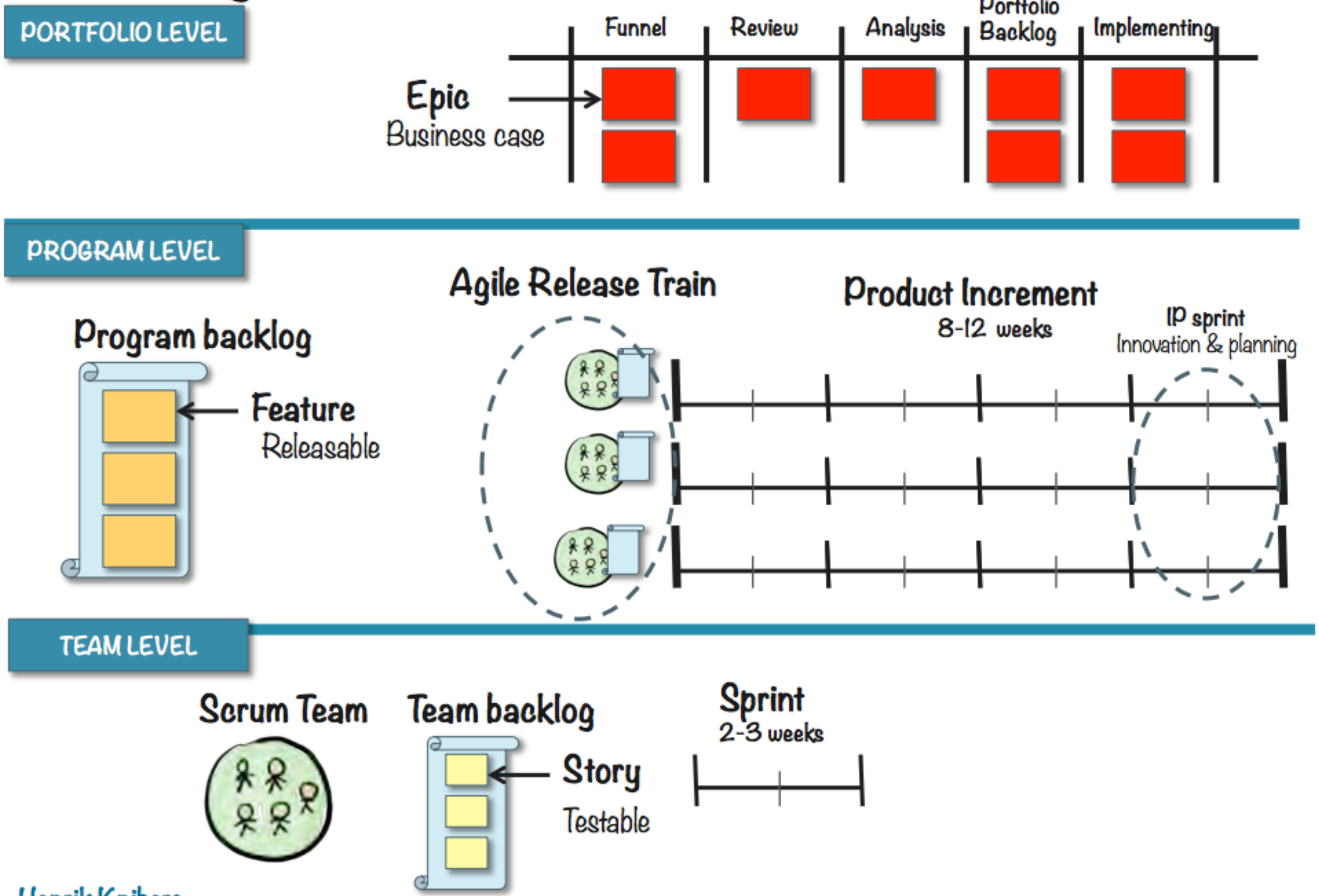
SCRUM
@SCALE

SAFe®

Spotify®

LeSS

Scaled Agile Framework



PORTFOLIO LEVEL

Portfolio Kanban System

Funnel	Review	Analysis	Portfolio Backlog	Implementing

Epic
Business case

PROGRAM LEVEL

Program backlog

Feature
Releasable

Agile Release Train

Product Increment
8-12 weeks

IP sprint
Innovation & planning

TEAM LEVEL

Scrum Team

Team backlog

Story
Testable

