A Manager's View of Scrum

•••

AmsterdamPHP
20 March 2025

Today:

Which practices separate great Scrum teams from others?

Today:

Which practices separate great Scrum teams from others?

Top 10 things the best Scrum teams get right every time!

You won't believe nr. 2!!

Find out how to get started inside.

Practicalities

A barely organized list of good things to do.

Questions welcome.

QR-code for slides at the end

Who am I

Jakob Buis

~~Developer~~
~~Team lead~~
~~Engineering Manager~~
Management consultant

Professional team builder

Should you listen to me?

Yes, because:

Never been fired

Herd of 6 elePHPants

Worked with 15+ Scrum teams in various companies & industries

As a developer, manager and Scrum Master

Professional Scrum Master II

Should you listen to me?

Yes, because:

Never been fired

Herd of 6 elePHPants

Worked with 15+ Scrum teams in various companies & industries

As a developer, manager and Scrum Master

Professional Scrum Master II

No, because:

I fuck up, a lot

Worked with 16 teams, mostly in smaller companies (< 300 people)

Most of my ideas come from other people (links included!)

Understanding Scrum

Daily Scrum

Definition of Done

Developer

Empiricism

Increment

Product Backlog

Product Goal

Product Owner

Sprint

Sprint Backlog

Sprint Goal

Sprint Planning

Sprint Retrospective

Sprint Review

Technical debt

1

Experiment beyond Scrum not before Scrum

The SCRUM Police

Daily Scrum on Mon & Wed

Demo work not completed

Retrospective every 3rd sprint

Daily Scrum on Mon & Wed
Demo work not completed
Retrospective every 3rd sprint

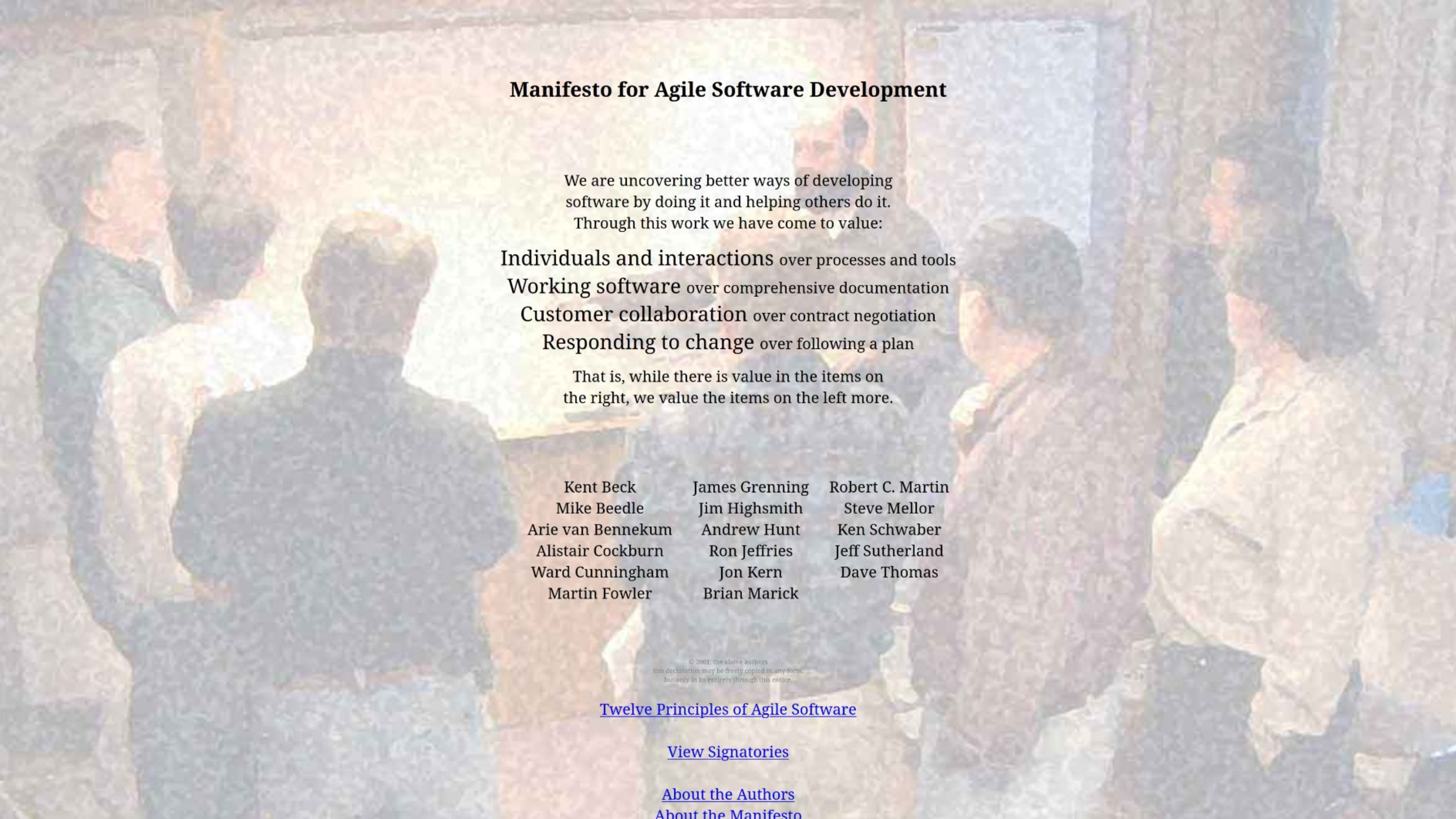
Scrum doesn't work for us

1

Experiment beyond Scrum not before Scrum

2

Working, tested software every sprint



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Twelve Principles of Agile Software

