Ron Quartel

Thinking Big While Working Small:
With Discovery Mapping

Agenda
(After introduction of Speakers and Topic)

Why – Why do we need new ways of visualizing work?
What problems are we trying to solve?

Complex vs Complicated
Cynefin Model

What – what is the solution?
How can we represent complexity?

What - Thinking Big: Product Mapping

Thinking Big – A tool for Product Managers to visualize the big picture so that the entire tribe has a context of where we are going and the context of the work items. Show progress, releases, and can be used for forecasting completion.

What – Working Small: Discovery Trees

Discovery Trees are mostly for devs: aid breakdown, understand work, track work, report on progress, forecast completion (optional), and maintaining context.

Real World Examples
Of Discovery Trees

Retrospective
What did you learn?

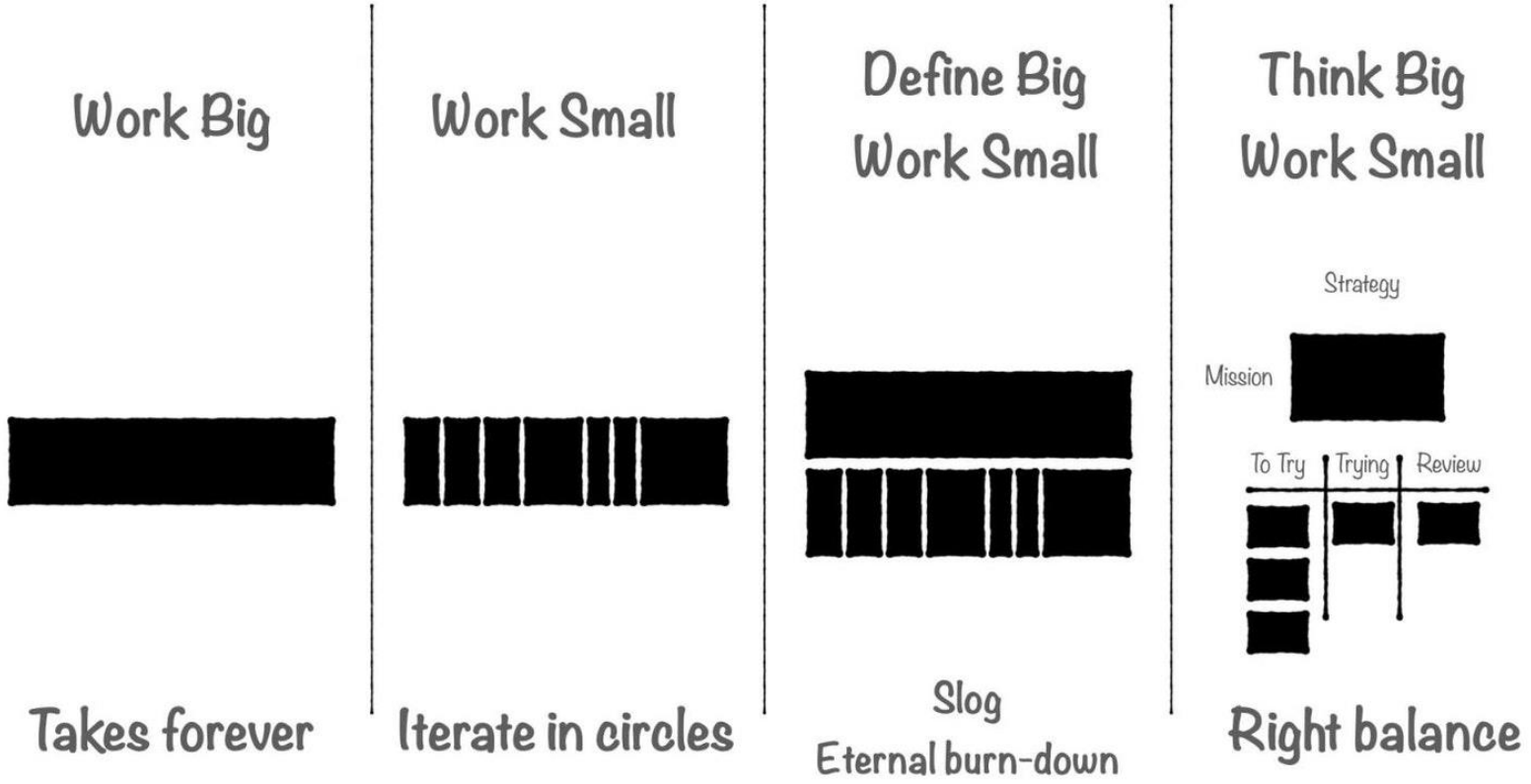
Benefits, Summary, Closing
Summary of Benefits
Tooling
Similar Concepts
Finding Out More - References
Staying in Touch - Links

Thinking Big While Working Small

Introduction to Topic

Thinking Big While Working Small

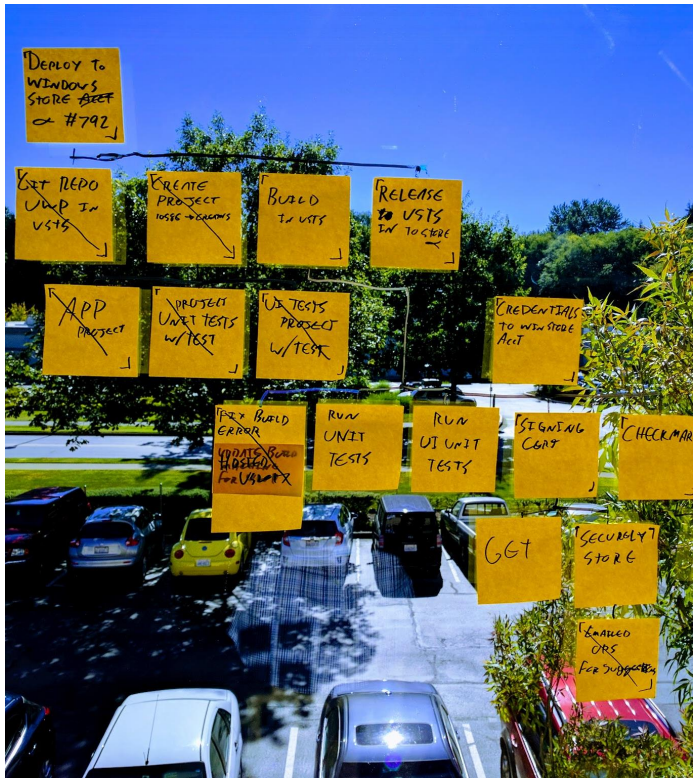
John Cutler



Discovery Mapping:
Product Mapping + Discovery Trees

Product Map

Discovery Trees



Who is Ron Quartel?

- Writing software since my teens (Apple II)
- Agilist since 2002 - eXtreme Programming (XP)
- Dev, Dev manager, consultant, entrepreneur, TPM, **agile coach, technical coach**, pre-sales engineer, trainer, presenter
- Tweeter, Blogger, Code Crafter, Pioneer, Entrepreneur
- Founder of Fluid Scaling Technology (FAST Agile)

FAST AGILE

Who is Paige Watson?

Three Truths and A Lie about Paige Watson

- When a 5000+ line BASIC file failed to render a picture of Farah Faucet, he decided to give up programming forever.
- In 2000, Paige heard about XP development and asked a co-worker to try pairing with him, only to be laughed at.
- Paige left a lucrative career in the Adult Industry to teach the world to build better software.
- Paige is one of the only two people in the world to attain the extremely hard "Certified FAST Instructor" certification.
- Paige is currently a Technical Coach, Mentor and runs the Seattle Code Crafters meetup.

Who else do we have here today?

Developers, PMs, Scrum Masters, POs, Managers, other…

Our Hope: by the end of this session…

- Product Managers have a new tool
- Developers have a new tool
- Everyone understands complex vs complicated
- Everyone has experienced Discovery Mapping in an exercise
- Every can see the power and benefits these tools can bring
- Everyone really likes Ron and Paige – and want to stay in touch with us

Lindsay Warren • 1st

Certified Scrum Master and Team Lead at MAXIMUS Canada

1h •••

My team and I did our first discovery map late last week and it was such an effective tool! It also enabled my team to self direct and plan so I could focus on the big picture and trust that if I needed to drill down, the map would explain where we were at and why. Thanks for the great tool!

Why – Why do we need new ways of visualizing work?

What problems are we trying to solve?