Sprint
2-3 weeks

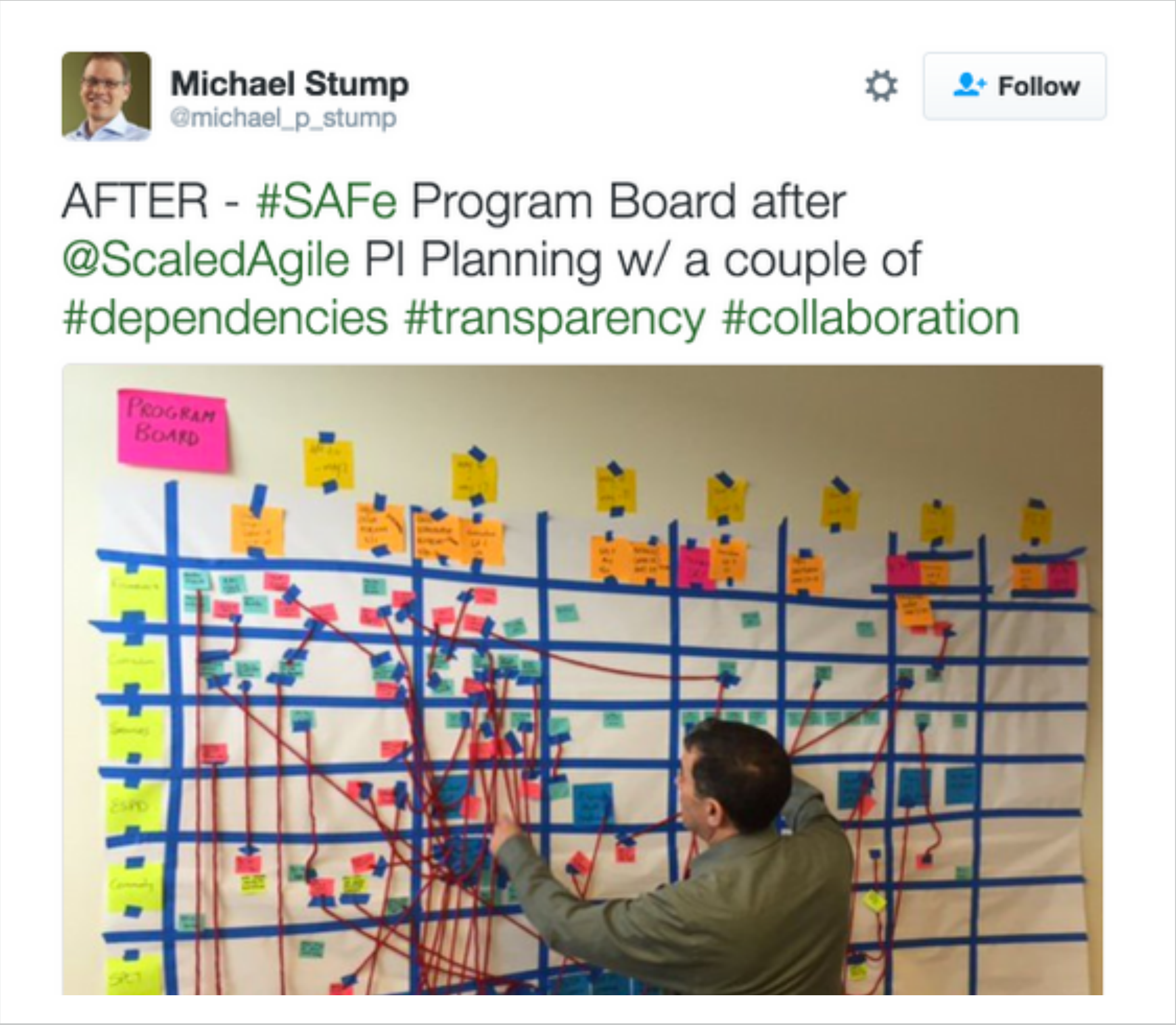
Henrik Kniberg

PI Planning

- Meet face-to-face
- Align goals
- Cross-team dependencies
- Match demand to capacity

Limitations

- Visibility across geos
- Cross-geo dependencies
- Manual digitization



Michael Stump
@michael_p_stump

AFTER - #SAFe Program Board after @ScaledAgile PI Planning w/ a couple of #dependencies #transparency #collaboration