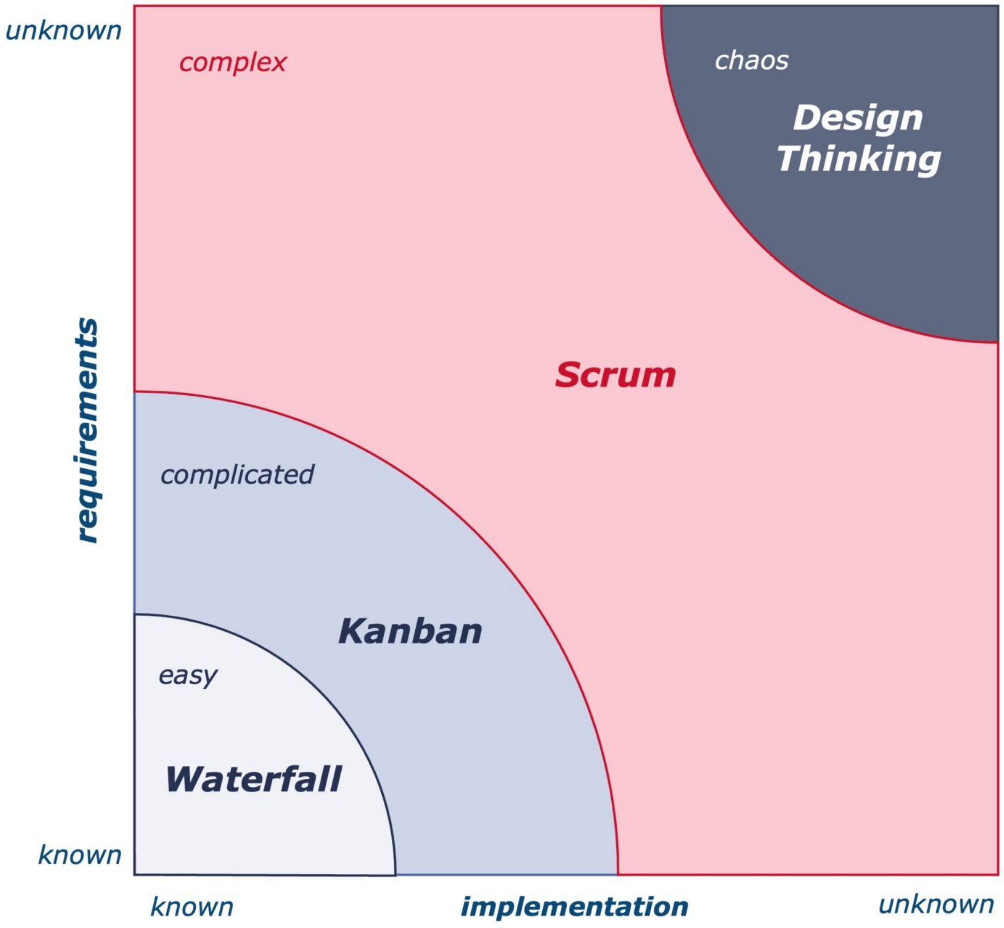
View Signatories

About the Authors
About the Manifesto

Principle 1:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Stacey Matrix



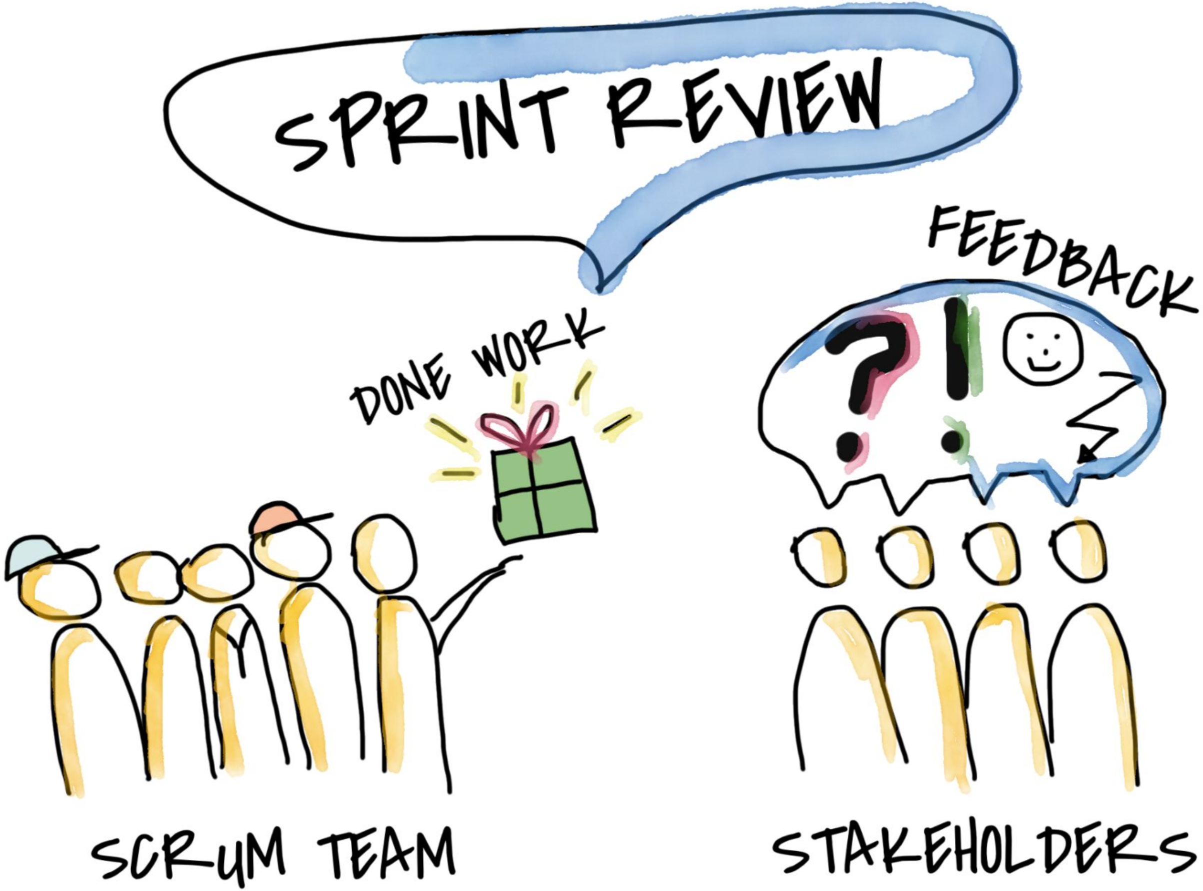
Principle 7:

Working software is the primary measure of progress.

Principle 7:

Working software is the primary measure of progress.

in the customers hands!


SPRINT REVIEW
DONE WORK
FEEDBACK
?!
SCRUM TEAM
STAKEHOLDERS

To do:

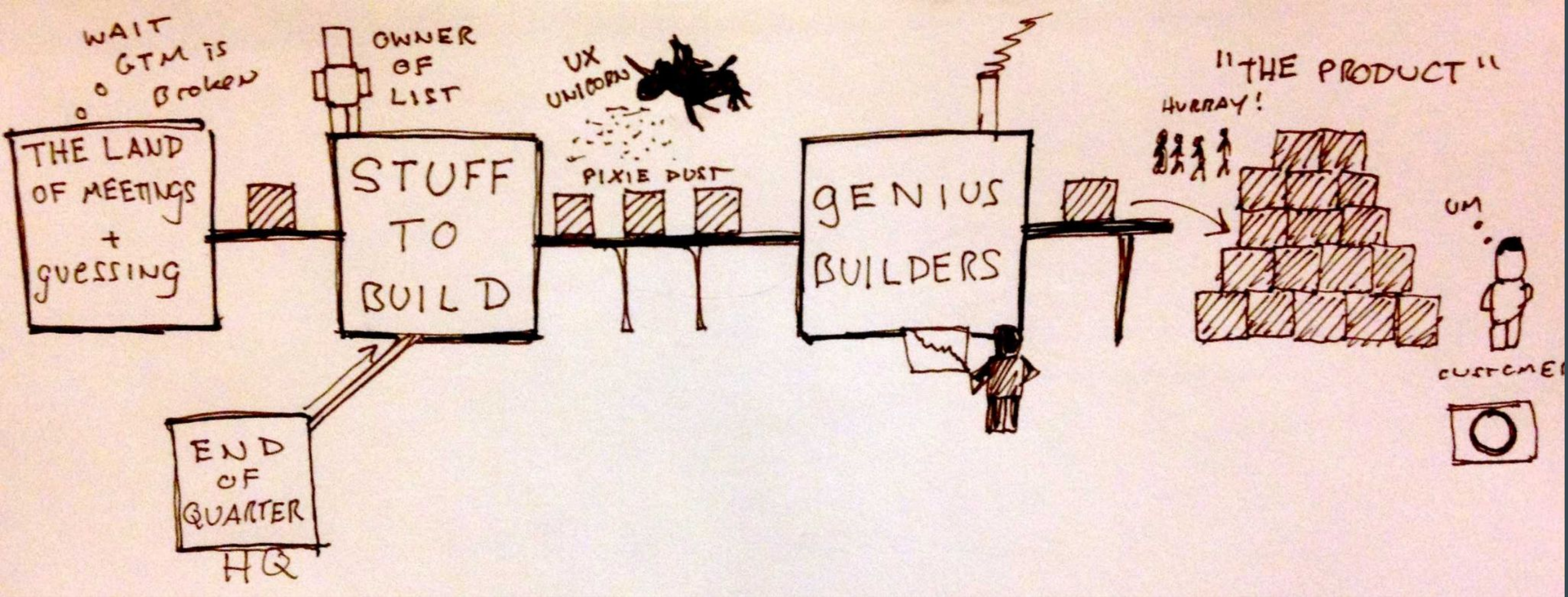
Get really good at vertical slicing

Start here: https://www.reddit.com/r/agile/comments/1btxpzd

Erase all dependencies, use feature flags

Be smart about risks, avoid big-design up-front

Feature addiction



Feature addiction

Unclear goals. The sprint goal !== the work.

Feature addiction

Unclear goals. The sprint goal !== the work.

You almost certainly have a product management problem.

Feature addiction

Unclear goals. The sprint goal !== the work.

You almost certainly have a product management problem.

Read this: https://kellanem.com/notes/faqs-from-coaching
on "the team isn't working hard enough"

Feature addiction

Unclear goals. The sprint goal !== the work.

You almost certainly have a product management problem.

Read this: https://kellanem.com/notes/faqs-from-coaching
on "the team isn't working hard enough"