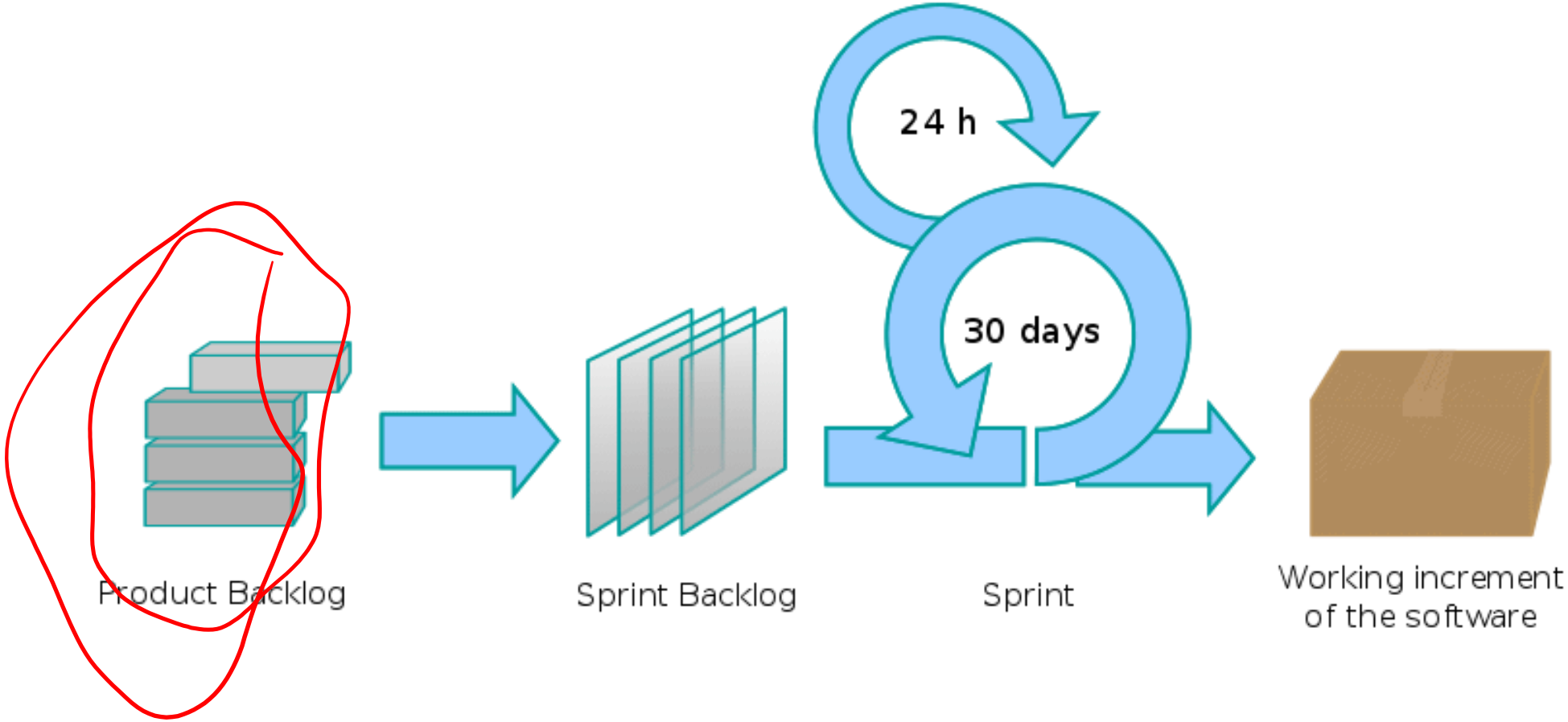
How we think of work

Trees…

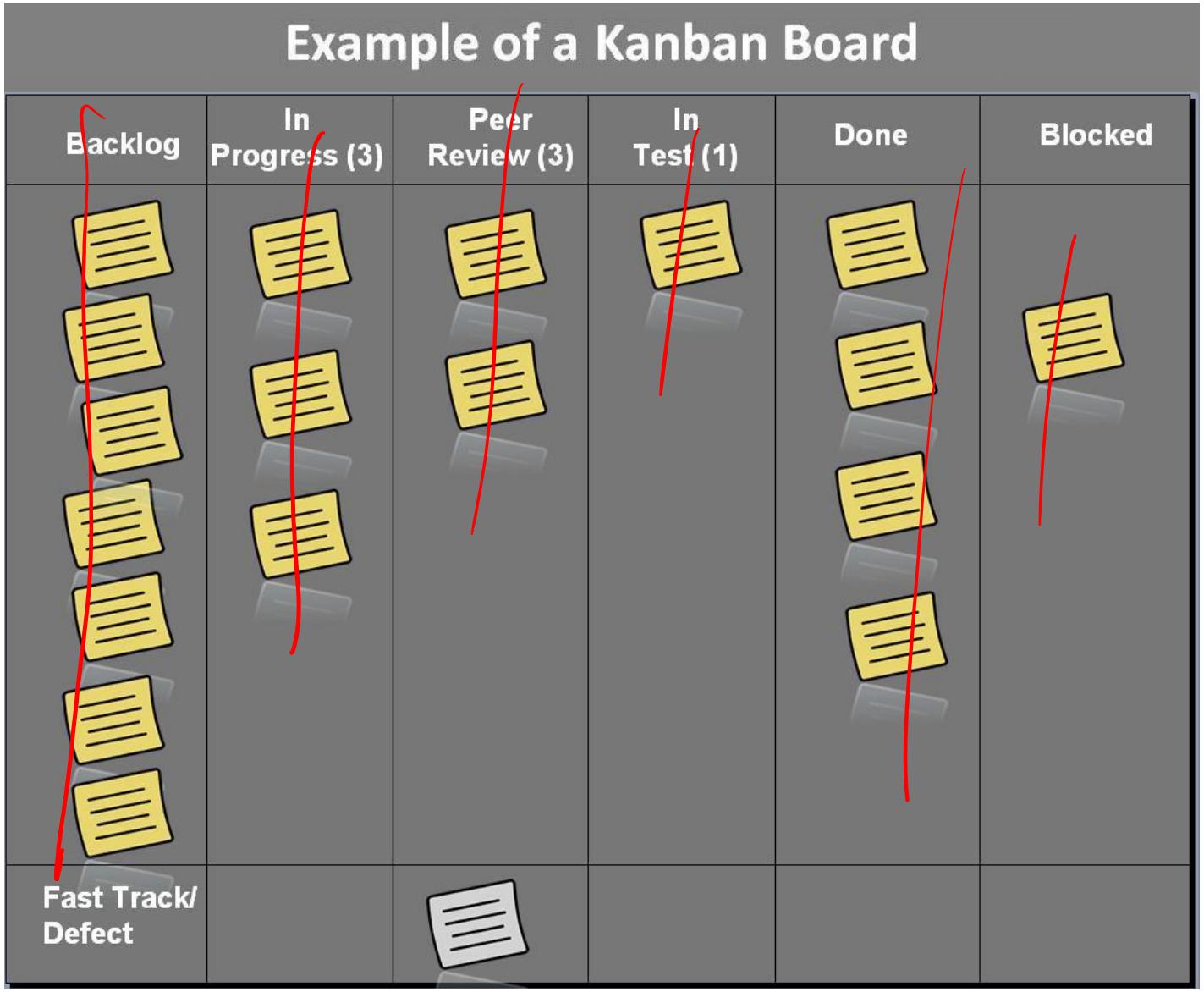
Yet whilst we think of work in tree structures (with n dimensions), we represent work in 1 – few dimensions only

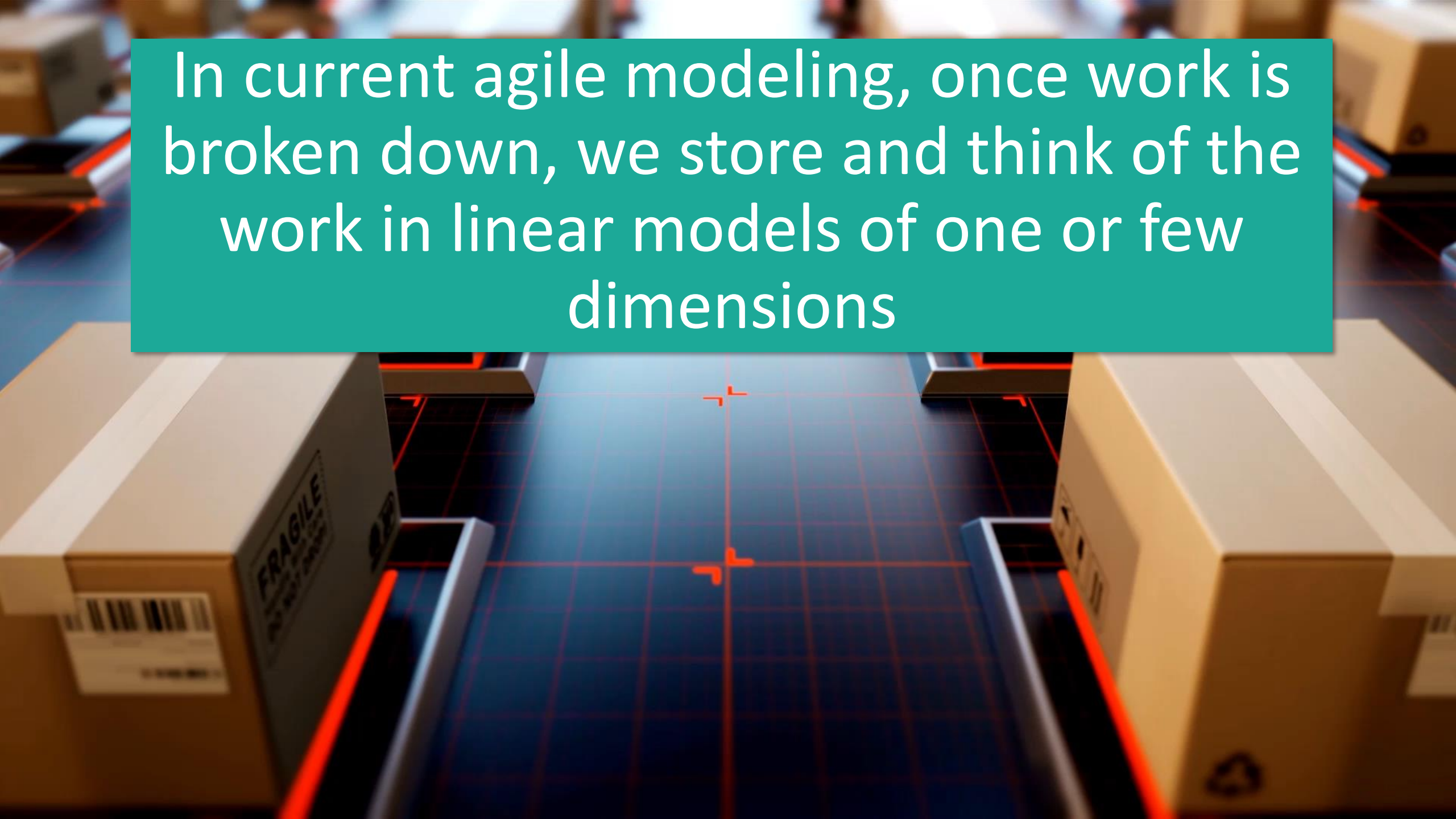
Let’s look at 1 dimensional work representation…

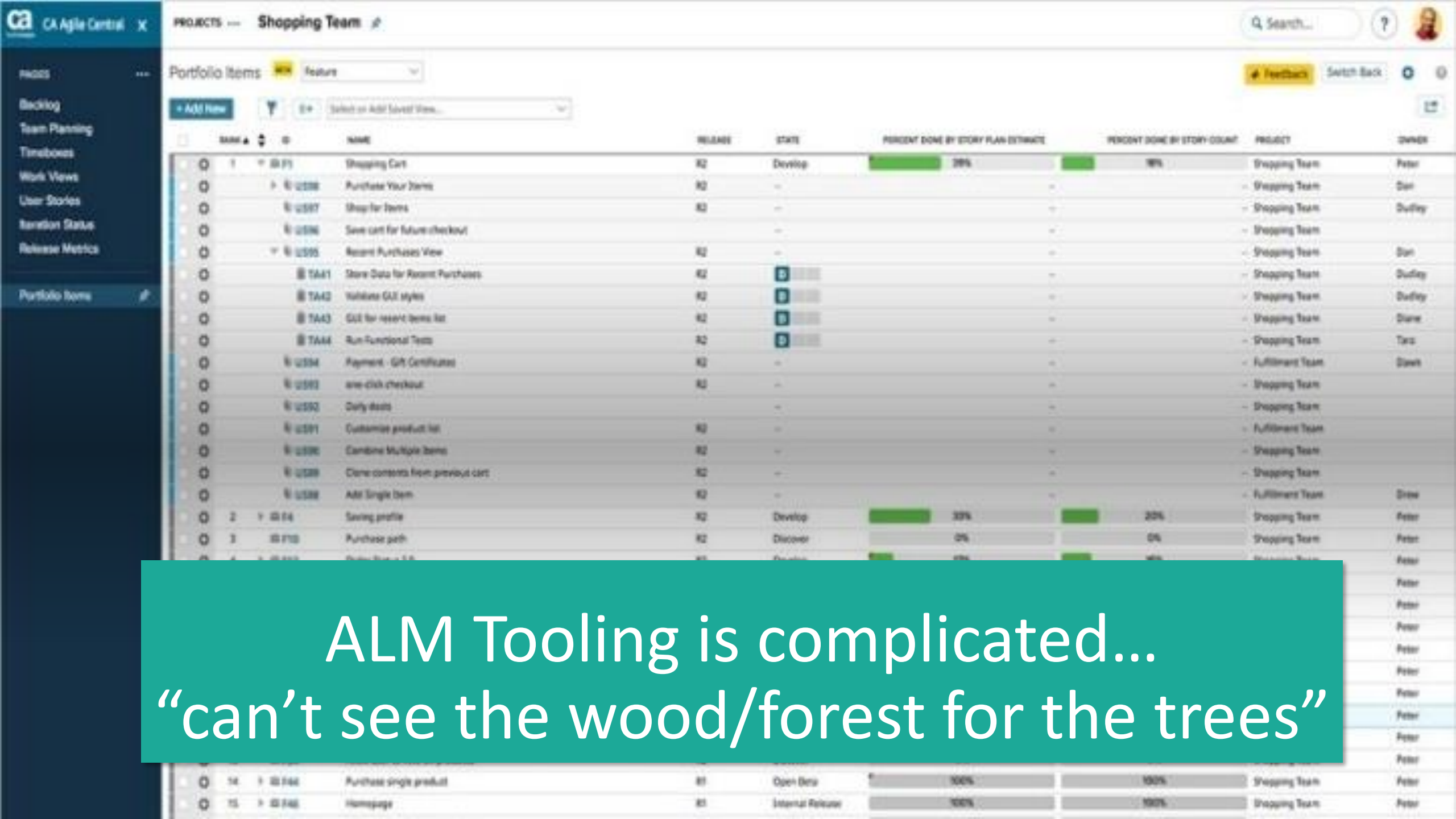
Scrum taught us to represent and think of work in one dimension – the prioritized backlog…



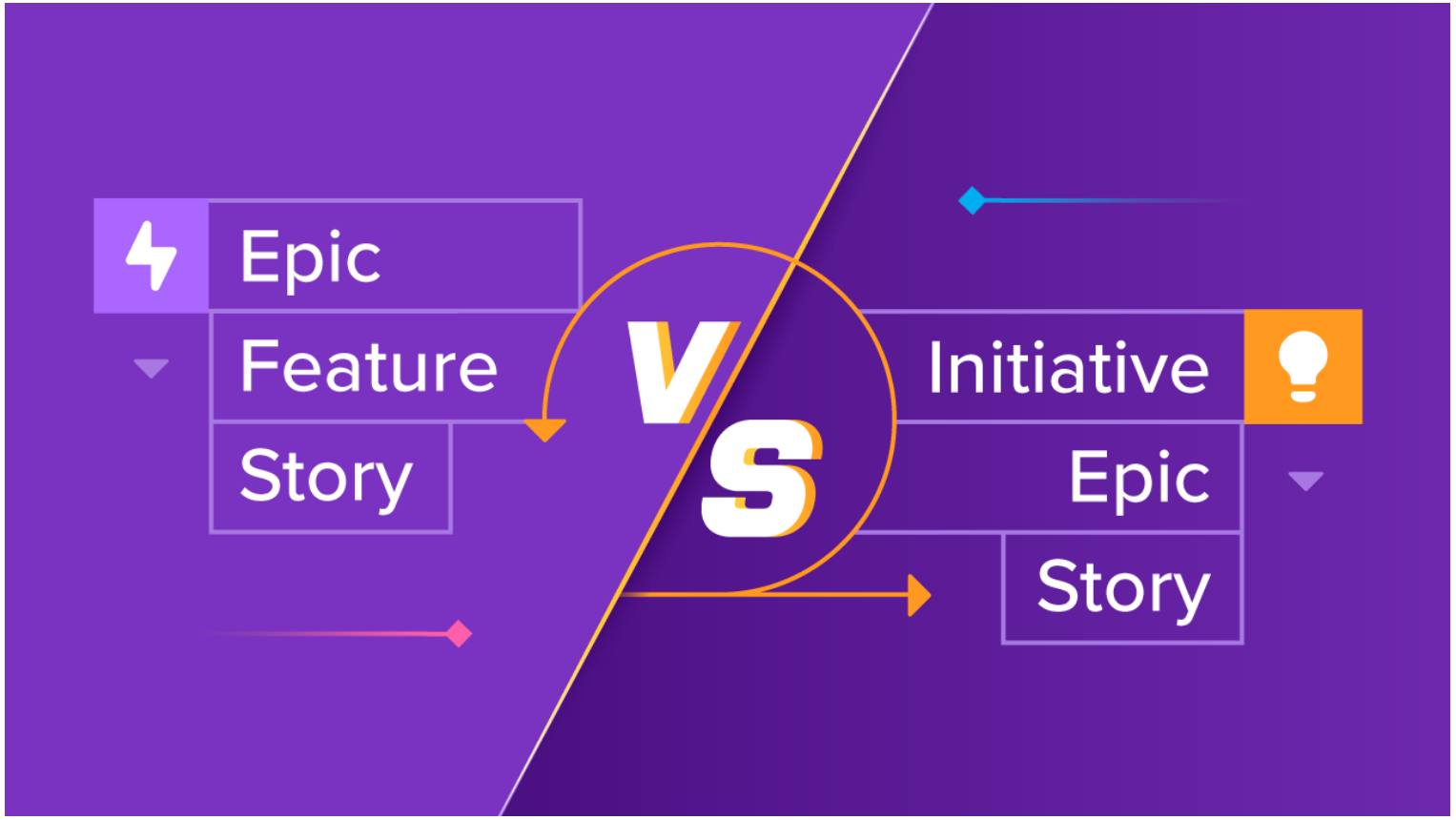
Kanban is a series of one-dimensional representations of work