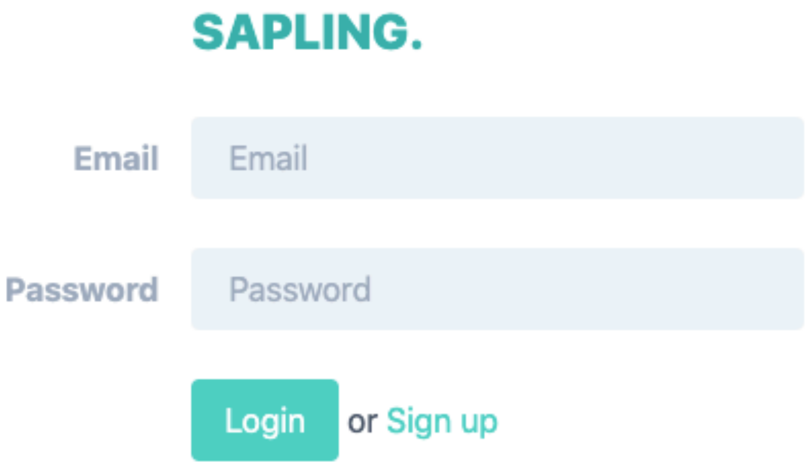
Continuous Planning

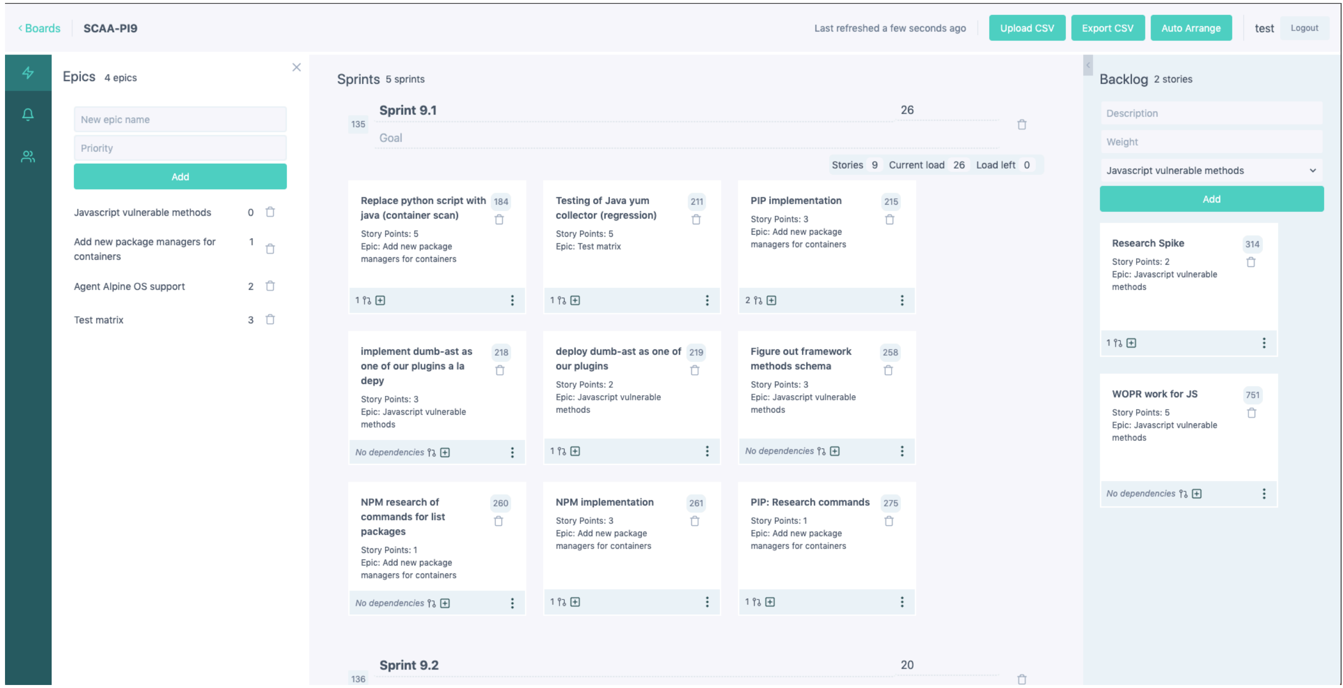
- Similar in spirit to CI/CD
- Global visibility into a continuously-updated plan
- Automate away overhead
- Transparency and accessibility

Sapling

A tool for Continuous Planning

Special support for SAFe PI Planning





Epics 4 epics

Javascript vulnerable methods	0	🗑
Add new package managers for containers	1	🗑
Agent Alpine OS support	2	🗑
Test matrix	3	🗑

Sprints 5 sprints

Sprint 9.1 135 26

Goal

Stories 9 Current load 26 Load left 0

Replace python script with java (container scan) 184
Story Points: 5
Epic: Add new package managers for containers
1