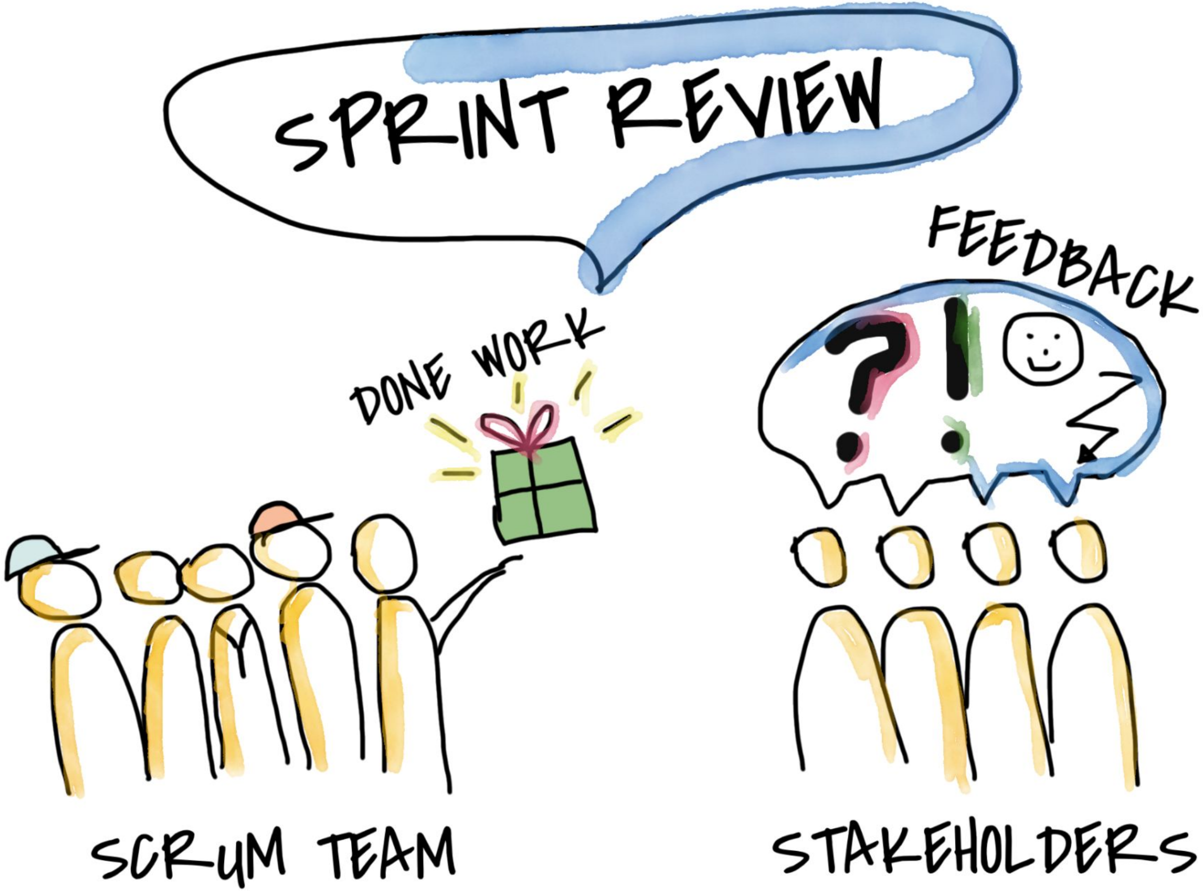
Run away.

2

Working, tested software every sprint

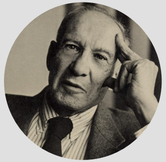
3

Know how your customers are using the product


SPRINT REVIEW
DONE WORK
FEEDBACK
?!
SCRUM TEAM
STAKEHOLDERS

There is nothing so useless
as doing with great efficiency
that which should not be done at all.

Peter Drucker

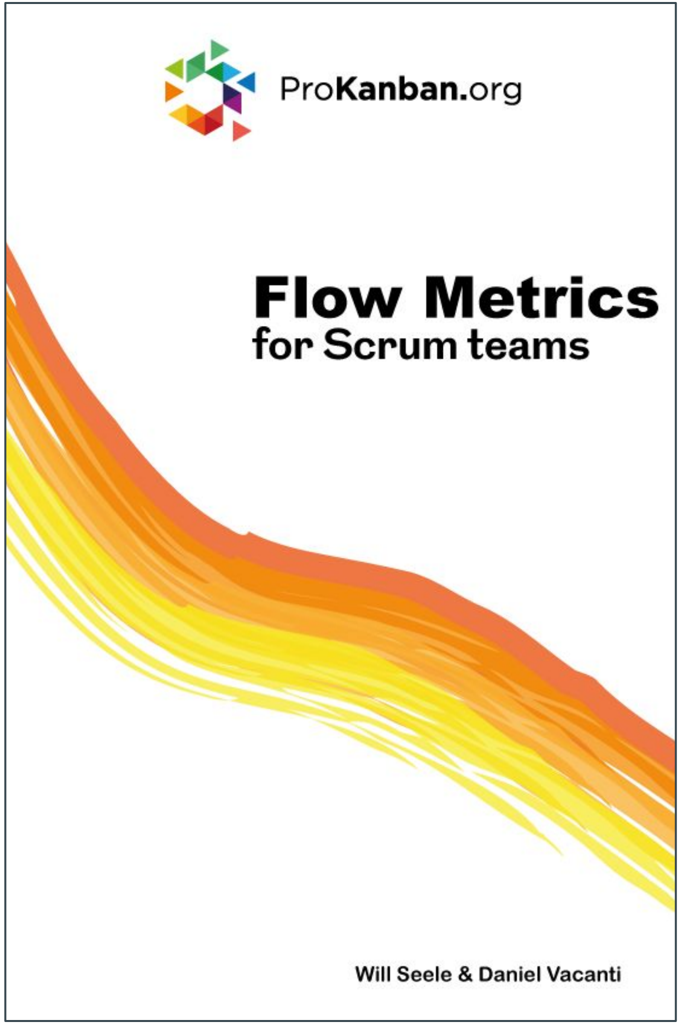


To do: tracking

feature_foo_clicks		
id	user_id	timestamp
1	1	2025-03-10T14:30:10Z
2	2	2025-03-10T14:31:23Z
3	1	2025-03-11T09:16:00Z
4	3	2025-03-12T04:10:59Z

To do: visualisation

1. Options (Backlog)
2. Discovery
3. Building
    a. Not started
    b. Coding
    c. Code Review
    d. Ready for release
4. Validating
5. Done



3

Know how your customers are using the product

4

You are not smarter than the customer

To do:

... don't do that

Respect the customer

Regular contact helps

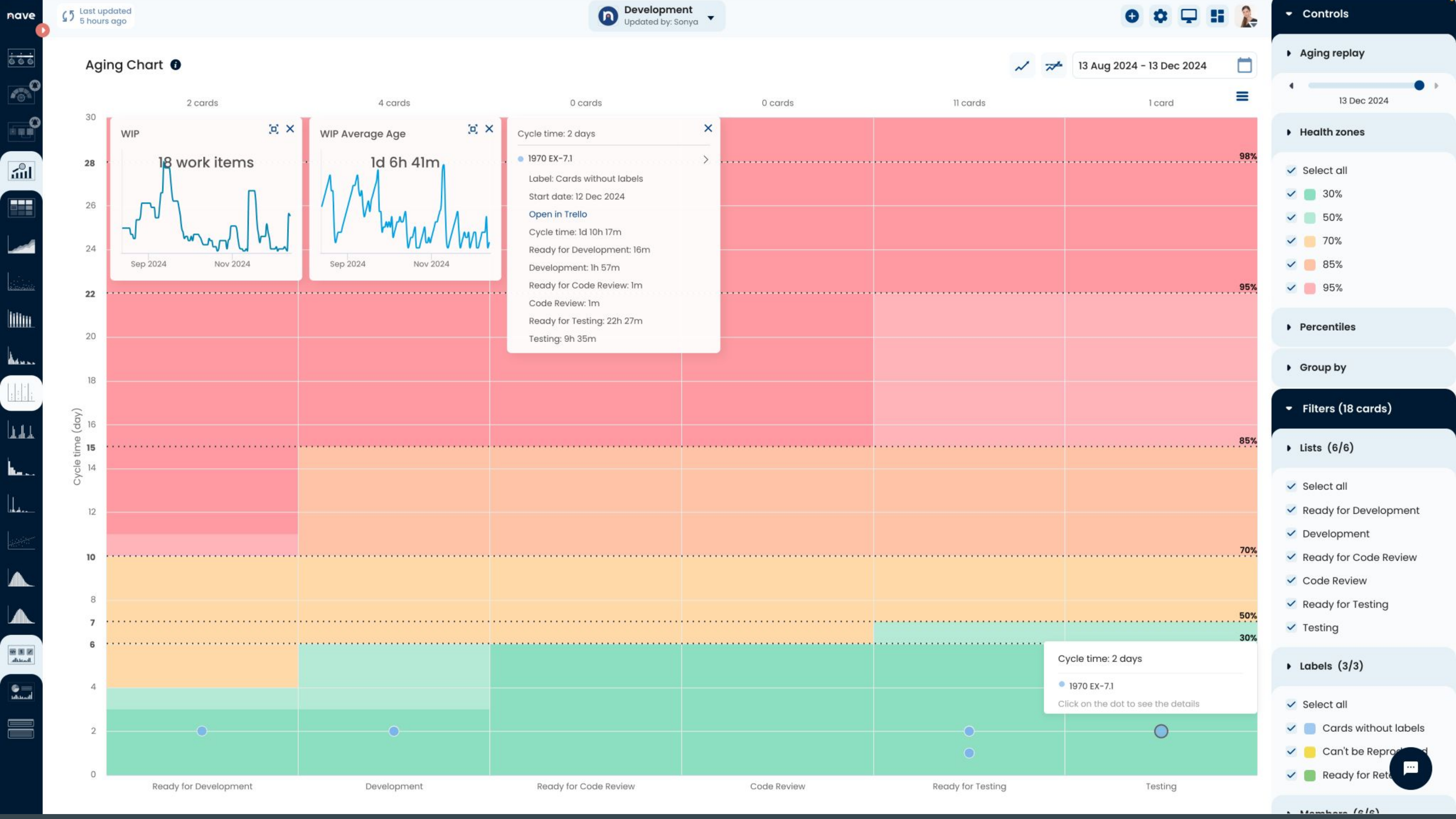
4

You are not smarter than the customer

5

Do all work
on the board

RUNNING EVENTS
REPORTS & REPORTING
CONCLUDING THE FELLOWSHIP
FOLLOW-UP AFTER THE FELLOWSHIP IS OVER
FUTURE OPPORTUNITI
ATTENDANCE DEPENDED ON TIME + VALUE
THE REPORT WRITING IS HARD WORK IF YOUR NOT ACADEMIC
I SPENT MORE TIME NEGOTIATING FORMAT THAN BEING CREATIVE
I NEVER KNEW IT WAS OVER NO COMPLETION DATE BETTY
THE NETWORK WAS GREAT. BEING INVOLVED CoM.
IT STARTED WHEN I LANDED ON A PLANE
TAKES 2 WEEKS FOR CoM MARKETING! MAJOR PAIN
COLLATING ALL THE INFORMATION AT THE END WAS HARD DISTILLING FRAMING
GOT HELP FROM OTHERS TO "GOVERNMENT IFY" REPORT
I DIDN'T MAKE GREAT CONNECTIONS IN CoM ITSELF
TICKETING IS AN ISSUE FREE MEANS PEOPLE DON'T COME
I WOULD HAVE MADE IT PRACTICAL IF IT WAS FOR EVERYONE
THE EXPOSE ADDS VALUE TO MY BRAND
Flexibility of Proj was beneficial
Professional



Cycle time: 2 days

1970 EX-7.1

Label: Cards without labels

Start date: 12 Dec 2024

Open in Trello

Cycle time: 1d 10h 17m

Ready for Development: 16m

Development: 1h 57m

Ready for Code Review: 1m

Code Review: 1m

Ready for Testing: 22h 27m

Testing: 9h 35m

Cycle time: 2 days

1970 EX-7.1

Click on the dot to see the details

To do:

Create item: [title] + [assigned you] + [in progres]

Consider skipping ticket when:

- doing it right now
- takes < 10 minutes (and you're 99% certain)
- is a repeating action (automate it!)

Bias to having a single board per team.

Items never go back: stuck is preferable.

5

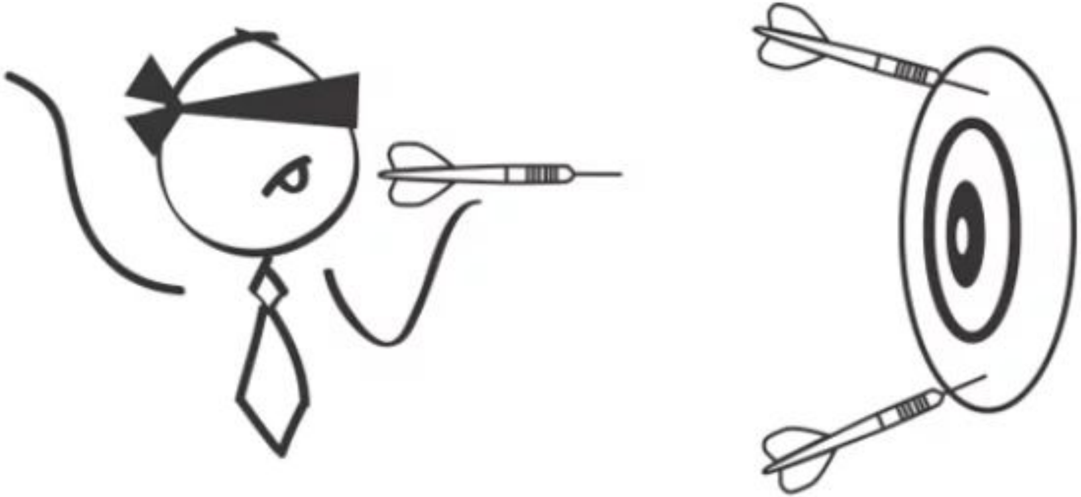
Do all work
on the board

6

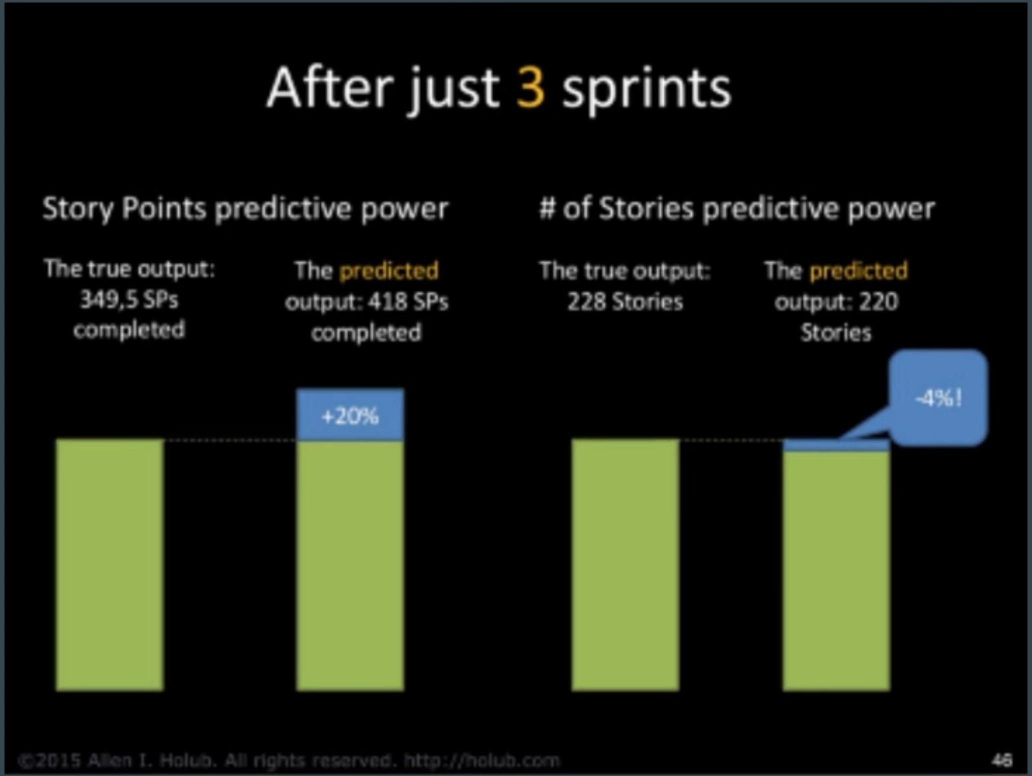
All estimates are bullshit

Estimation

The fine art of guessing



#NoEstimates



After just 3 sprints

Story Points predictive power

The true output: 349,5 SPs completed

The predicted output: 418 SPs completed

# of Stories predictive power

The true output: 228 Stories

The predicted output: 220 Stories

#NoEstimates (Allen Holub)
https://www.youtube.com/watch?v=QVBlnCTu9Ms

To do:

Good:

same-sizing everything: "1 story point" and "too big"

https://mdalmijn.com/p/roman-estimation-a-simple-easy-and

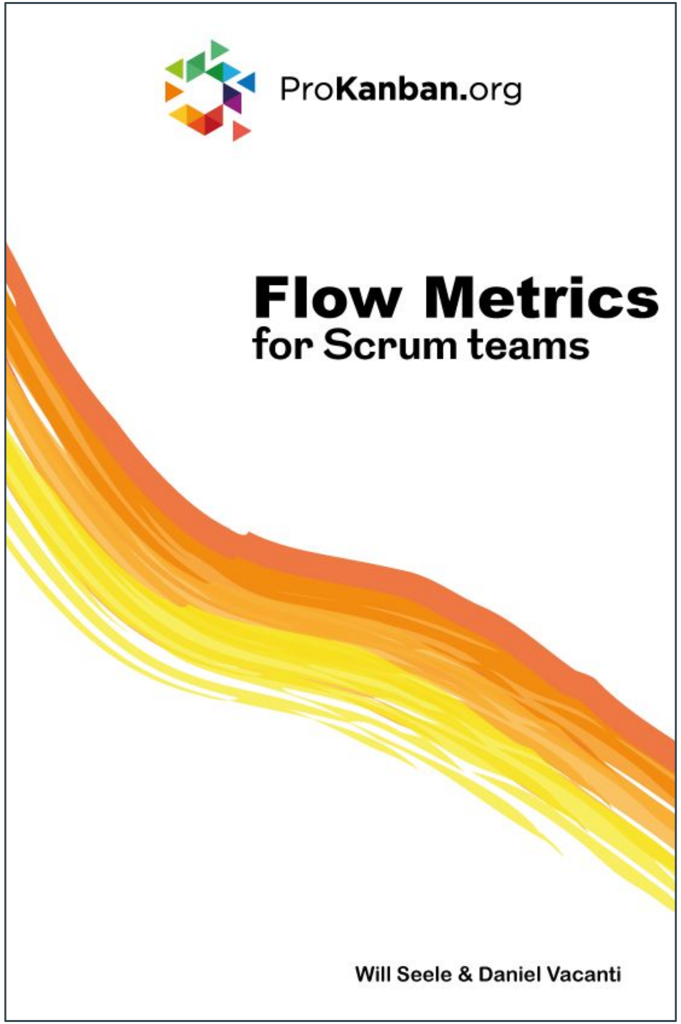
Better:

use data

PBI	Started	Finished
1	2025-03-01	2025-03-10
2	2025-02-16	2025-03-18
3	2025-03-17	
4		