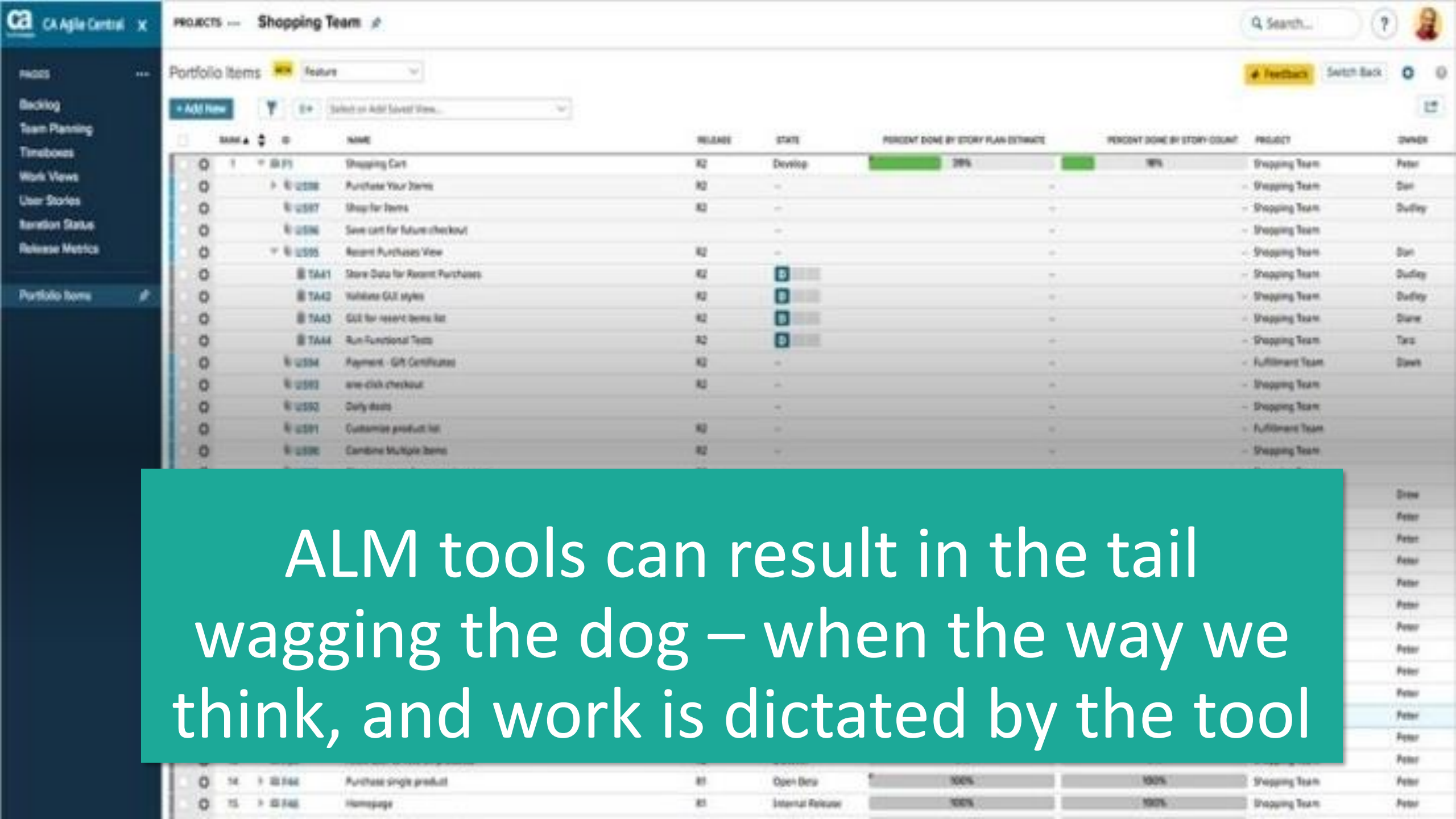



In current agile modeling, once work is broken down, we store and think of the work in linear models of one or few dimensions

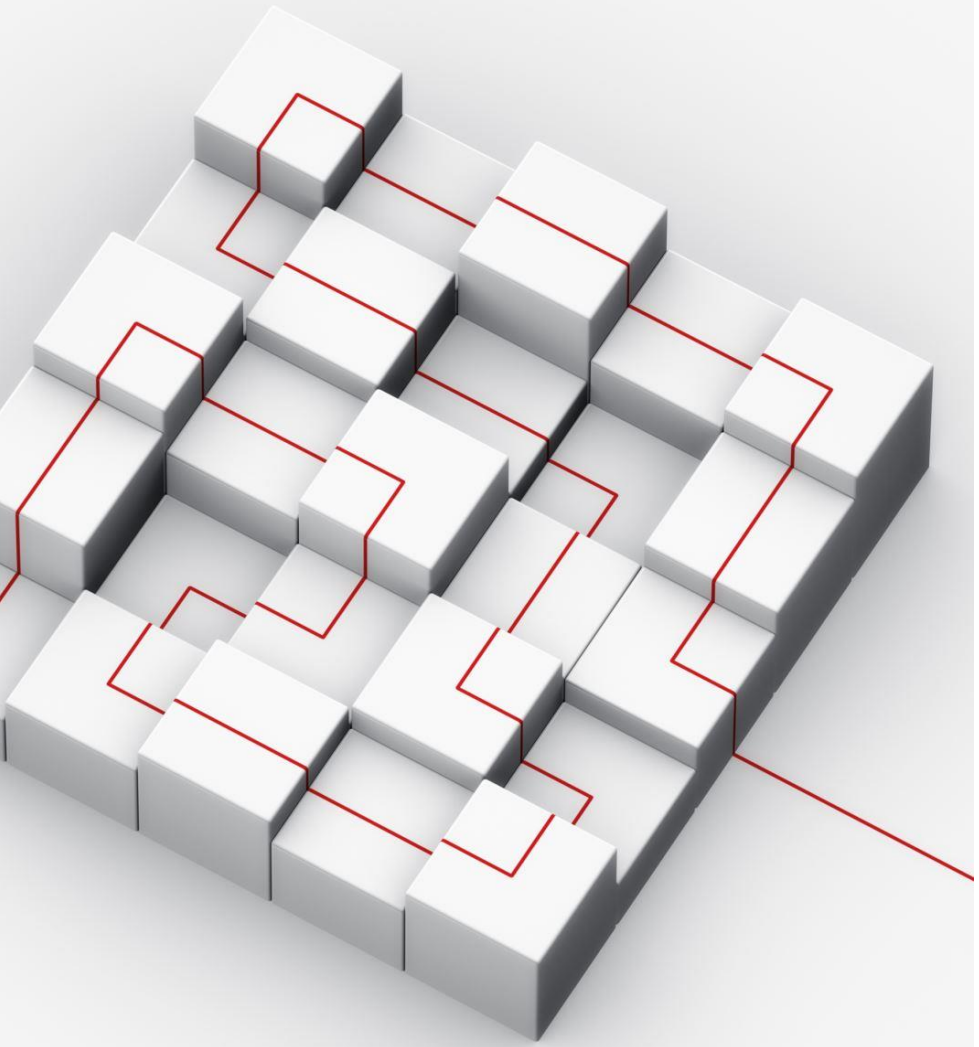

ALM Tooling is complicated...
“can’t see the wood/forest for the trees”

Confusion around agile naming conventions doesn’t help…





ALM tools can result in the tail wagging the dog – when the way we think, and work is dictated by the tool



Problems to fix:

- Scrum and Kanban represent work single dimensions
- ALM tooling represents work in a very complicated way
- ALM tooling can dictate your process
- Agile naming is confusing and inconsistent for work items

What we need is a better way to represent complexity…

But first – let’s make sure we all understand complex vs complicated…

Complex vs Complicated

Cynefin Model

Cynefin Model (Dave Snowden)

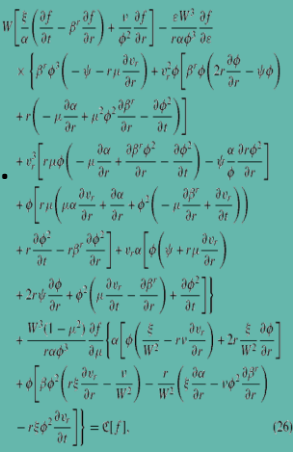
Complex

Complicated

Chaotic

Obvious

e.g.



e.g. a + b = c

Complex

- Non-causal
- Non-deterministic
- Might no be reducible
- Emergent & responsive
- Pattern is predictable but exact outcome isn’t

Examples:
Human Body, Weather

Complicated

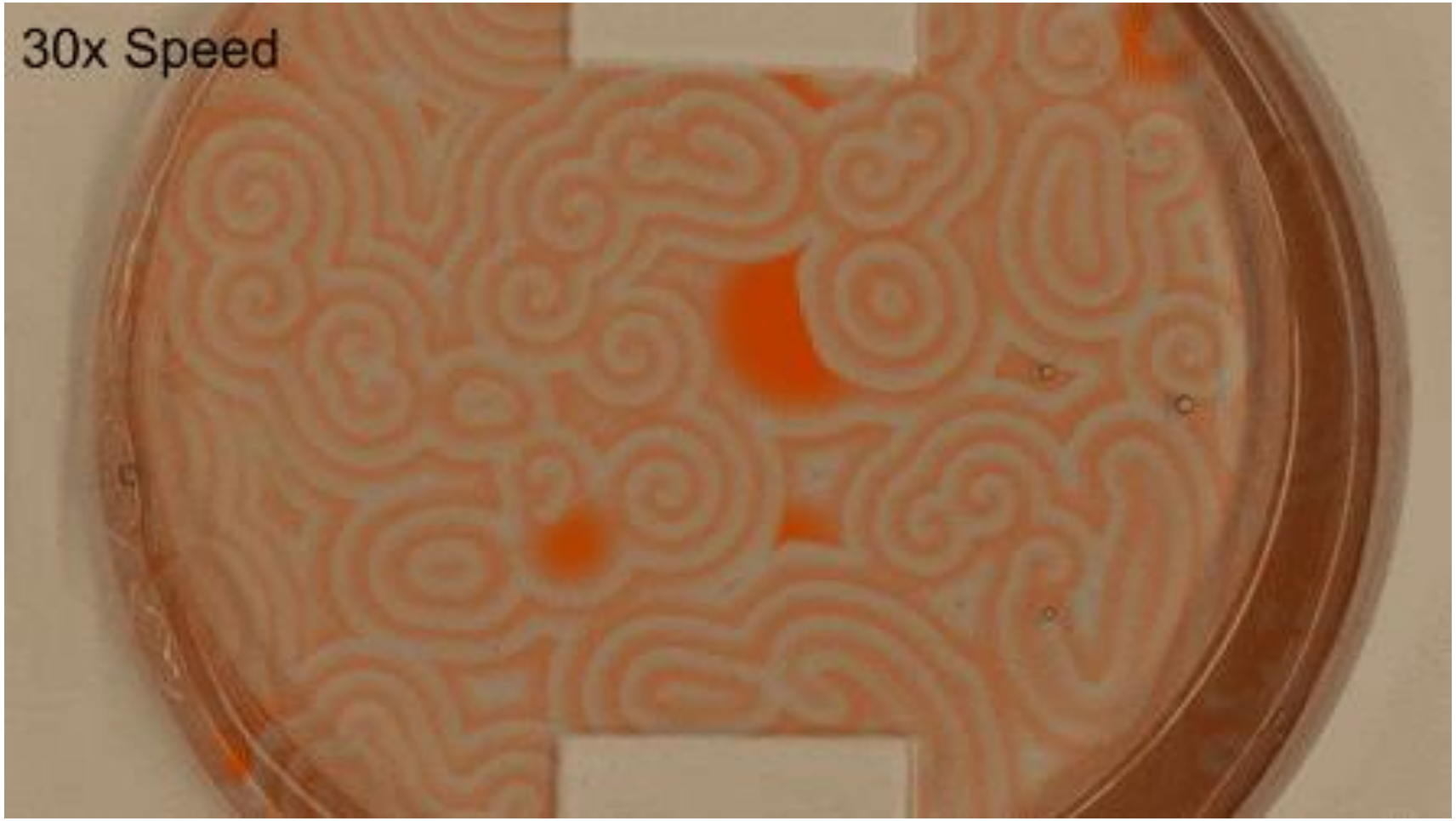
- Causal - Cause and effect is known
- Deterministic (same as above)
- Reducible
- Completely defined
- Finite dimensions
- Predictable