Testing of Java yum collector (regression) 211
Story Points: 5
Epic: Test matrix
1

PIP implementation 215
Story Points: 3
Epic: Add new package managers for containers
2

implement dumb-ast as one of our plugins a la depy 218
Story Points: 3
Epic: Javascript vulnerable methods
No dependencies

deploy dumb-ast as one of our plugins 219
Story Points: 2
Epic: Javascript vulnerable methods
1

Figure out framework methods schema 258
Story Points: 3
Epic: Javascript vulnerable methods
No dependencies

NPM research of commands for list packages 260
Story Points: 1
Epic: Add new package managers for containers
No dependencies

NPM implementation 261
Story Points: 3
Epic: Add new package managers for containers
1

PIP: Research commands 275
Story Points: 1
Epic: Add new package managers for containers
1

Sprint 9.2 136 20

Backlog 2 stories

Research Spike 314
Story Points: 2
Epic: Javascript vulnerable methods
1

WOPR work for JS 751
Story Points: 5
Epic: Javascript vulnerable methods
No dependencies

PI Planning workflow

- Teams begin with a prioritized list of *epics*
- Time is divided into *sprints*
- Epics are divided into *user stories* and assigned to sprints
- Stories are assigned *story points*
- Sprint *capacities* should be respected

Story dependencies

- Within teams
  - e.g. “Research” item should always precede an “Implementation” item
- Across teams
  - e.g. platform team should finish API before frontend team uses it
  - Requires communication
- PI Planning is all about surfacing and negotiating cross-team dependencies

Enabling Continuous Planning

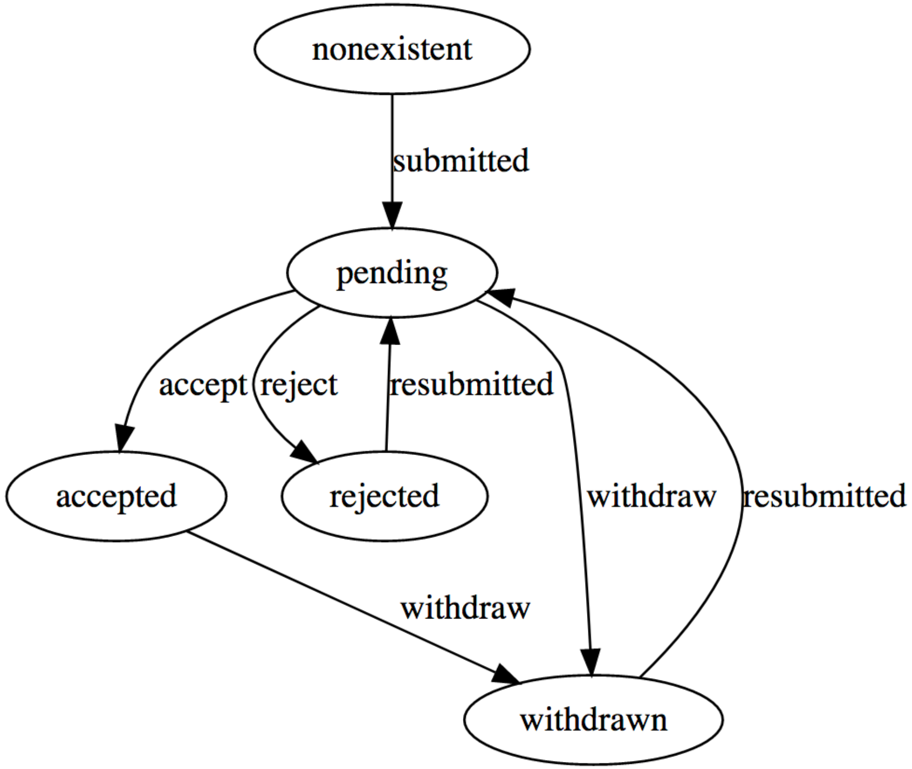
- Sapling supports the workflow, allowing users to edit stories, sprints, epics, and boards
- Stories are assigned to sprints automatically, so the plan is always up to date
- Teams can request to add stories to each other’s boards
- ✓ Dependencies across the entire organization are always visible
- JIRA import/export