Measure

- WIP
- Cycle time
- Work item age
- Throughput



Monte Carlo simulation

Record throughput per day:

0 7 2 6 6 3 7 2 9 1 13 0 0 2

Monte Carlo simulation

Record throughput per day:

0 7 2 6 6 3 7 2 9 1 13 0 0 2

Sample next 7 days:

2 0 2 7 0 3 2

Monte Carlo simulation

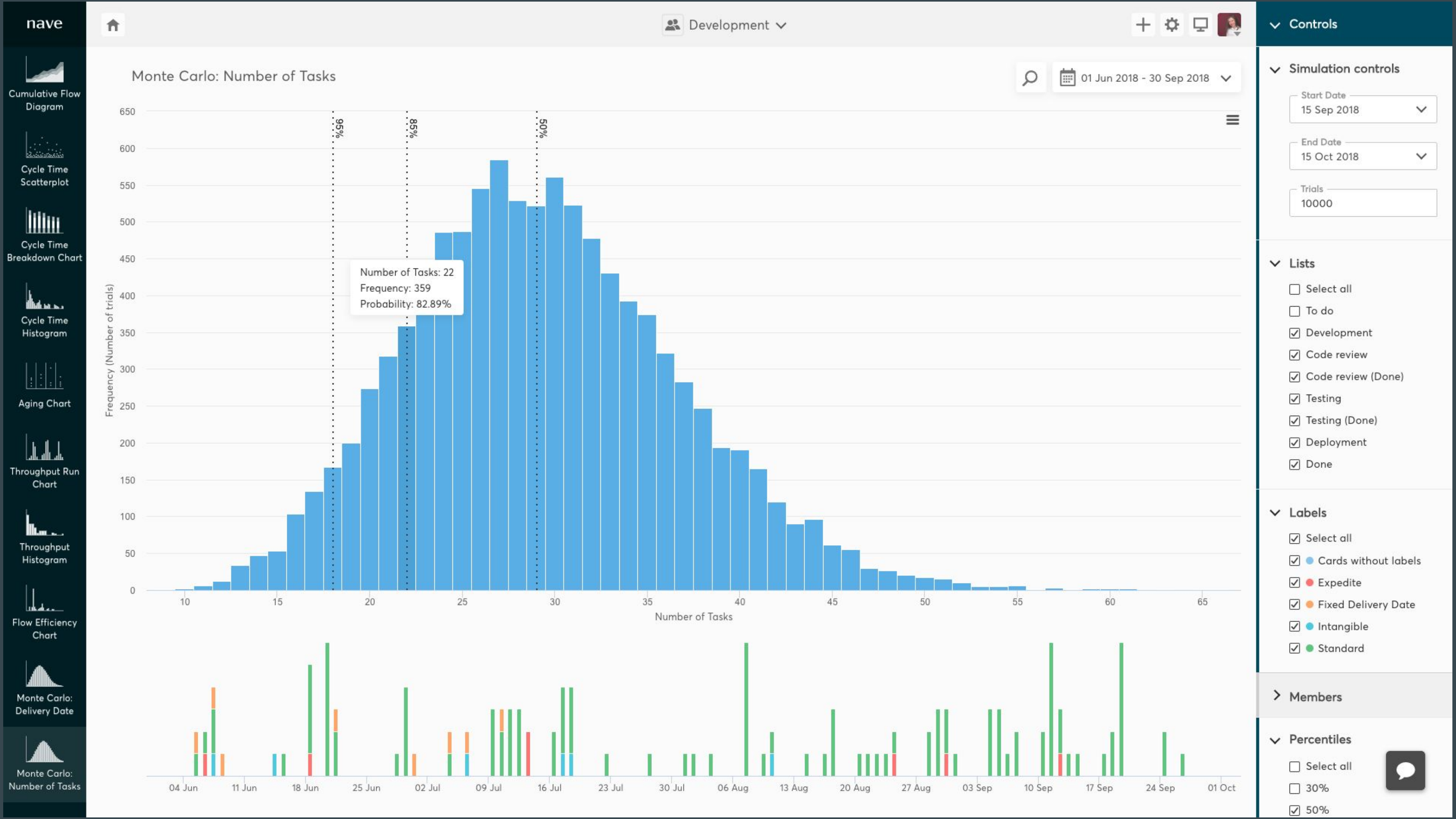
Record throughput per day:

0 7 2 6 6 3 7 2 9 1 13 0 0 2

Sample next 7 days:

2 0 2 7 0 3 2 = 16

Next week, we'll finish 16 stories.



Monte Carlo: Number of Tasks

01 Jun 2018 - 30 Sep 2018

Number of Tasks: 22
Frequency: 359
Probability: 82.89%

Controls

Simulation controls

Start Date: 15 Sep 2018

End Date: 15 Oct 2018

Trials: 10000

Lists

☐ Select all
☐ To do
☑ Development
☑ Code review
☑ Code review (Done)
☑ Testing
☑ Testing (Done)
☑ Deployment
☑ Done