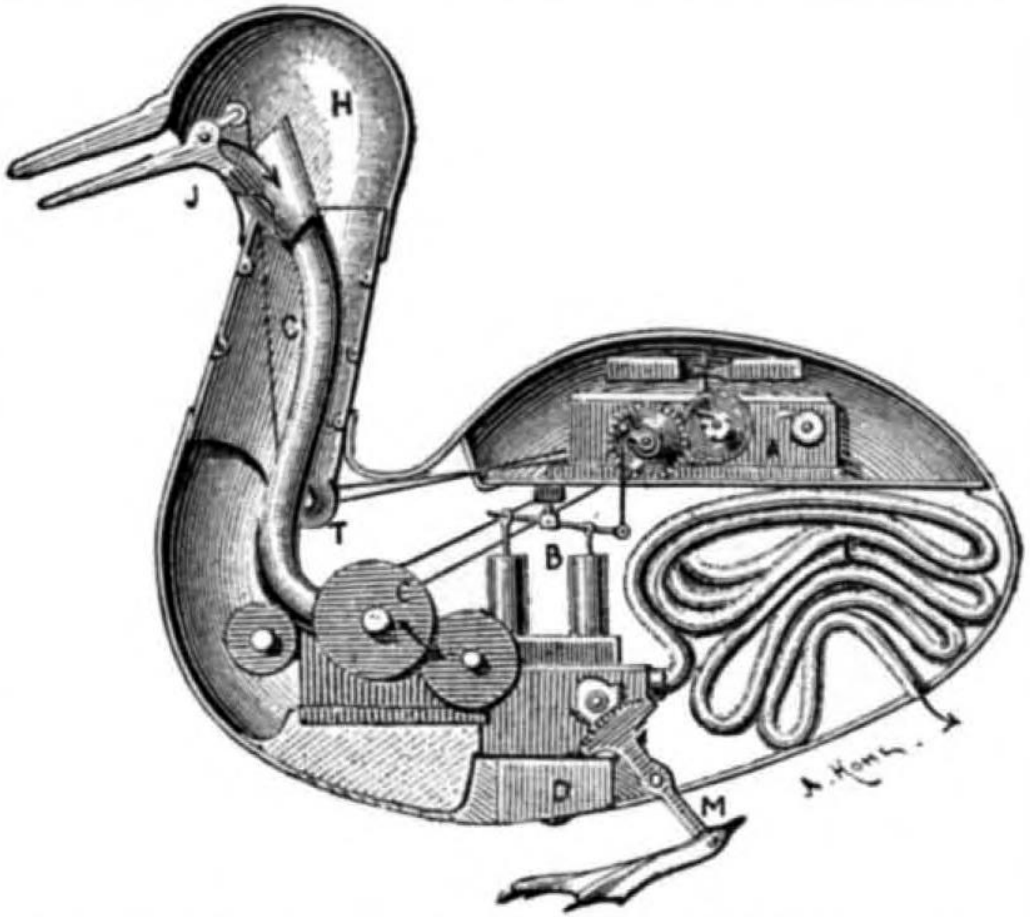
Examples:
Car Engine, Tax Code

Let’s Play Complex vs Complicated

T-GDI

Belousov-Zhabotinsky Reaction





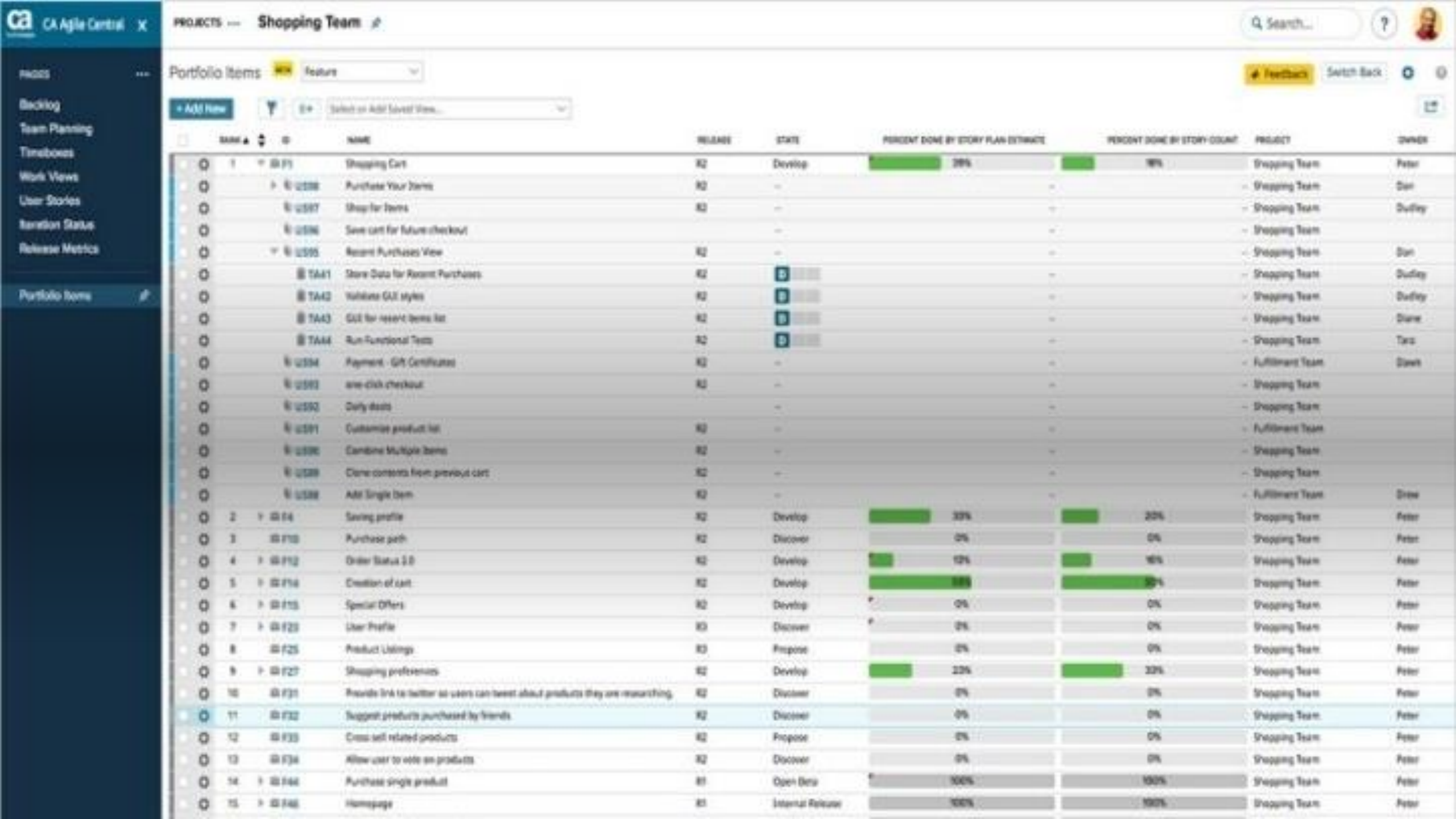
INTERIOR OF VAUCANSON'S AUTOMATIC DUCK.

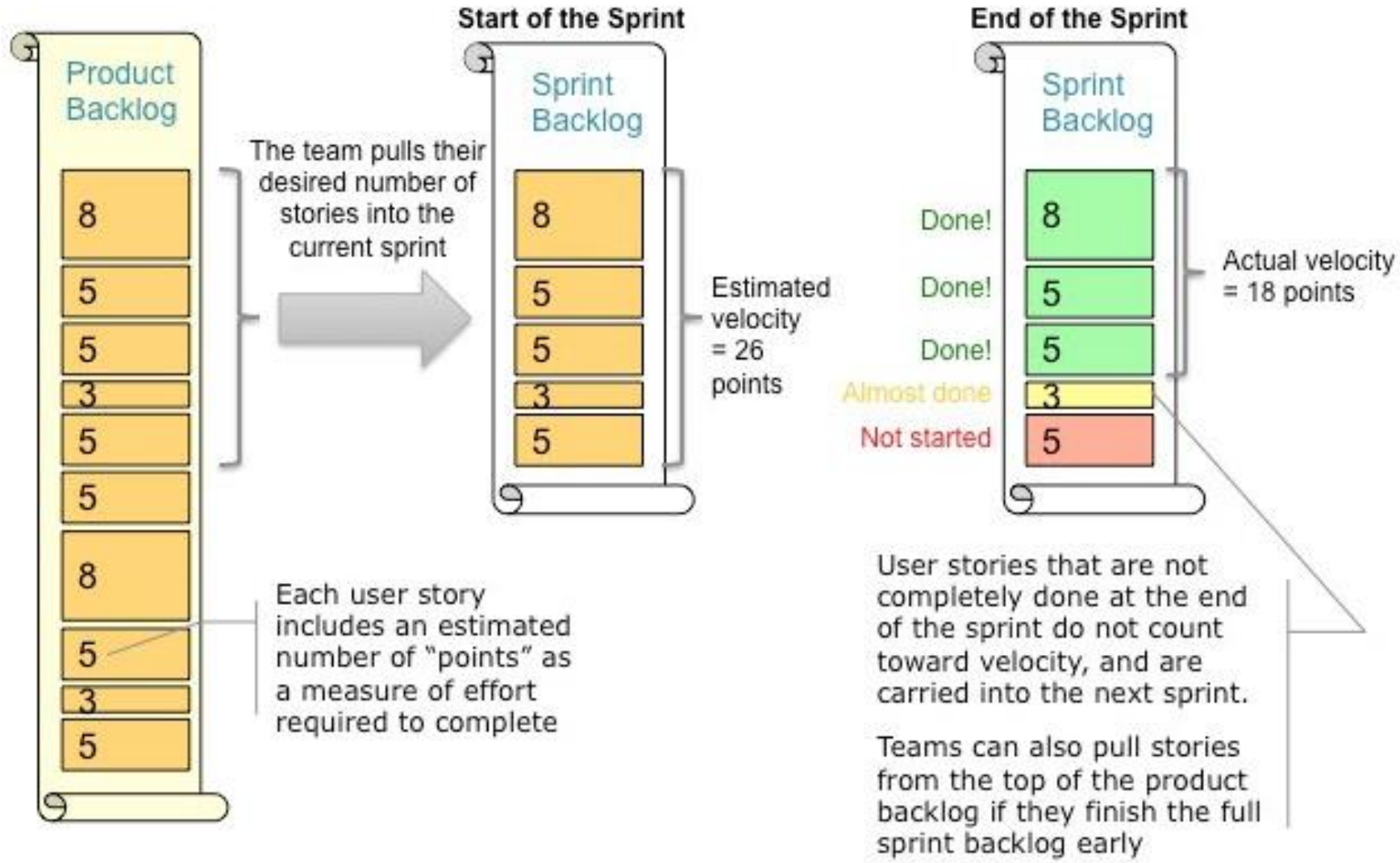
A, clockwork; *B*, pump; *C*, mill for grinding grain; *F*, intestinal tube; *J*, bill; *H*, head; *M*, feet.