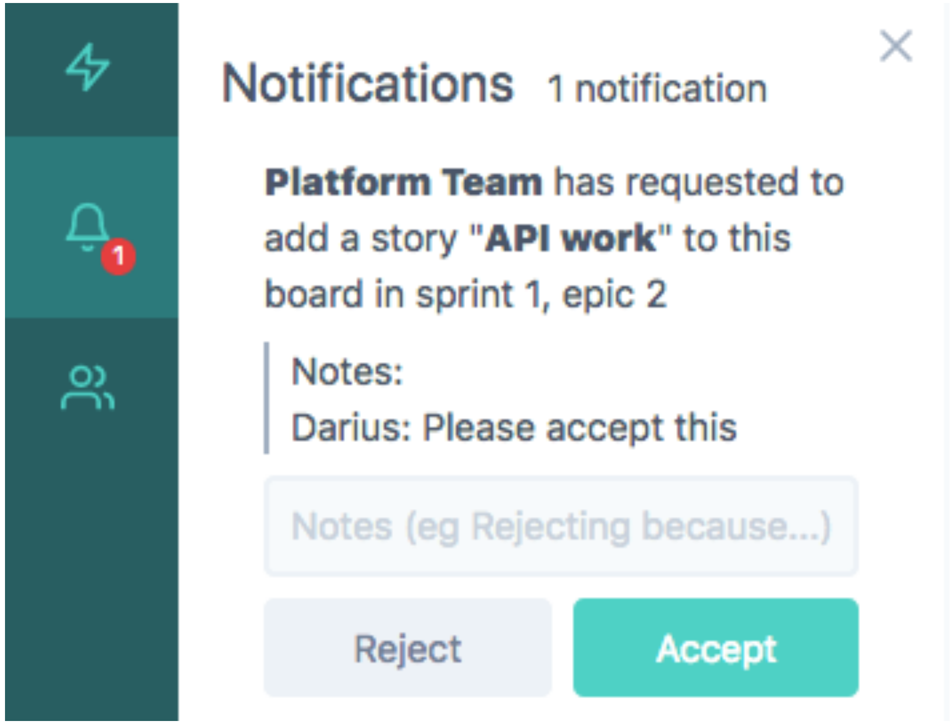
Story Assignments

- Hard constraints (integrity constraints):
    - Sprint capacity
    - Dependencies
    - Pins
- Soft constraints (optimization):
    - Epic priority: minimize stories in lower-priority epics before higher ones
    - Context: minimize distance between dependent stories
    - Parallelism: maximize number of people working at the same time
    - Conservativeness: maximize earlier sprint assignments over later ones

