Table of Contents

Johanna Rothman - Agile Program Management 3
Summary 3
Action / Learning 3
Presentation 3
Note 3

Johanna Rothman - Agile Program Management

Do you have measurement dysfunction on your program? Are you trying to measure teams and extrapolate each team's status to the program? That doesn't work. Teams have personal status, and you can't add them together to understand the program state. You can use a handful of program measurements that help everyone understand where the program is and where it's headed.

Instead of trying to "scale" measurements, take a new approach. In this talk, Johanna will share program measurements—qualitative and quantitative—that show everyone the program state.

Summary

- Content rating (0-no new ideas, 5 - a new ideas/approach, 9-new ideas): 5
- Style rating (0-average presentstion, 5 - my level, 9-I learned something about presenting): 3

Action / Learning

- Quote - "People ask for predictive measures when they don't trust you to deliver" Johanna Rothman
- Book "This is lean"
- Write about this Individual Pay performance - "We are nuts" - Arguing about $500 or $1000 for a person that is paid $100K - In the rounding error
- Idea of action items as a Kanban board (measure - speed of decisions)

Presentation

Original: https://submissions.agilealliance.org/attachments/2569

My copy: alpm.measurestoseevalueanddelivery.r1.pdf

Note

Measurement problems with in programs

Program - several projects / one business objective

Several structures depending on structure of program

“People ask for predictive measures when they don't trust you to deliver” Because cannot depend on stream of value

Managers care about Customer acquisition / retention Revenue Customer experience

Change conversations for managers How much cost → how much invest (target date or budget) Are you on track, what the earned value → show working product When will we see revenue → show you value now / release now (when you send the bills out) What do the customers think → progress against release criteria, customer sat data

Story point / velocity “Do you want story point or do you want working software” - courageous team Measure of capacity, not productivity Not always predictable Individual to team

Team velocity is personal to team Team have different cycle time and lead times because they do different work We don't normalize stories inside or across teams

Can we measure learning Software is learning, innovation - maybe we have enough now We learn to build momentum

Hardware not done until in physical form

We need to learn to build momentum Earlier everyone can see something releasable, the more you build momentum

Use trend data, not snapshot data

Measure what you want to see and where you want to go First get to running testing features over time

Not traffic light management Until you see something. It's yellow They are for cars not programs

Measure completed, remaining and total features over time Including adding features Measure as a count Depends on deliverables, not epics or themes

Epics and themes - We can wave the hand, I can wave this way to

Product backlog burn up Feature set, stacked on top of one another. How much per unit time, show features grow, done as count Where are we up to



Can be done with stickies - one per feature Total and how much done

Added features for free Drive further metaphor - don't get the same release data

What would it take to release every day At least release every month

Agile roadmap in the large 6 quarter roadmap

Then agile roadmap in the small One quarter agile roadmap Deliverable based planning - small slices through the architecture Specific value for different users Rolling wave plan as not sure

How far out can we plan

Want less of WIP - CFD Defects Other less of - multi-tasking

How many projects are you working on - add sticky to wall once a week with number of projects

People respond to what you measure Measure what you want more of and less of Tell people which is which

Spike not WIP as it is learning

Program measurements Run rate / week and month - check against the target or investment question Program level WIP as feedback for program - product owner value team, feature teams momentum

Product owner value team Product manager Plus product owners - the deliverables

Release frequency

External release as a business decision Internal release - can you use continuous delivery inside your org If not how often can we release

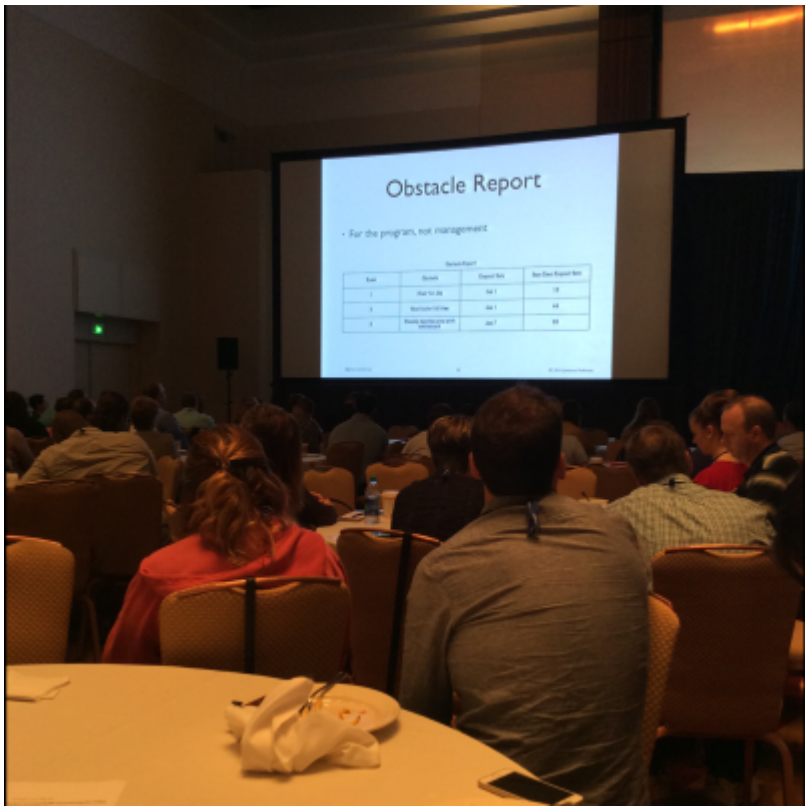
Measure How long does it take to move from build to release Build time and is it increasing or decreasing