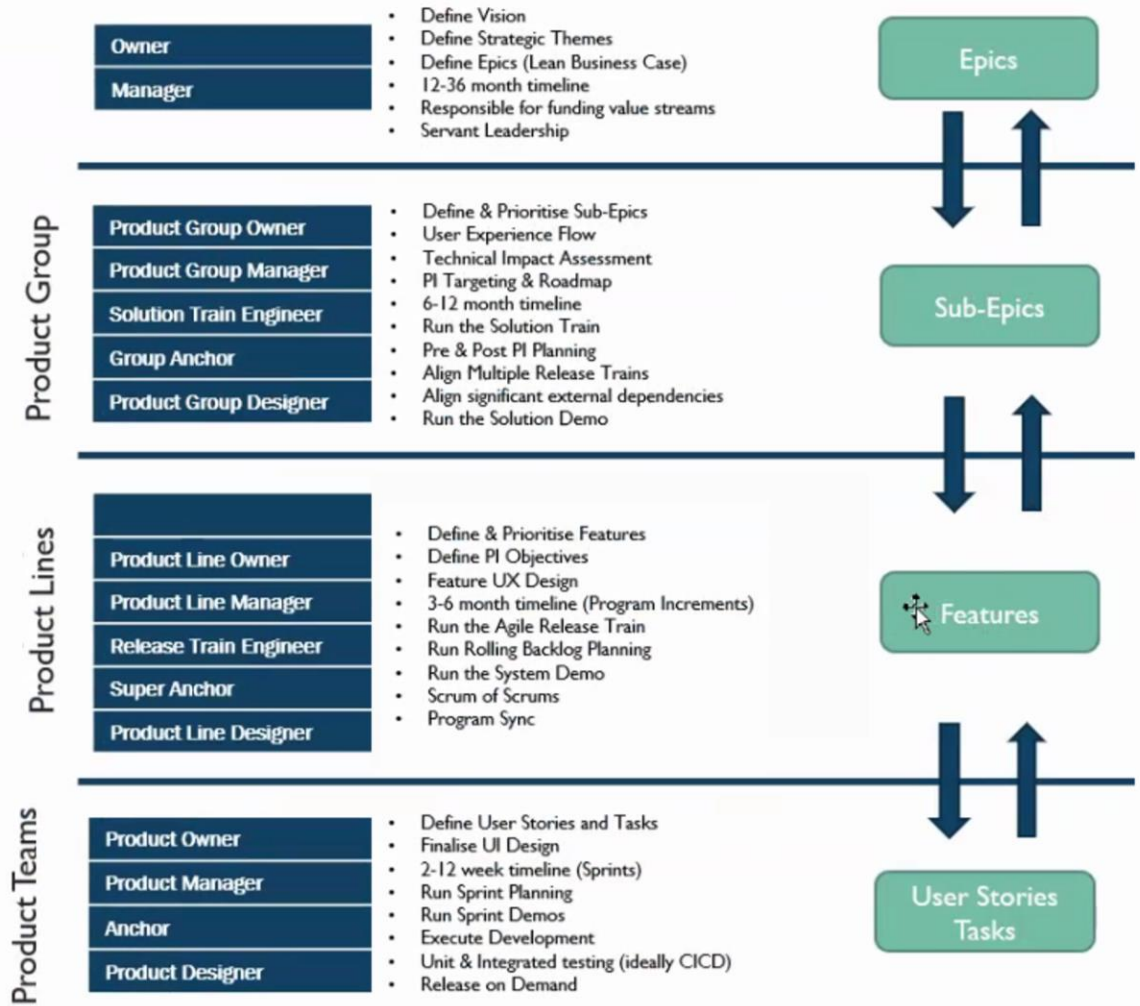
Chladni Plate

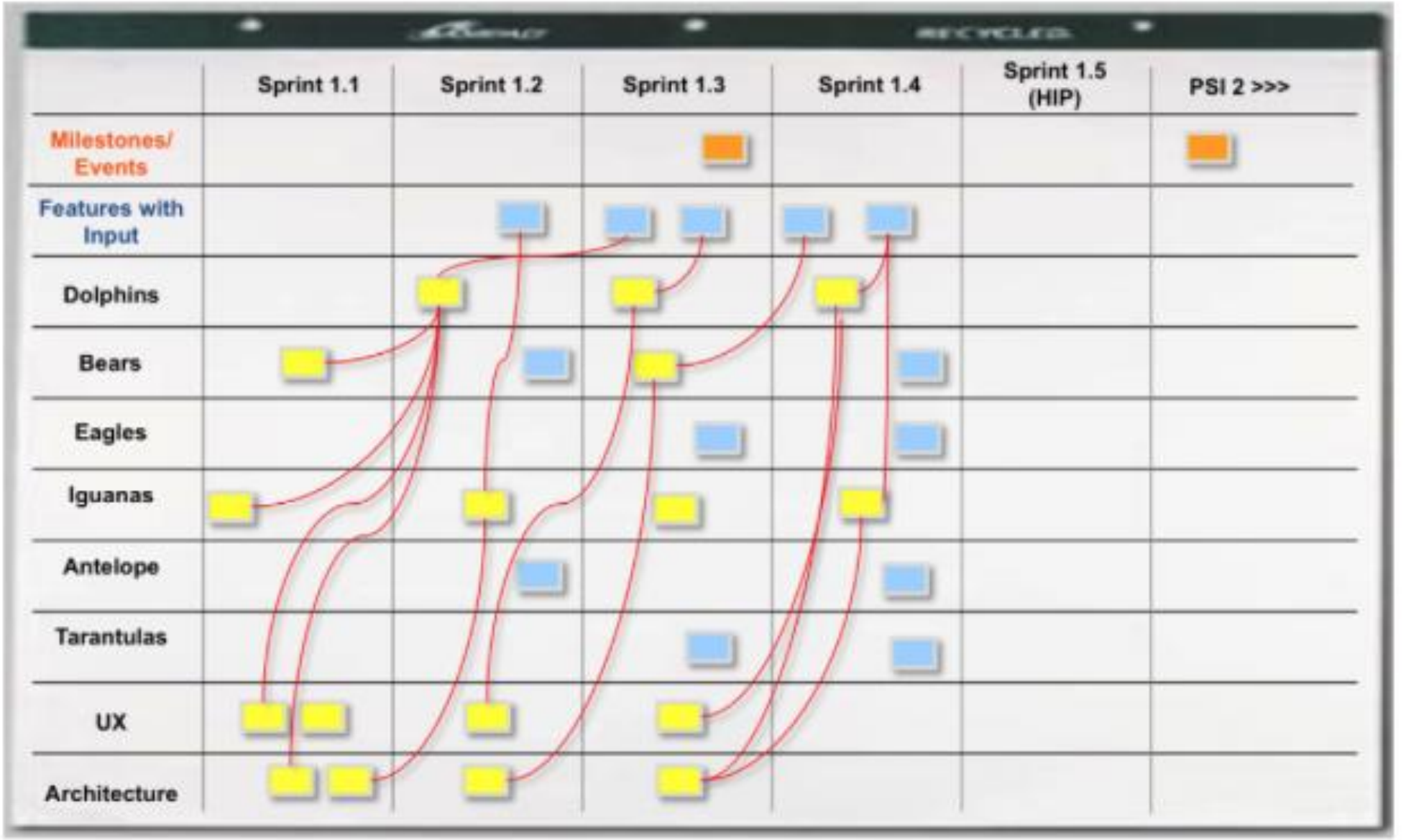
August
September
October

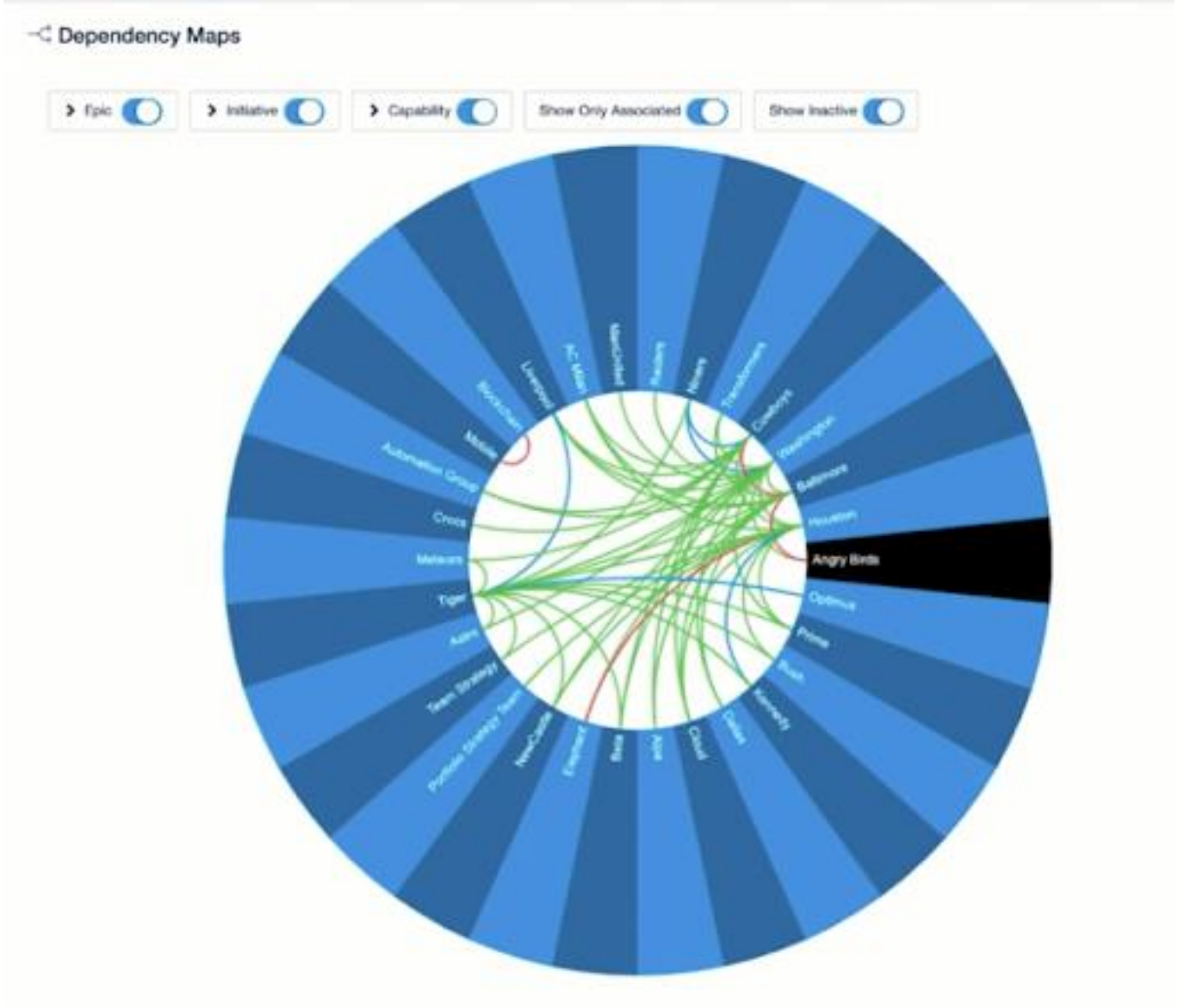
T+ 00:08:30
STAGE 2 TELEMETRY
SPEED
ALTITUDE
18384
226
UPCOMING
STAGE 1 LANDING
THE FIRST STAGE OF FALCON 9 IS ATTEMPTING AN EXPERIMENTAL LANDING ON THE AUTONOMOUS SPACEPORT DRONE SHIP
LAUNCH: CRS-8
ENGINE CHILL
FALCON INTERNAL POWER
STARTUP
MAX-Q
S2 STARTUP
EXPERIMENTAL LANDING
DRAGON DEPLOY
DRAGON INTERNAL POWER
STRONGBACK
LIFTOFF
MECO
BOOSTBACK BURN
SECO
SPACEX