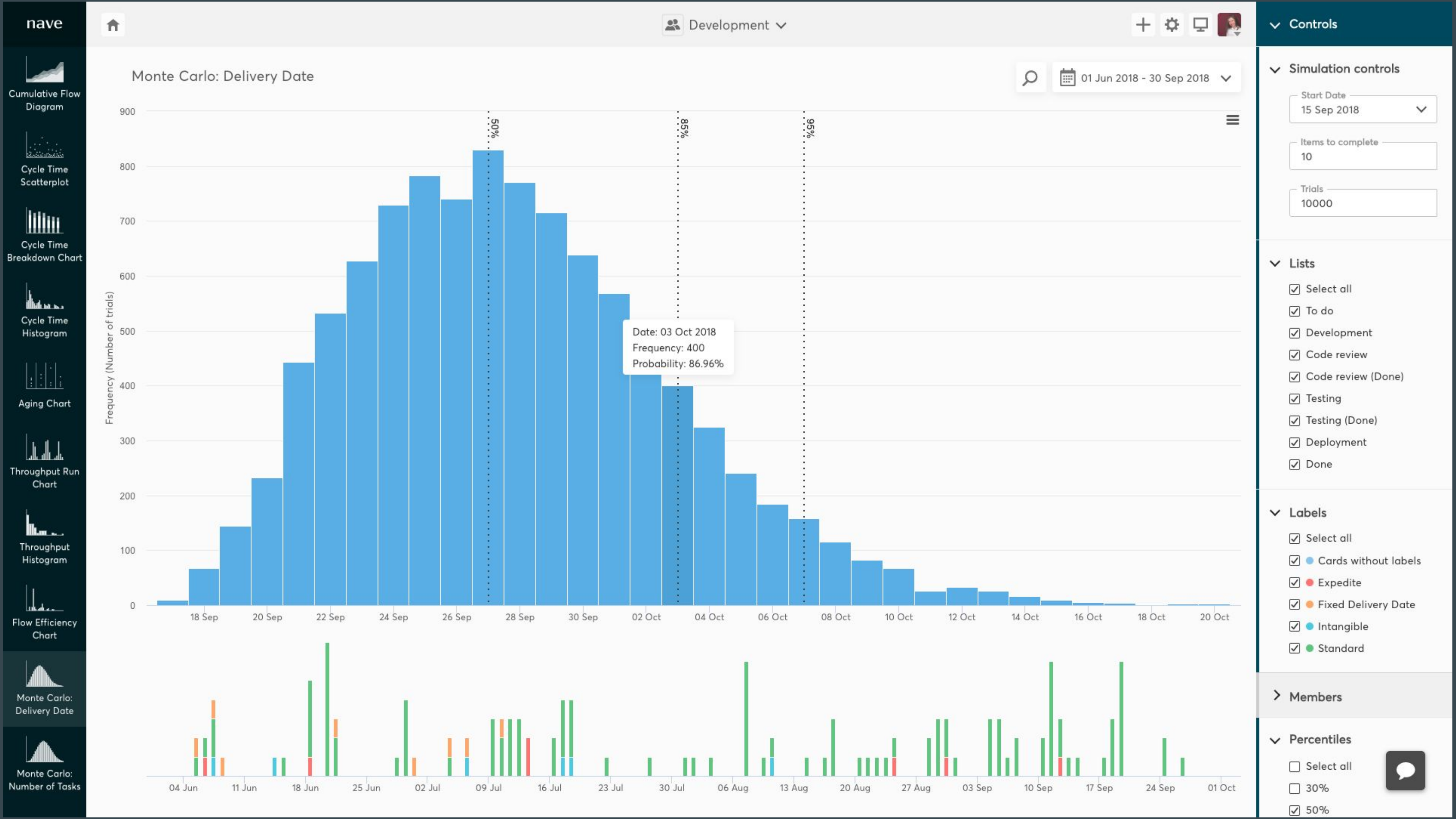
Labels

☑ Select all
☑ ● Cards without labels
☑ ● Expedite
☑ ● Fixed Delivery Date
☑ ● Intangible
☑ ● Standard

Members

Percentiles

☐ Select all
☐ 30%
☑ 50%



Monte Carlo: Delivery Date

01 Jun 2018 - 30 Sep 2018

Date: 03 Oct 2018
Frequency: 400
Probability: 86.96%

Simulation controls

Start Date: 15 Sep 2018

Items to complete: 10

Trials: 10000

Lists

- ☑ Select all
- ☑ To do
- ☑ Development
- ☑ Code review
- ☑ Code review (Done)
- ☑ Testing
- ☑ Testing (Done)
- ☑ Deployment
- ☑ Done

Labels

- ☑ Select all
- ☑ ● Cards without labels
- ☑ ● Expedite
- ☑ ● Fixed Delivery Date
- ☑ ● Intangible
- ☑ ● Standard

Members

Percentiles

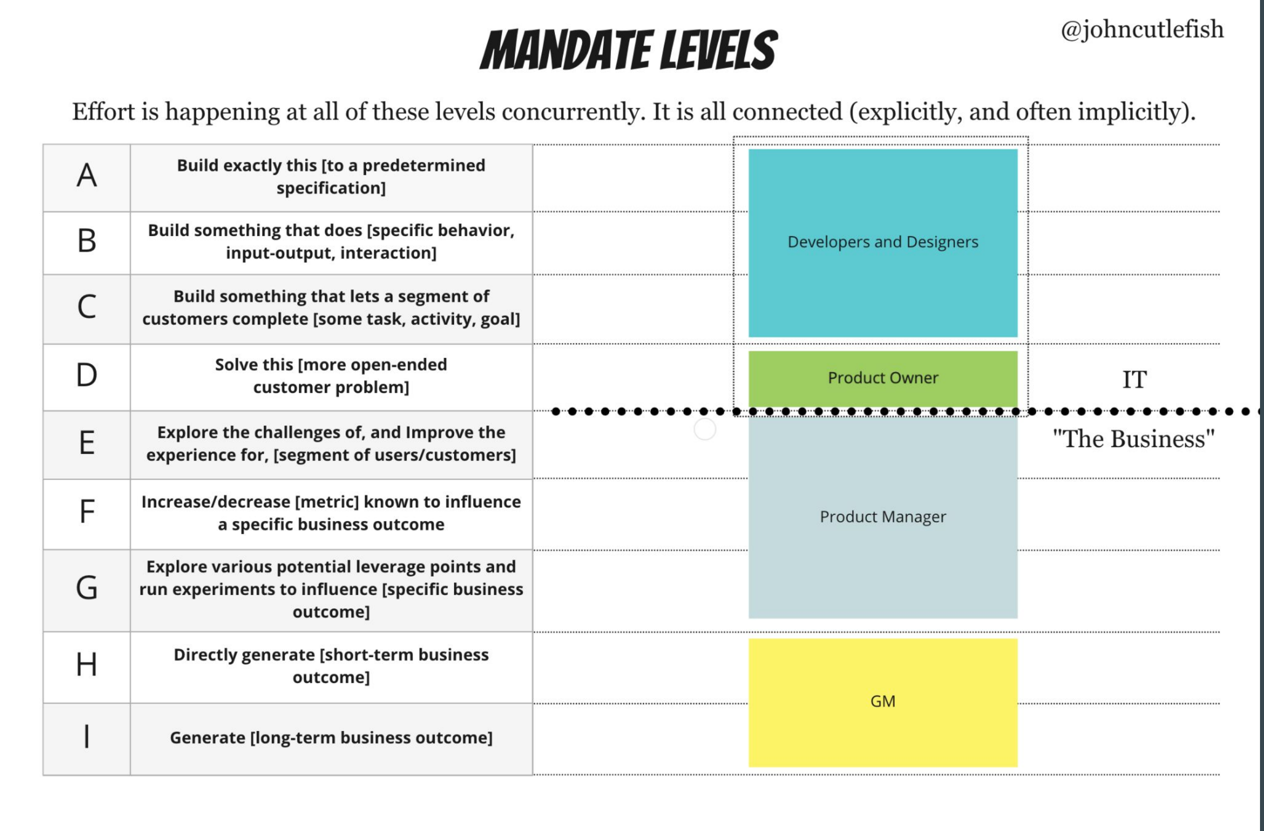
- □ Select all
- □ 30%
- ☑ 50%

6

All estimates are bullshit

7

Focus on solving business problems



MANDATE LEVELS

@johncutlefish

Effort is happening at all of these levels concurrently. It is all connected (explicitly, and often implicitly).

A	**Build exactly this [to a predetermined specification]**
B	**Build something that does [specific behavior, input-output, interaction]**
C	**Build something that lets a segment of customers complete [some task, activity, goal]**
D	**Solve this [more open-ended customer problem]**
E	**Explore the challenges of, and Improve the experience for, [segment of users/customers]**
F	**Increase/decrease [metric] known to influence a specific business outcome**
G	**Explore various potential leverage points and run experiments to influence [specific business outcome]**
H	**Directly generate [short-term business outcome]**
I	**Generate [long-term business outcome]**

https://cutlefish.substack.com/p/tbm-2752-mandate-levels

To do

Don't start here

Working software in the customers hands

Build a prototype

Get users in the room while designing

7

Focus on solving business problems

8

Deal with technical debt

QUALITY

CONSISTENCY

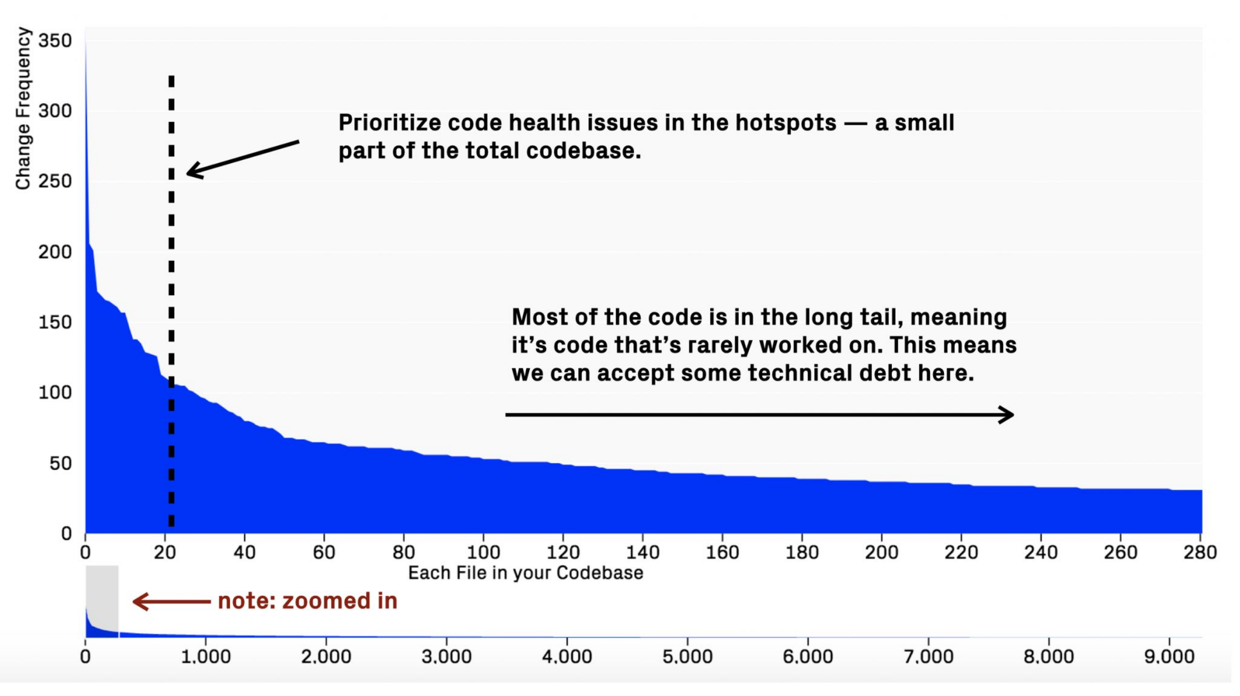
To do

Minimum:

1. error monitoring service (sentry)
2. static analysis (psalm, phpstan)
3. existing coding standard enforced by linter

Fix bugs

To do



Prioritize code health issues in the hotspots — a small part of the total codebase.