Cycle time (interdependencies show up here) - how long take a feature get in into release column If not feature teams then cycle time could be long

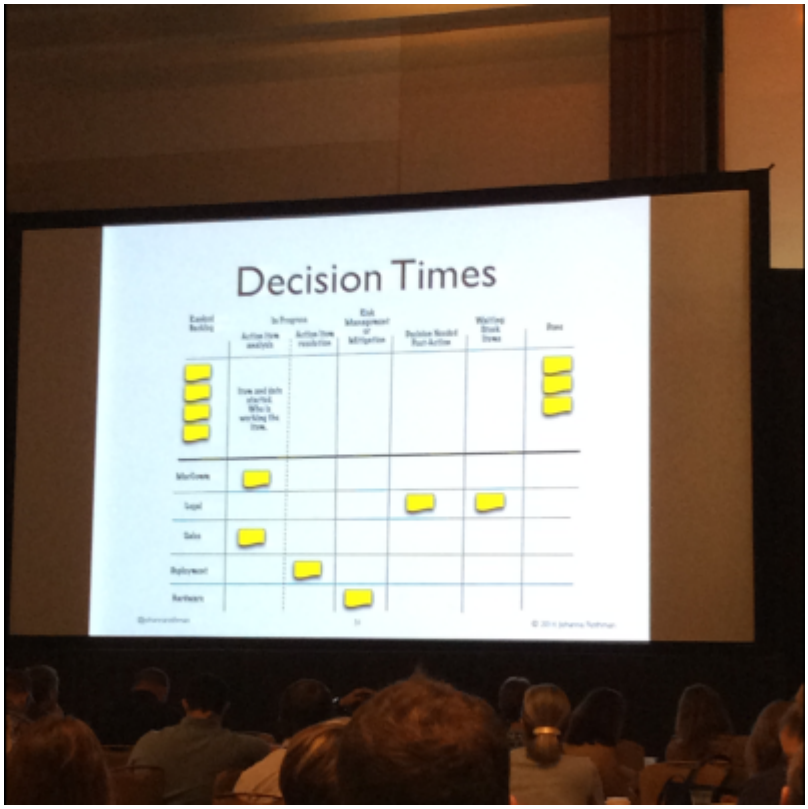
Product measures Create scenarios for what is important for your product Performance, reliability, quality attributes of the product Can be used as a release criteria

Qualitative measurements Customer feedback / happiness How often you get feedback from customers Obstacle report How long it takes to make a decision

Obstacle report



Decision times Kanban board for action items



Measurement traps Never attempt to measure anyone or any teams productivity (meaningless)

What is productivity? Features over time Teams take features not individual

Personal productivity is relevant because we

Resource efficiency is based on experts not flow

What flow efficient

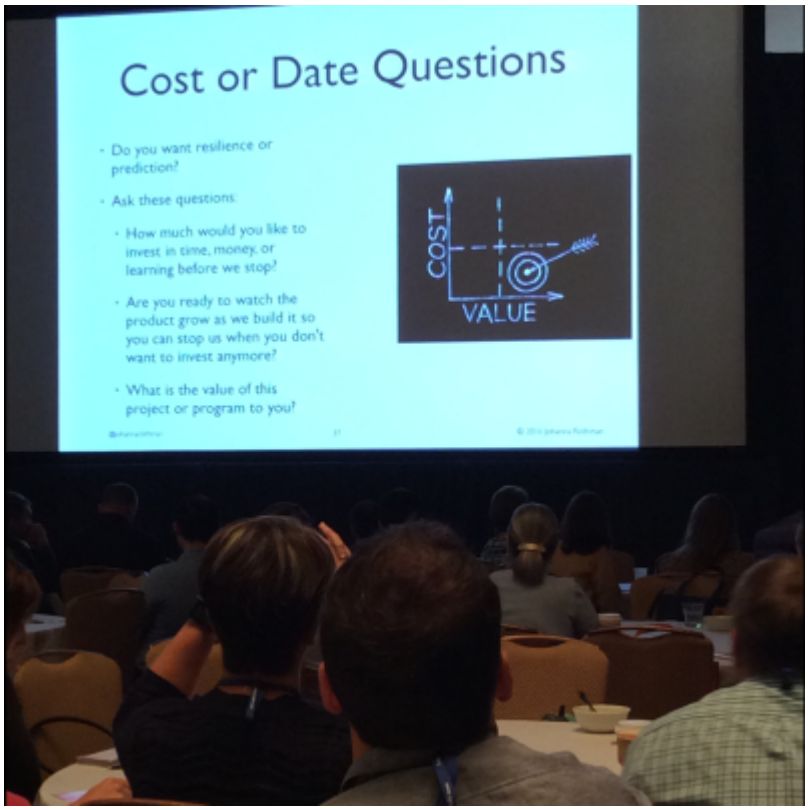
Throughput

Book “this is lean”

Want flow efficiency Increase throughput Decrease WIP

Cost or date question

Do you want resilience or prediction?



If you must estimate 3 point estimate Swag Add percentage confidence

15% in April, 30% in May

And give a date for the next estimate

Deliver often to build trust Maybe release early

Individual Pay performance “We are nuts” Arguing about $500 or $1000 for a person that is paid $100K In the rounding error

Your are a team - hear is a pot of money for the team

Program, Executive, Conference, Agile2016, Measurements

From:
https://www.hanssamios.com/dokuwiki/ - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:
https://www.hanssamios.com/dokuwiki/johanna_rothman_-_agile_program_management

Last update: **2020/06/04 11:16**