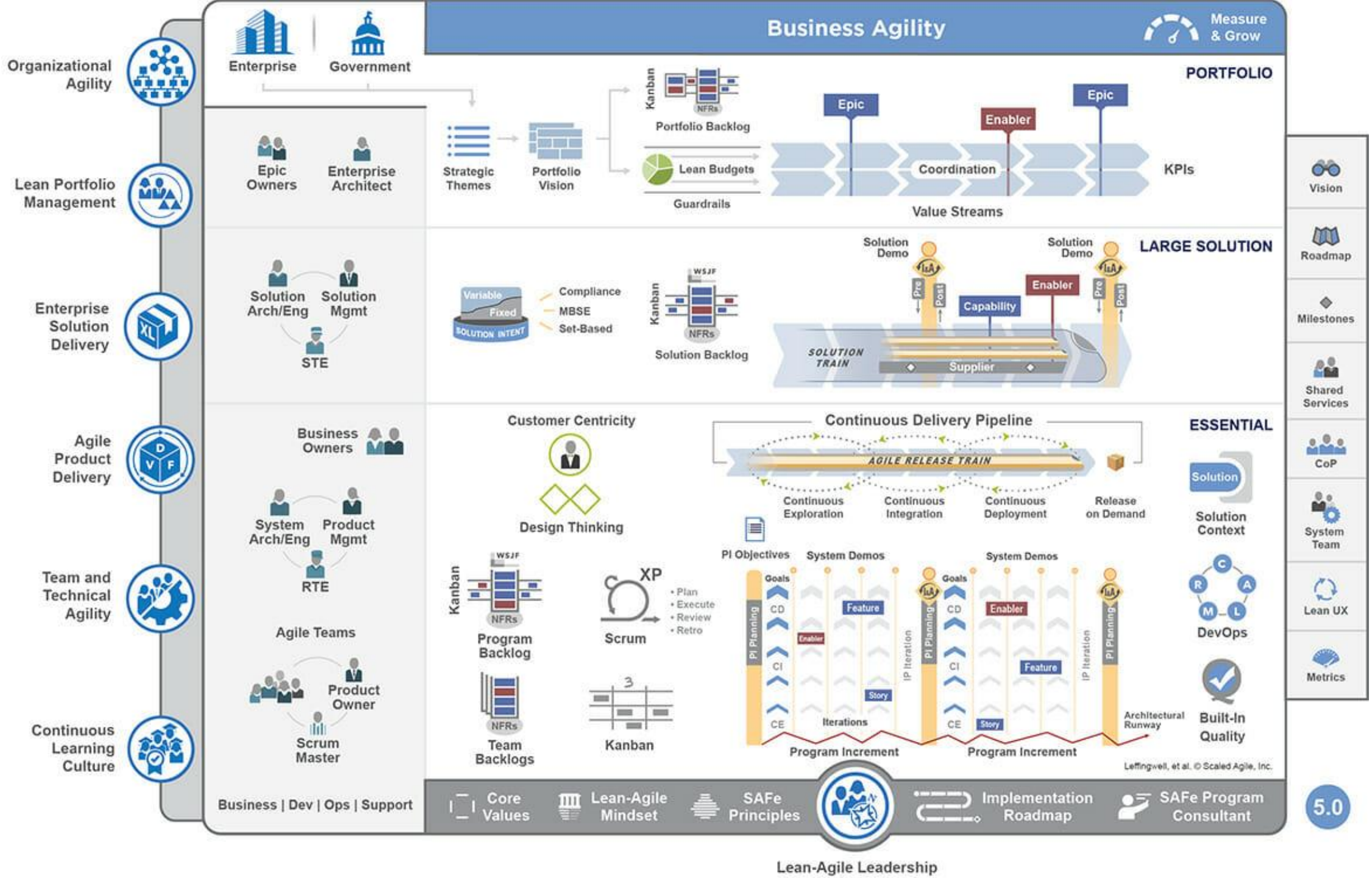

CA Agile Central
PROJECTS
Shopping Team
Search...
Backlog
Team Planning
Timeboxes
Work Views
User Stories
Iteration Status
Release Metrics
Portfolio Items
Portfolio Items
Feature
+ Add New
Feedback
Switch Back
RANK
ID
NAME
RELEASE
STATE
PERCENT DONE BY STORY PLAN ESTIMATE
PERCENT DONE BY STORY COUNT
PROJECT
OWNER
F1 Shopping Cart R2 Develop 28% 16% Shopping Team Peter
US58 Purchase Your Items R2 – – Shopping Team Dan
US57 Shop for Items R2 – – Shopping Team Dudley
US56 Save cart for future checkout – – Shopping Team
US55 Recent Purchases View R2 – – Shopping Team Dan
TA41 Store Data for Recent Purchases R2 D – Shopping Team Dudley
TA42 Validate GUI styles R2 D – Shopping Team Dudley
TA43 GUI for recent items list R2 D – Shopping Team Diane
TA44 Run Functional Tests R2 D – Shopping Team Tara
US54 Payment - Gift Certificates R2 – – Fulfillment Team Dawk
US53 one-click checkout R2 – – Shopping Team
US52 Daily deals – – Shopping Team
US51 Customize product list R2 – – Fulfillment Team
US50 Combine Multiple Items R2 – – Shopping Team
US49 Clone contents from previous cart R2 – – Shopping Team
US48 Add Single Item R2 – – Fulfillment Team Drew
2 F4 Saving profile R2 Develop 33% 20% Shopping Team Peter
3 F10 Purchase path R2 Discover 0% 0% Shopping Team Peter
4 F12 Order Status 2.0 R2 Develop 12% 16% Shopping Team Peter
5 F14 Creation of cart R2 Develop 100% 100% Shopping Team Peter
6 F15 Special Offers R2 Develop 0% 0% Shopping Team Peter
7 F23 User Profile R3 Discover 0% 0% Shopping Team Peter
8 F25 Product Listings R3 Propose 0% 0% Shopping Team Peter
9 F27 Shopping preferences R2 Develop 23% 33% Shopping Team Peter
10 F31 Provide link to twitter so users can tweet about products they are researching R2 Discover 0% 0% Shopping Team Peter
11 F32 Suggest products purchased by friends R2 Discover 0% 0% Shopping Team Peter
12 F33 Cross sell related products R2 Propose 0% 0% Shopping Team Peter
13 F34 Allow user to vote on products R2 Discover 0% 0% Shopping Team Peter
14 F44 Purchase single product R1 Open Beta 100% 100% Shopping Team Peter
15 F46 Homepage R1 Internal Release 100% 100% Shopping Team Peter


Product Backlog
8
5
5
3
5
5
8
5
3
5
Each user story includes an estimated number of "points" as a measure of effort required to complete
The team pulls their desired number of stories into the current sprint
Start of the Sprint
Sprint Backlog
8
5
5
3
5
Estimated velocity = 26 points
End of the Sprint
Sprint Backlog
Done!
Done!
Done!
Almost done
Not started
8
5
5
3
5
Actual velocity = 18 points
User stories that are not completely done at the end of the sprint do not count toward velocity, and are carried into the next sprint.
Teams can also pull stories from the top of the product backlog if they finish the full sprint backlog early


Owner
Manager
Define Vision
Define Strategic Themes
Define Epics (Lean Business Case)
12-36 month timeline
Responsible for funding value streams
Servant Leadership
Epics
Product Group
Product Group Owner
Product Group Manager
Solution Train Engineer
Group Anchor
Product Group Designer
Define & Prioritise Sub-Epics
User Experience Flow
Technical Impact Assessment
PI Targeting & Roadmap
6-12 month timeline
Run the Solution Train
Pre & Post PI Planning
Align Multiple Release Trains
Align significant external dependencies
Run the Solution Demo
Sub-Epics
Product Lines
Product Line Owner
Product Line Manager
Release Train Engineer
Super Anchor
Product Line Designer
Define & Prioritise Features
Define PI Objectives
Feature UX Design
3-6 month timeline (Program Increments)
Run the Agile Release Train
Run Rolling Backlog Planning
Run the System Demo
Scrum of Scrums
Program Sync
Features
Product Teams
Product Owner
Product Manager
Anchor
Product Designer
Define User Stories and Tasks
Finalise UI Design
2-12 week timeline (Sprints)
Run Sprint Planning
Run Sprint Demos
Execute Development
Unit & Integrated testing (ideally CICD)
Release on Demand
User Stories
Tasks


Sprint 1.1
Sprint 1.2
Sprint 1.3
Sprint 1.4
Sprint 1.5 (HIP)
PSI 2 >>>
Milestones/ Events
Features with Input
Dolphins
Bears
Eagles
Iguanas
Antelope
Tarantulas
UX
Architecture
Blue = Features
Yellow = Feature Input
Orange = Milestone/ Event
Red String = A dependency requiring a feature input to be completed before a feature can be completed


Dependency Maps
Epic
Initiative
Capability
Show Only Associated
Show Inactive
Angry Birds


Organizational Agility
Lean Portfolio Management
Enterprise Solution Delivery
Agile Product Delivery
Team and Technical Agility
Continuous Learning Culture
Enterprise
Government
Business Agility
Measure & Grow
PORTFOLIO
Epic Owners
Enterprise Architect
Strategic Themes
Portfolio Vision
Kanban
NFRs
Portfolio Backlog
Lean Budgets
Guardrails
Epic
Enabler
Epic
Coordination
KPIs
Value Streams
LARGE SOLUTION
Solution Arch/Eng
Solution Mgmt
STE
Variable
Fixed
SOLUTION INTENT
Compliance
MBSE
Set-Based
WSJF
Kanban
NFRs
Solution Backlog
Solution Demo
Pre
Post
Capability
Enabler
SOLUTION TRAIN
Supplier
ESSENTIAL
Business Owners
Customer Centricity
Design Thinking
Continuous Delivery Pipeline
AGILE RELEASE TRAIN
Continuous Exploration
Continuous Integration
Continuous Deployment
Release on Demand
Solution
Solution Context
System Arch/Eng
Product Mgmt
RTE
PI Objectives
System Demos
Goals
CD
CI
CE
PI Planning
Feature
Enabler
Story
IP Iteration
Iterations
Architectural Runway
Program Increment
Kanban
NFRs
Program Backlog
XP
Scrum
Plan
Execute
Review
Retro
C
A
R
M
L
DevOps
Agile Teams
Product Owner
Scrum Master
NFRs
Team Backlogs
3
Kanban
Built-In Quality
Leffingwell, et al. © Scaled Agile, Inc.
Business | Dev | Ops | Support
Core Values
Lean-Agile Mindset
SAFe Principles
Lean-Agile Leadership
Implementation Roadmap
SAFe Program Consultant
5.0
Vision
Roadmap
Milestones
Shared Services
CoP
System Team
Lean UX
Metrics

Map of Complexity Science

by Brian Castellani

1940-1950s | 1960's | 1970's | 1980's | 1990's | 2000's | 2010's

Dynamic Systems Theory
Fractal Geometry
Chaos Theory
Tien Yien Li (Founder)
James Yorke (Founder)
Edward Lorenz (Lorenz Attractor/Butterfly Effect)
Mitchell Feigenbaum (Chaos Constant)
Benoit Mandelbrot (Founder)

Ludwig von Bertalanffy (Systems Theory Founder & Systems Biology)
Systems Theory
Anatol Rapoport (mathematical psychology)
Kenneth Boulding (economics)
James Grier Miller (living systems theory)
Donella Meadows (globalization)

Jay Forrester (system dynamics)
Fritjof Capra (web of life)
James Lovelock & Lynn Margulis (Gaia Hypothesis)
ecological Systems Theory
Peter Checkland (management/ soft systems modeling)
Social Systems Theory
Systems Science Engineering
Deborah Hammond (history of systems thinking)
George Klir (Systems Science)
Stafford Beer (management)

Complexity Science

Margaret Mead (anthropology)
Gregory Bateson (ecology of mind)
Niklas Luhmann (Sociology)
Ilya Prigogine (Dissipative Structures)
Stuart Kauffmann (Biology/Evolution)
Per Bak (Self-Organized Criticality)

Self-Organization
Autopoiesis and Adaptation
Emergence
Dynamics in Systems
Erich Jantsch (Self-Organizing Universe)
Yaneer Bar-Yam (Dynamics of Complex Systems)
Samuel Bowles (Dynamics of Human Systems)
Barry Wellman (Internet, INSNR)

Geoffrey West (Scaling Laws)
MarK Granovetter (Strength of Weak Ties)
Steve Strogatz (Small World)
Mark Newman (Complex Networks)
New Science of Networks
Duncan Watts (Small World)
Albert-Lazlo Barabasi (Scale-free networks)
Manuel Castells (Global Network Society)
Immanuel Wallerstein (World Systems Theory)
Global Network Society
Web Science
John Urry (mobile societies)

Norbert Weiner (Cybernetics Mathematics)
Cybernetics
Claude Shannon (Information theory)
Heinz von Foerster (2nd order cybernetics)
2nd Order Cybernetics
Francisco Varela & Humberto Maturana (Cognitive Science)
John Holland (Genetic Algorithms)
Socio-Cybernetics
Felix Geyer (Sociology)
Francisco Parra-Luna (Founder)

John Taylor (e-science)
E-Science

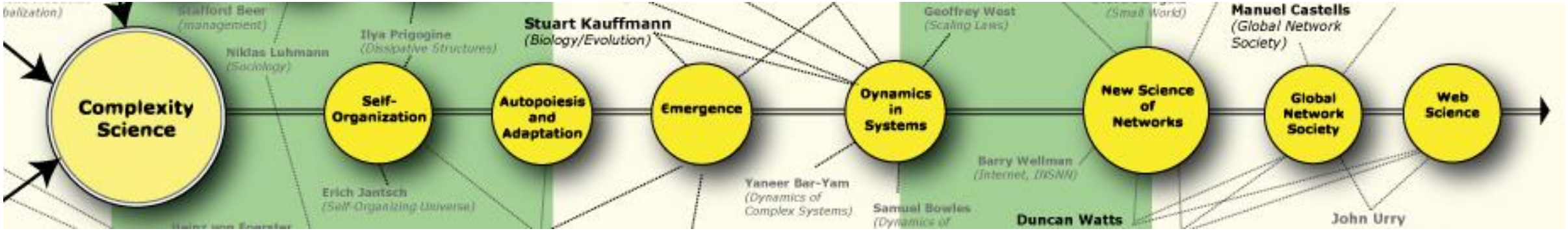
John von Neumann (mathematics)
W. Ross Ashby (Cybernetics of Mind)