Story Requests

- Structures the process of negotiating dependencies



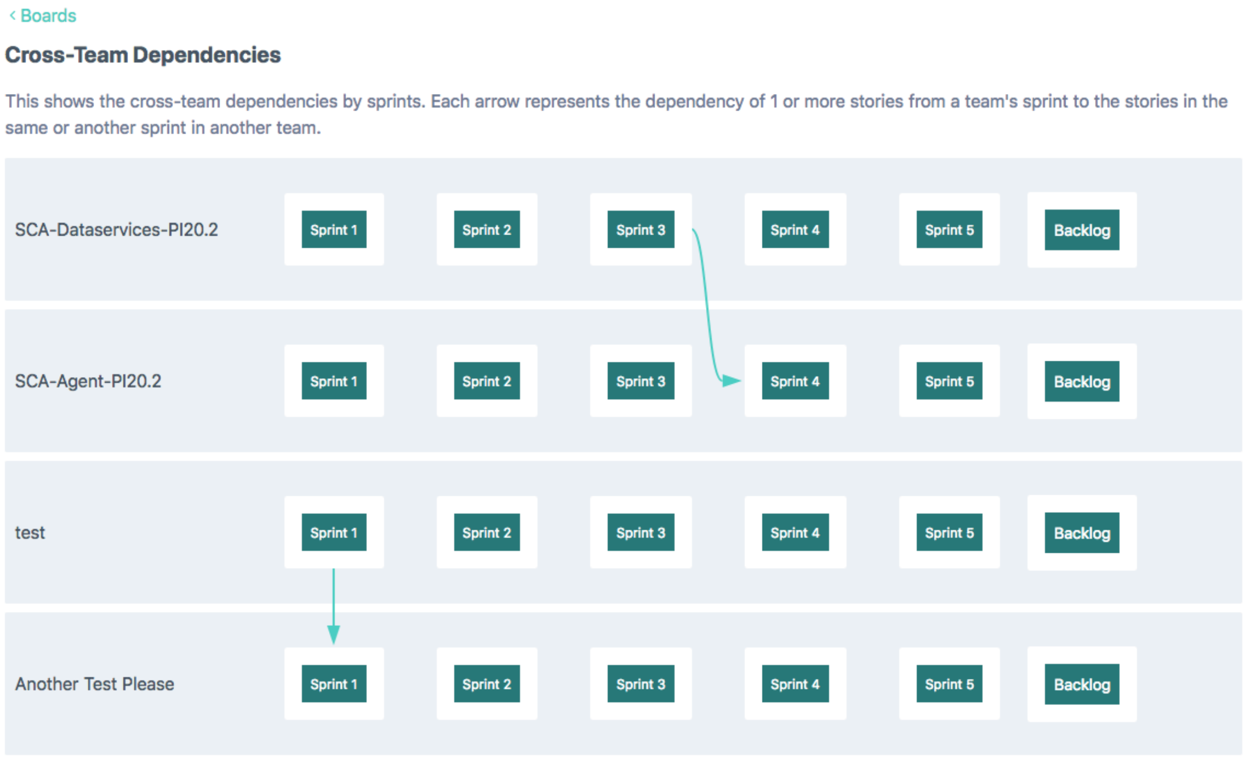


Platform Team has requested to add a story "**API work**" to this board in sprint 1, epic 2

Notes:
Darius: Please accept this

Story Requests

- Visualization of cross-team dependencies



‹ Boards

Cross-Team Dependencies

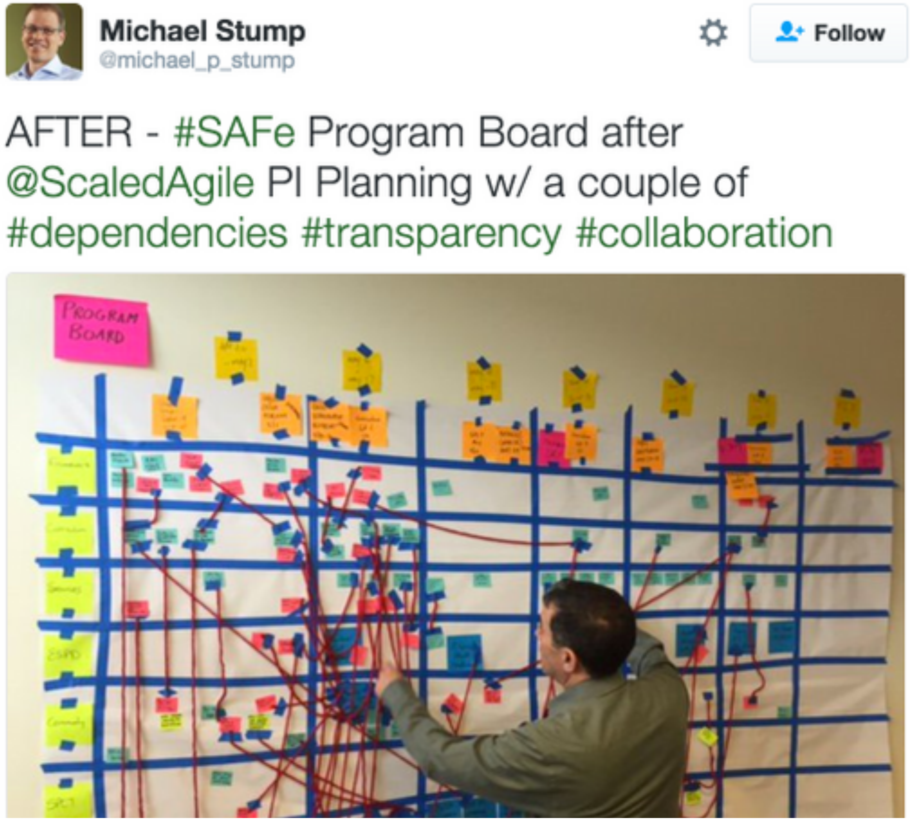
This shows the cross-team dependencies by sprints. Each arrow represents the dependency of 1 or more stories from a team's sprint to the stories in the same or another sprint in another team.

SCA-Dataservices-PI20.2 | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Backlog

SCA-Agent-PI20.2 | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Backlog

test | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Backlog

Another Test Please | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Backlog



Michael Stump
@michael_p_stump

AFTER - #SAFe Program Board after @ScaledAgile PI Planning w/ a couple of #dependencies #transparency #collaboration

Demo

Future work

- Stability and explainability (UX) of solver decisions
- Previews
- Incremental solving (allowing non-optimal solutions)
- Deeper JIRA integration

Thanks!

https://github.com/srcclr/sapling