Most of the code is in the long tail, meaning it's code that's rarely worked on. This means we can accept some technical debt here.

note: zoomed in

https://www.getunleash.io/blog/manage-technical-debt-measure-the-impact-and-prioritize-improvements-guided-by-development-data

8

Deal with technical debt

9

Have a strong Definition of Done

9

Have a strong Definition of Done

To do

Absolute

Automated

Agreed with PO

10

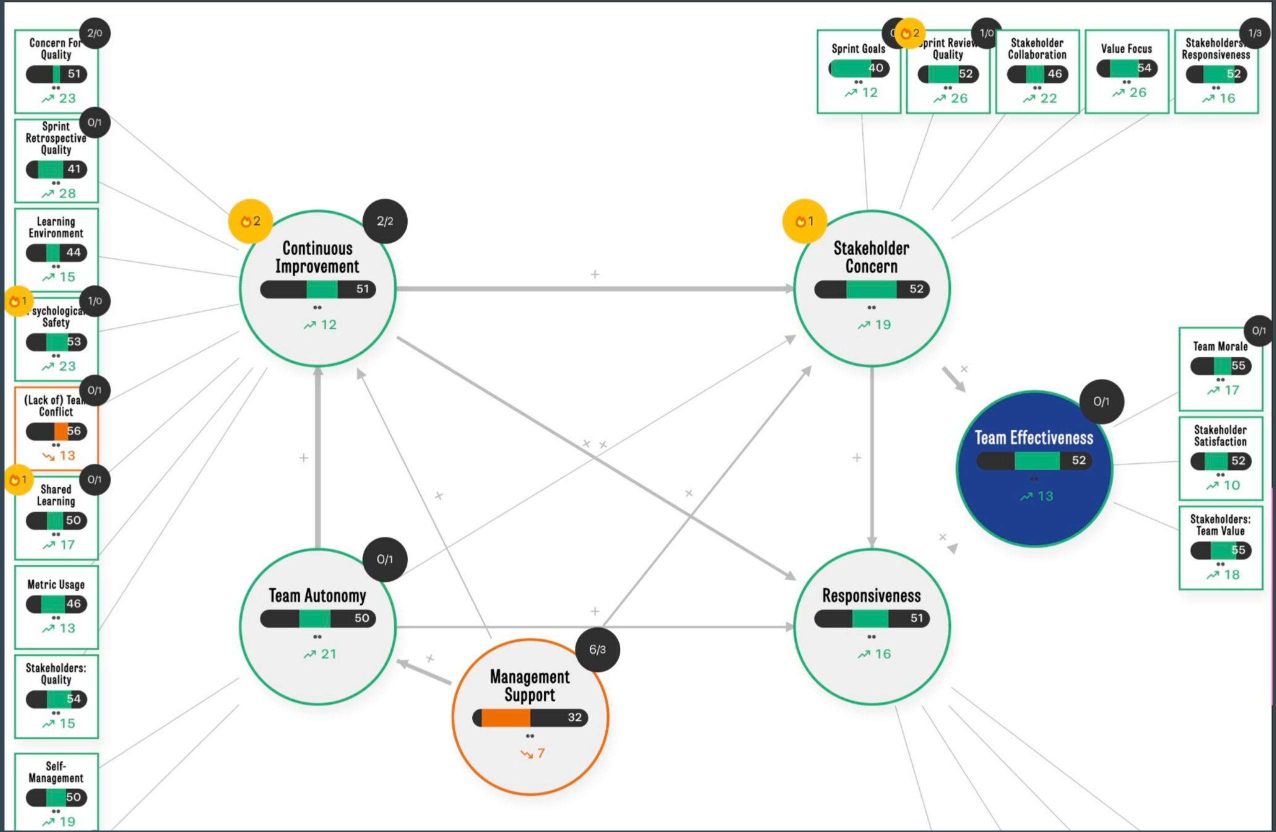
Make retrospectives effective

THIS IS FINE.

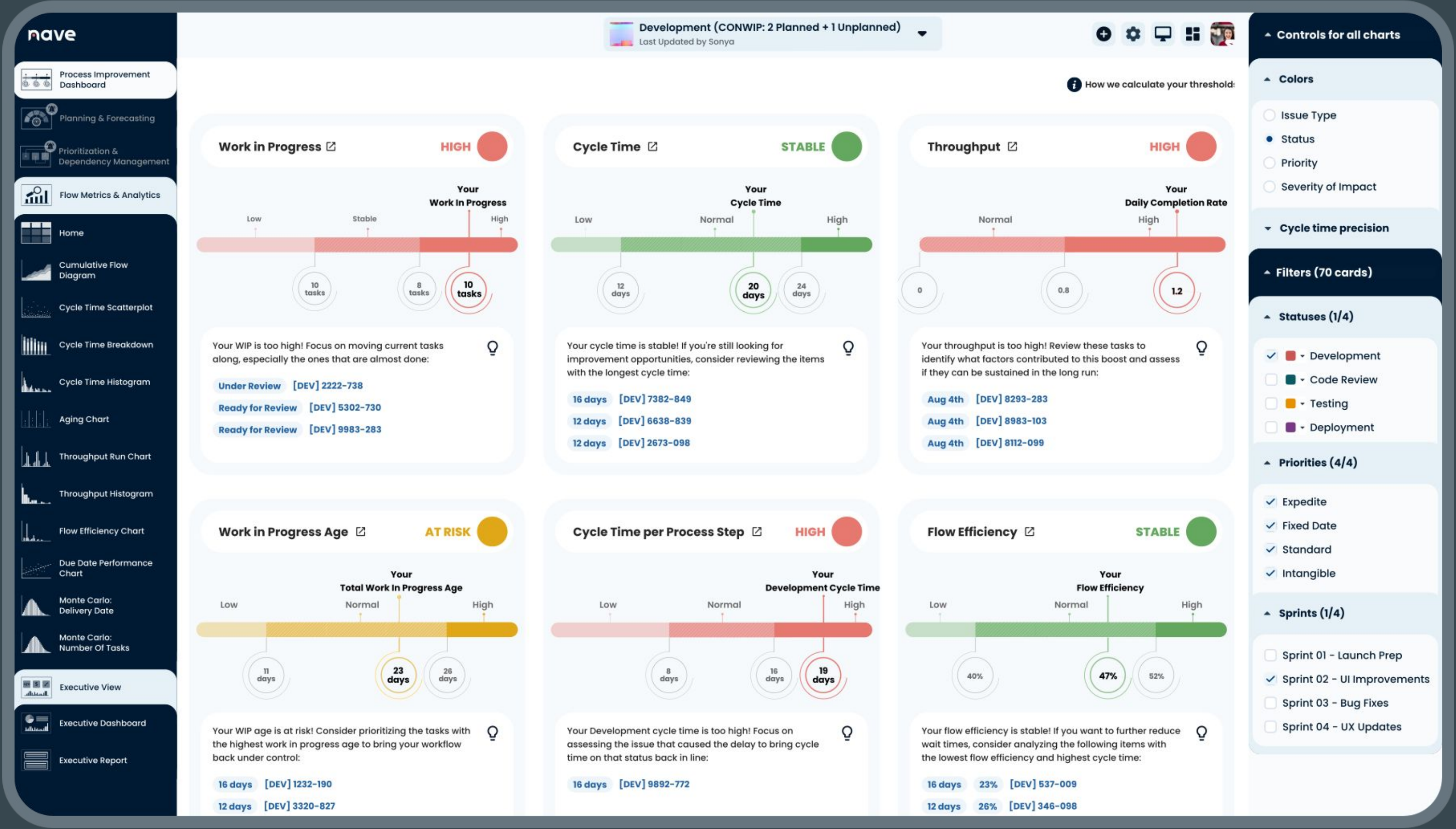
To do

Escalate what you cannot solve

Data-driven decision making



columinity.com



nave

Process Improvement Dashboard

Planning & Forecasting

Prioritization & Dependency Management

Flow Metrics & Analytics

Home

Cumulative Flow Diagram

Cycle Time Scatterplot

Cycle Time Breakdown

Cycle Time Histogram

Aging Chart

Throughput Run Chart

Throughput Histogram

Flow Efficiency Chart

Due Date Performance Chart

Monte Carlo: Delivery Date

Monte Carlo: Number Of Tasks

Executive View

Executive Dashboard

Executive Report

Development (CONWIP: 2 Planned + 1 Unplanned)
Last Updated by Sonya

How we calculate your thresholds

Work in Progress HIGH

Your Work In Progress

Low · Stable · High

10 tasks · 8 tasks · 10 tasks

Your WIP is too high! Focus on moving current tasks along, especially the ones that are almost done:

Under Review	[DEV] 2222-738
Ready for Review	[DEV] 5302-730
Ready for Review	[DEV] 9983-283

Cycle Time STABLE

Your Cycle Time

Low · Normal · High

12 days · 20 days · 24 days

Your cycle time is stable! If you're still looking for improvement opportunities, consider reviewing the items with the longest cycle time:

16 days	[DEV] 7382-849
12 days	[DEV] 6638-839
12 days	[DEV] 2673-098

Throughput HIGH

Your Daily Completion Rate

Normal · High

0 · 0.8 · 1.2

Your throughput is too high! Review these tasks to identify what factors contributed to this boost and assess if they can be sustained in the long run:

Aug 4th	[DEV] 8293-283
Aug 4th	[DEV] 8983-103
Aug 4th	[DEV] 8112-099

Work in Progress Age AT RISK

Your Total Work In Progress Age

Low · Normal · High

11 days · 23 days · 26 days

Your WIP age is at risk! Consider prioritizing the tasks with the highest work in progress age to bring your workflow back under control:

16 days	[DEV] 1232-190
12 days	[DEV] 3320-827

Cycle Time per Process Step HIGH

Your Development Cycle Time

Low · Normal · High

8 days · 16 days · 19 days

Your Development cycle time is too high! Focus on assessing the issue that caused the delay to bring cycle time on that status back in line:

16 days	[DEV] 9892-772

Flow Efficiency STABLE

Your Flow Efficiency

Low · Normal · High

40% · 47% · 52%

Your flow efficiency is stable! If you want to further reduce wait times, consider analyzing the following items with the lowest flow efficiency and highest cycle time:

16 days	23%	[DEV] 537-009
12 days	26%	[DEV] 346-098

Controls for all charts

Colors

○ Issue Type
● Status
○ Priority
○ Severity of Impact

Cycle time precision

Filters (70 cards)

Statuses (1/4)

☑ 🟥 Development
□ 🟩 Code Review
□ 🟧 Testing
□ 🟪 Deployment

Priorities (4/4)

☑ Expedite
☑ Fixed Date
☑ Standard
☑ Intangible

Sprints (1/4)

□ Sprint 01 - Launch Prep
☑ Sprint 02 - UI Improvements
□ Sprint 03 - Bug Fixes
□ Sprint 04 - UX Updates

getnave.com

10

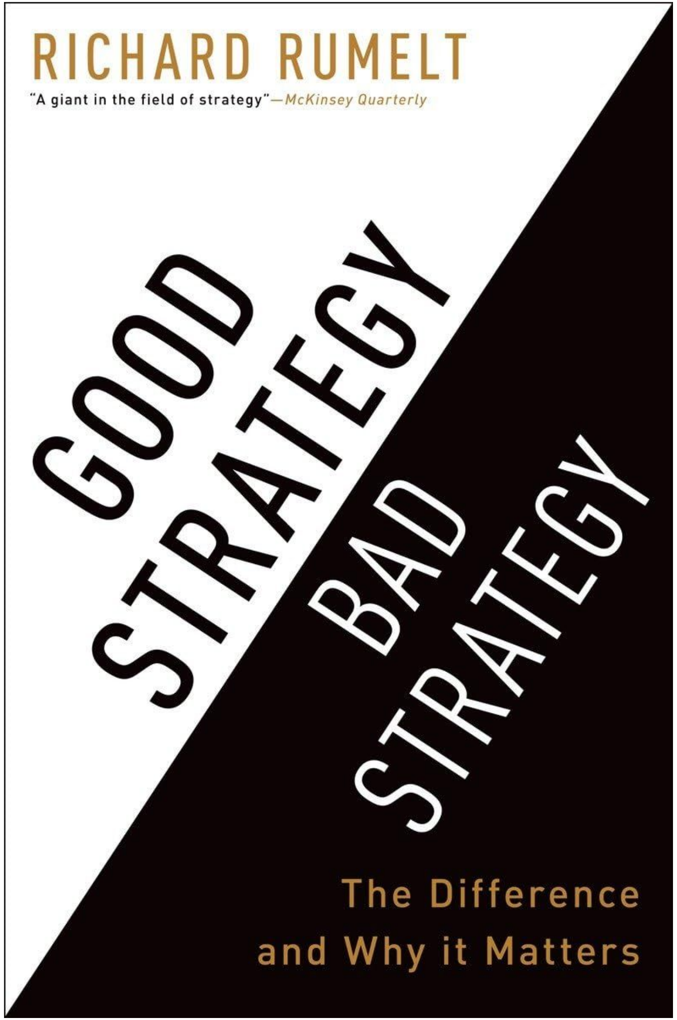
Make retrospectives effective

To do

Great Scrum Team - To do:

1. Experiment beyond Scrum, not before Scrum
2. Working tested software, every sprint.
3. Know how your customers are using the product
4. You are not smarter than your customer
5. Do all work on the board
6. All estimation is bullshit
7. Focus on solving business problems
8. Deal with technical debt
9. Having a strong Definition of Done
10. Make retrospectives effective

How to get started



RICHARD RUMELT

"A giant in the field of strategy"—*McKinsey Quarterly*

GOOD STRATEGY BAD STRATEGY

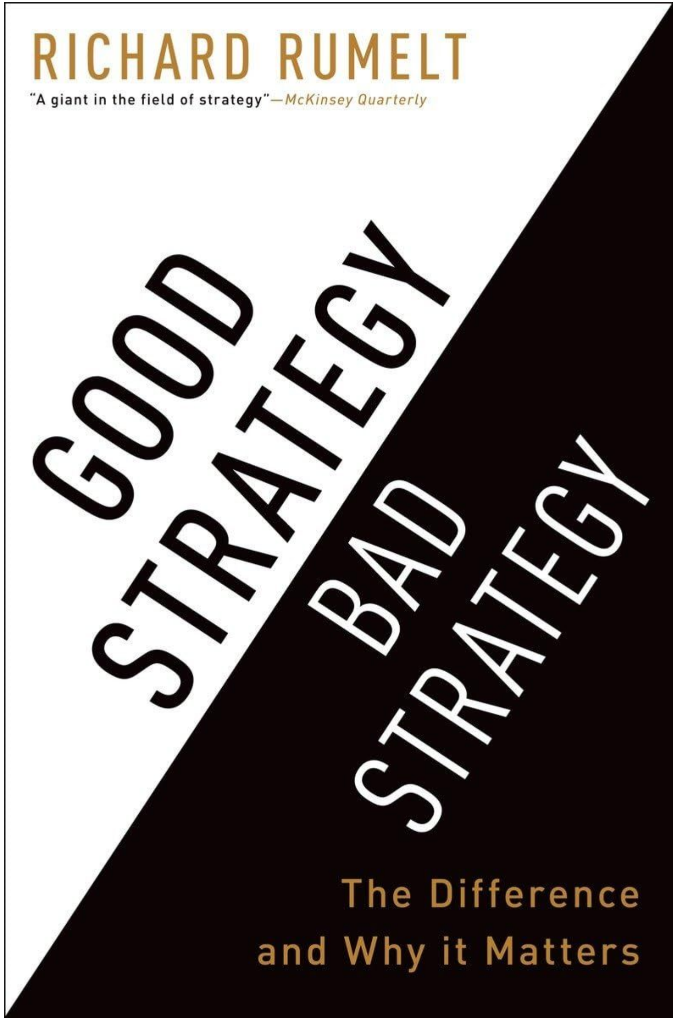
The Difference and Why it Matters

The kernel of a strategy contains three elements:

a diagnosis,

a guiding policy,

and coherent action.



RICHARD RUMELT

"A giant in the field of strategy"—McKinsey Quarterly

GOOD STRATEGY BAD STRATEGY

The Difference and Why it Matters

That's all

Slides
Feedback
Assessment

Don't forget to like, subscribe and hit the bell button.

Slides
Feedback
Assessment