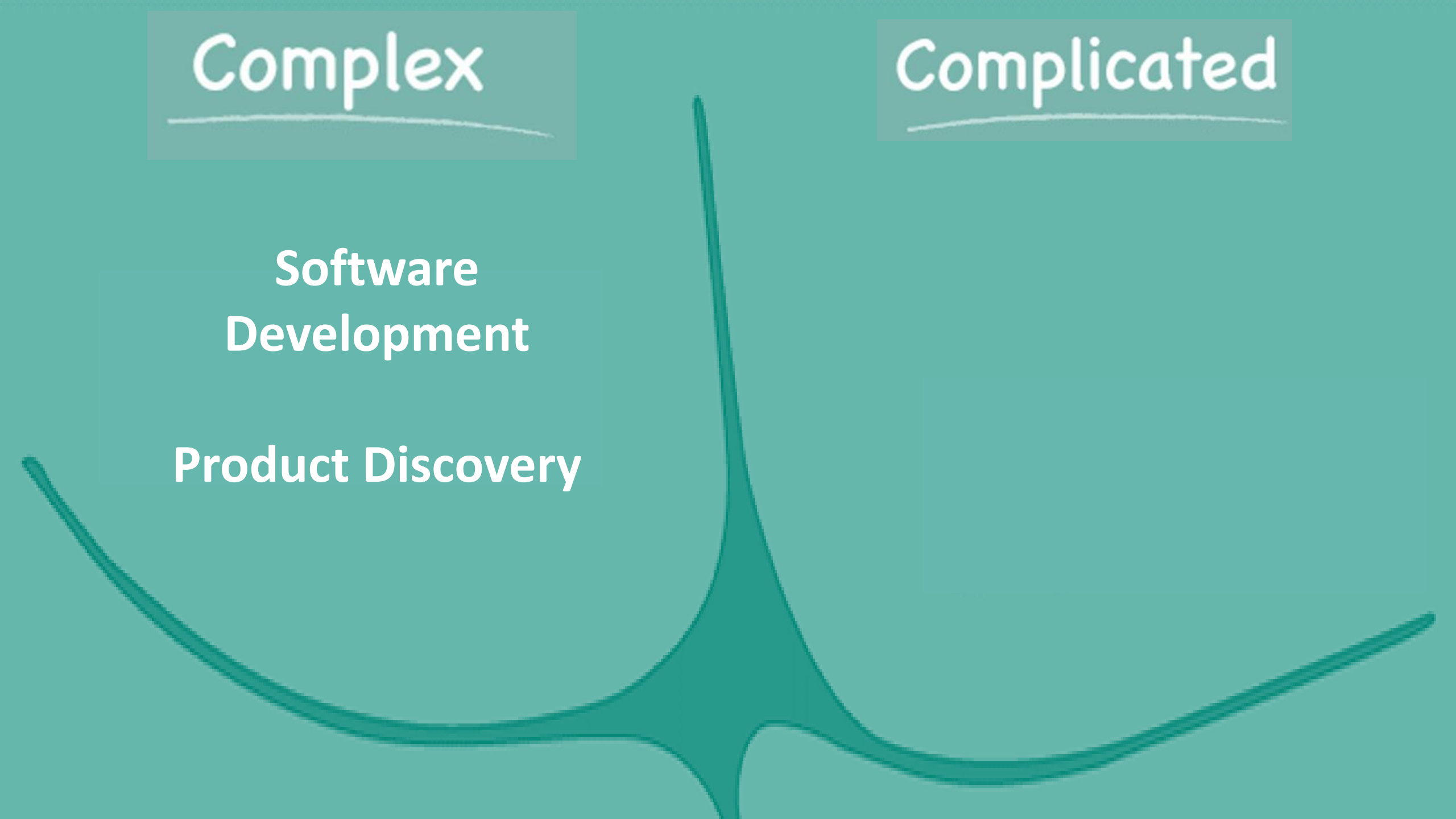
Stephen Wolfram (Complexity)
Agent Based Modeling
Cellular Automata
John Conway (Game of Life)
Genetic Algorithms
Christopher Langton (Founder)
Artificial Life
Robert Axelrod (Complexity of Cooperation)
Multi-Agent Modeling
Katia Sycara (Editor, AAMAS)
Michael Wooldridge (Editor, AAMAS)
Brian Castellani & Fred Hafferty (SACS Toolkit)
Melanie Mitchell (computation in complex systems)
Nigel Gilbert (computational Sociology)
John Miller (computational economics)
Computational Modeling
Jeanette Wing (computational thinking)
Claudio Cioffi-Revilla (computational social sience)
Scott Page (computational political science)
Joshua Epstein & Robert Axtell (Artificial Societies)

Artificial Intelligence
Walter Pitts (Founder)
Frank Rosenblatt (Connectionism)
Distributed Neural Networking
Warren McCulloch (Founder)
Teuvo Kohnan (Self-Organizing Map)
Michio Sugeno (Fuzzy Math)
Lotfi Zadeh (Founder)
Fuzzy Logic
Bart Kosko (Fuzzy Logic)
Data Mining
Michael Berry and Gordon Linoff (Data Mining in Business)

Complexity – see words like:

- Self-Organization
- Adaptation
- Emergence
- Networks




Complex
Complicated
Software Development
Product Discovery

Summary – Complex vs Complicated

Complex
Not exactly predictable yet Ordered

Complicated
Exactly predictable and Ordered

What – what is the solution?

How can we represent complexity?

We need a way to represent complex work

Alan Cooper

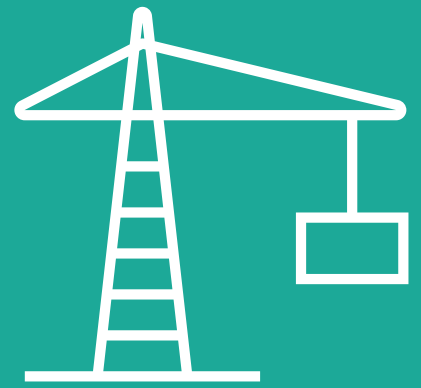
@MrAlanCooper

The world of software behavior is complex, multivariate, and conditional. The mind can't grasp all of the permutations without a map, a visual map, an adaptive, explanatory, exploratory, working tool of a map. Of many maps.

Representing Complexity

With Product Maps and Discovery Trees

What - Thinking Big: Product Mapping



Thinking Big – A tool for Product Managers to visualize the big picture so that the entire tribe has a context of where we are going and the context of the work items. Show progress, releases, and can be used for forecasting completion.

We started with a Story Map – but then found limitations/issues…

Story Mapping Limitations and Issues

- work items were splintering so that it was hard to represent on a single board

- it was hard to keep up with the speed as the tribe grew

SECURITY
ETL
MONITORING
LOGGING
DEVOPS
CI CD
API VERSIONING
API MANAGEMENT
TESTING
RESILIANCE
FAILOVER
PROCESS
inc SDL
Our Story Map Evolved into a Product Map : We kept only high-level items here

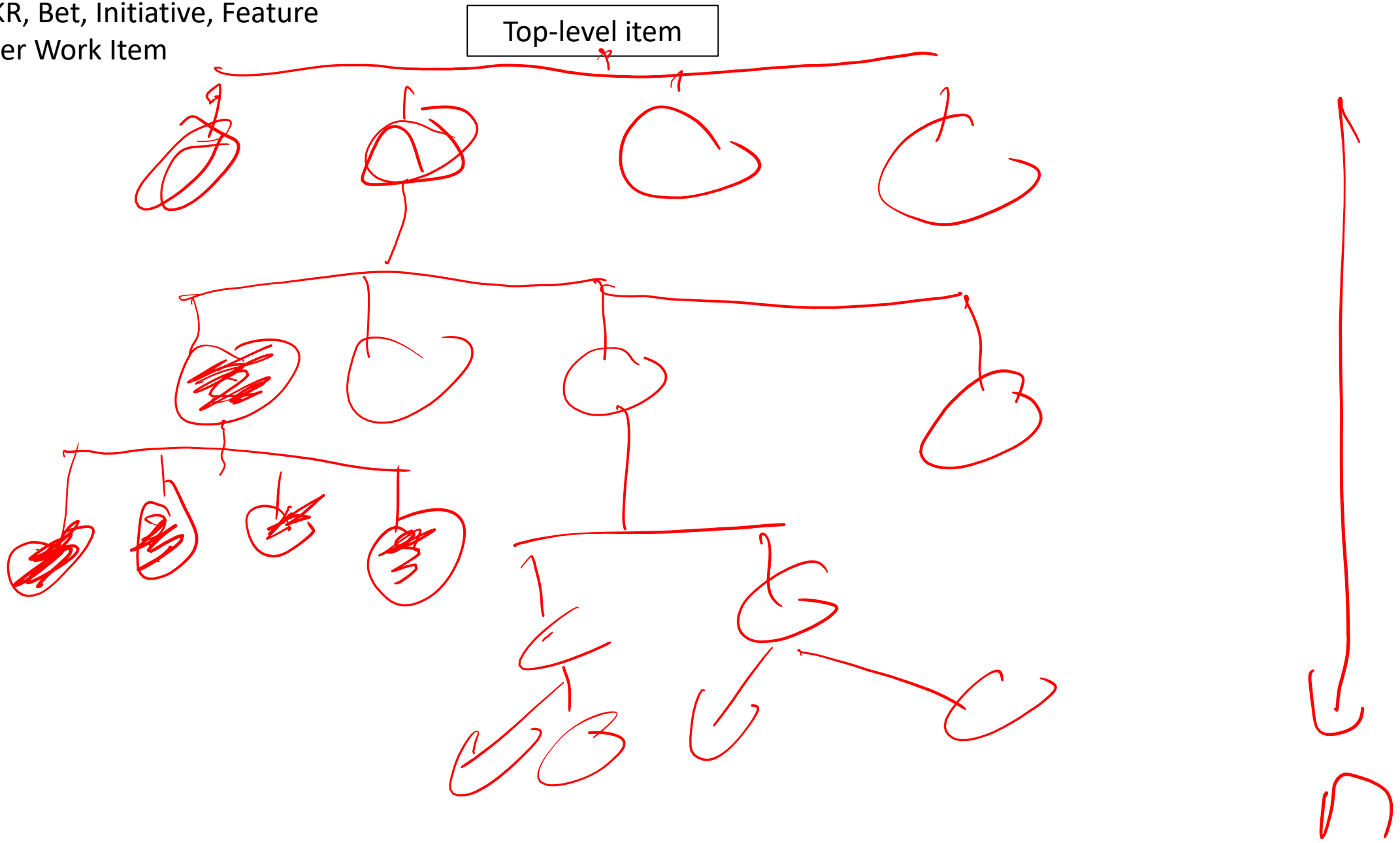
Summary of Benefits of Product Map

- Builds on story mapping
- View of the product from a Product Manager's POV
- Gives the broader and bigger picture of what we are building
- Gives context to work
- Can view quickly where we are at
- Can be used for quick forecasting
  (especially if done with wisdom of the crowds forecasting method)
- Can map out releases

What – Working Small: Discovery Trees

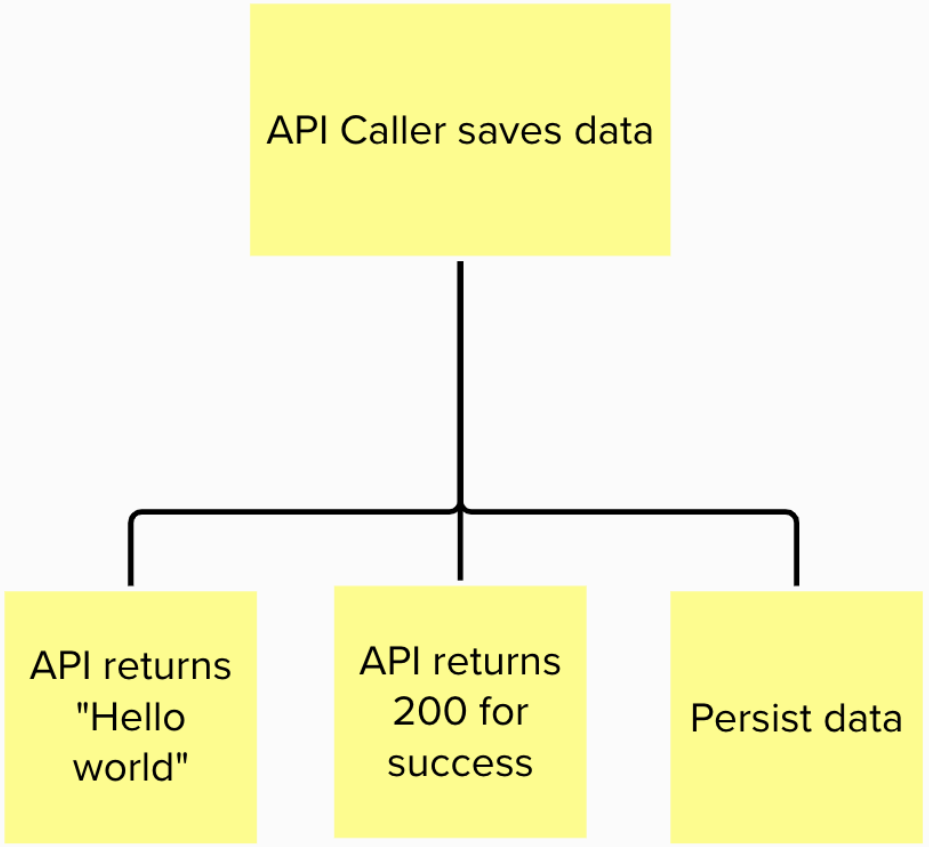
Discovery Trees are mostly for devs: aid breakdown, understand work, track work, report on progress, forecast completion (optional), and maintaining context.

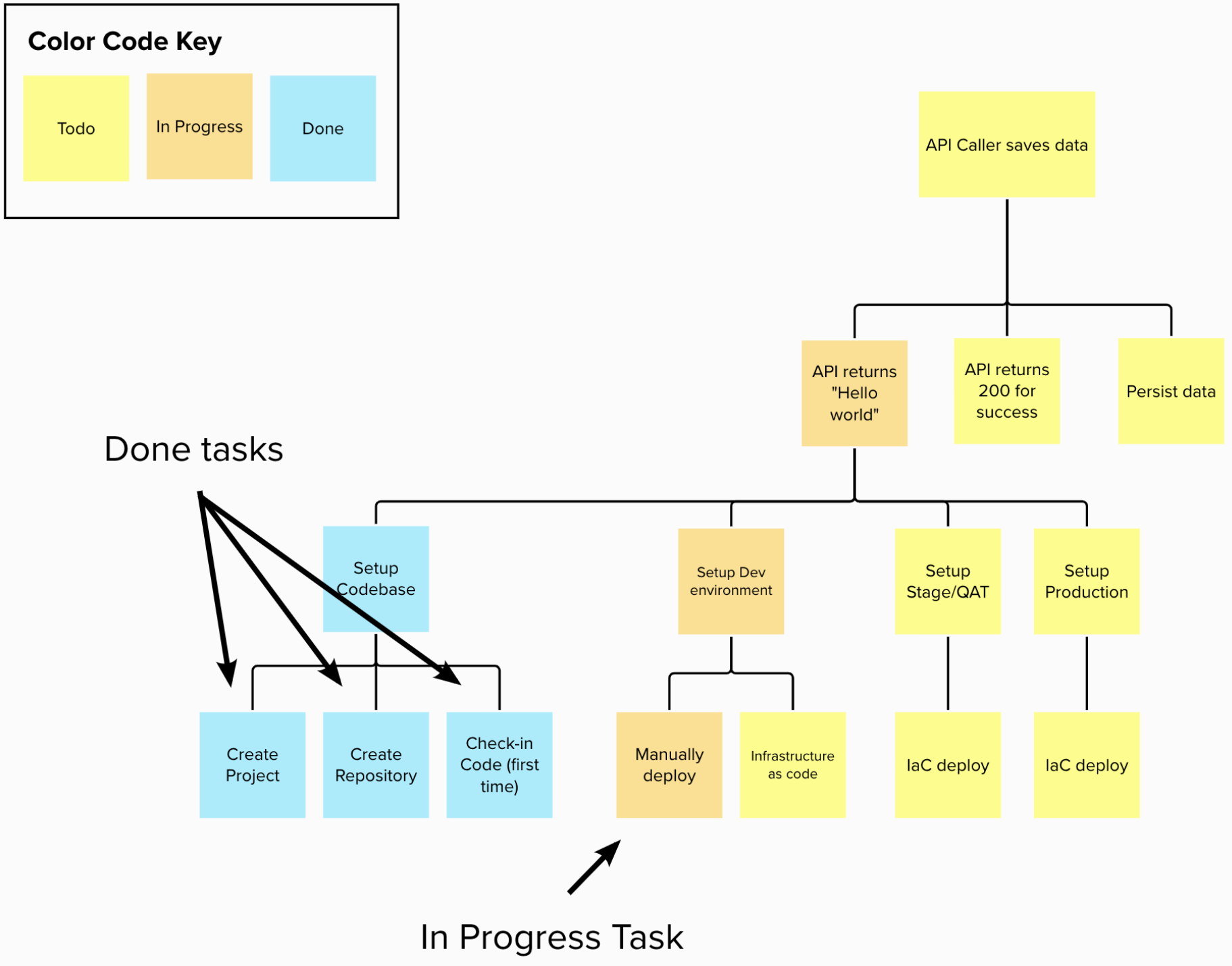
Epic, OKR, Bet, Initiative, Feature
i.e. Larger Work Item

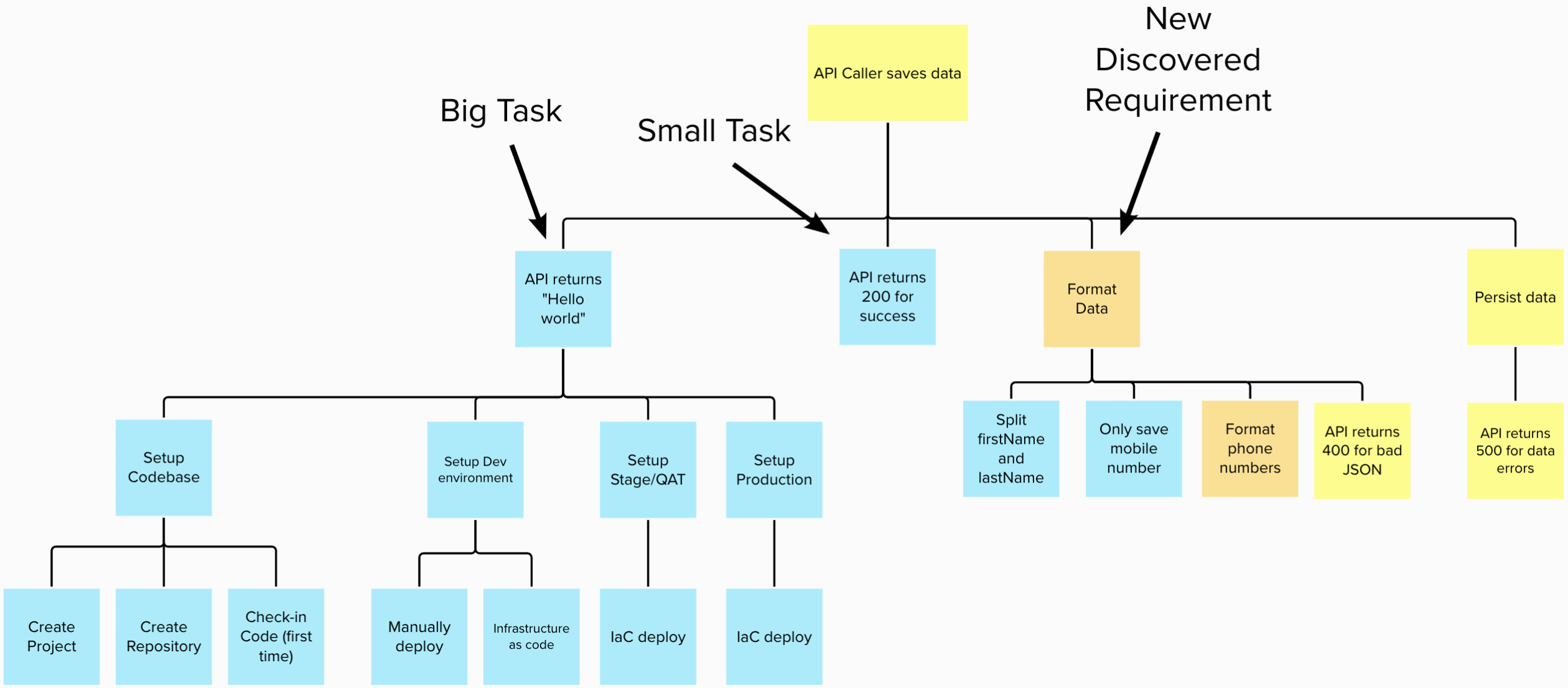


Example Discovery Tree : Delivery

Root node is a larger goal




Color Code Key
Todo
In Progress
Done
API Caller saves data
API returns "Hello world"
API returns 200 for success
Persist data
Done tasks
Setup Codebase
Setup Dev environment
Setup Stage/QAT
Setup Production
Create Project
Create Repository
Check-in Code (first time)
Manually deploy
Infrastructure as code
IaC deploy
IaC deploy
In Progress Task


Big Task
Small Task
API Caller saves data
New Discovered Requirement
API returns "Hello world"
API returns 200 for success
Format Data
Persist data
Setup Codebase
Setup Dev environment
Setup Stage/QAT
Setup Production
Split firstName and lastName
Only save mobile number
Format phone numbers
API returns 400 for bad JSON
API returns 500 for data errors
Create Project
Create Repository
Check-in Code (first time)
Manually deploy
Infrastructure as code
IaC deploy
IaC deploy

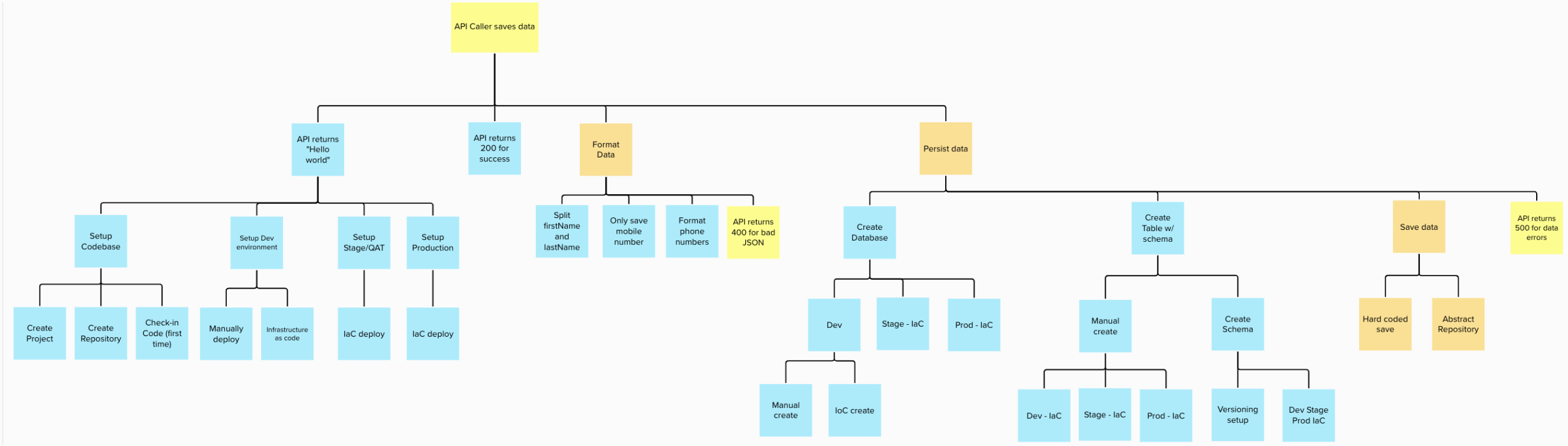
This is why it is called a Discovery Tree…

Woody Zuill
@WoodyZuill

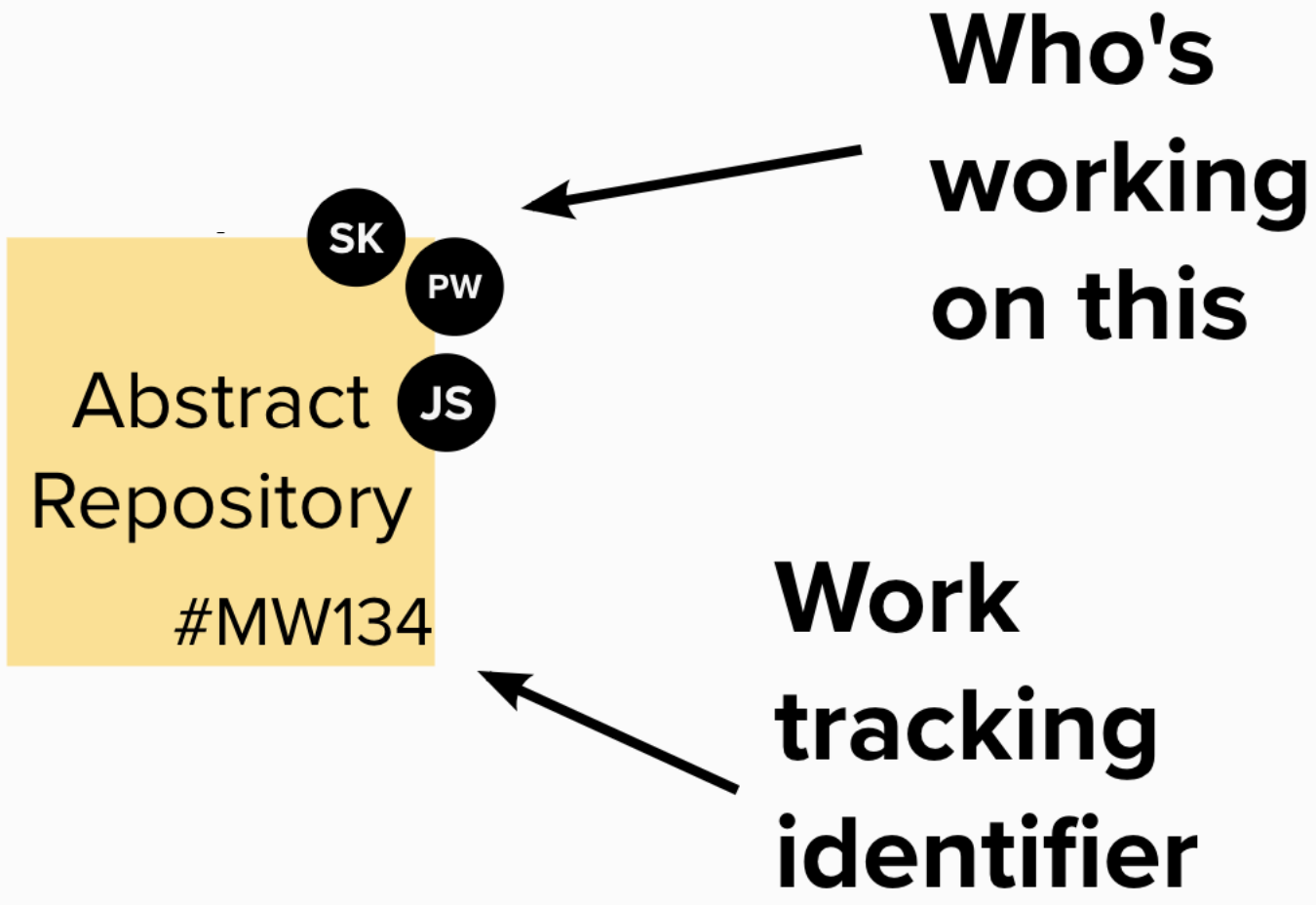
As I say: "it's in the doing of the work that we discover the work we must do", (or the point where balance is achieved)

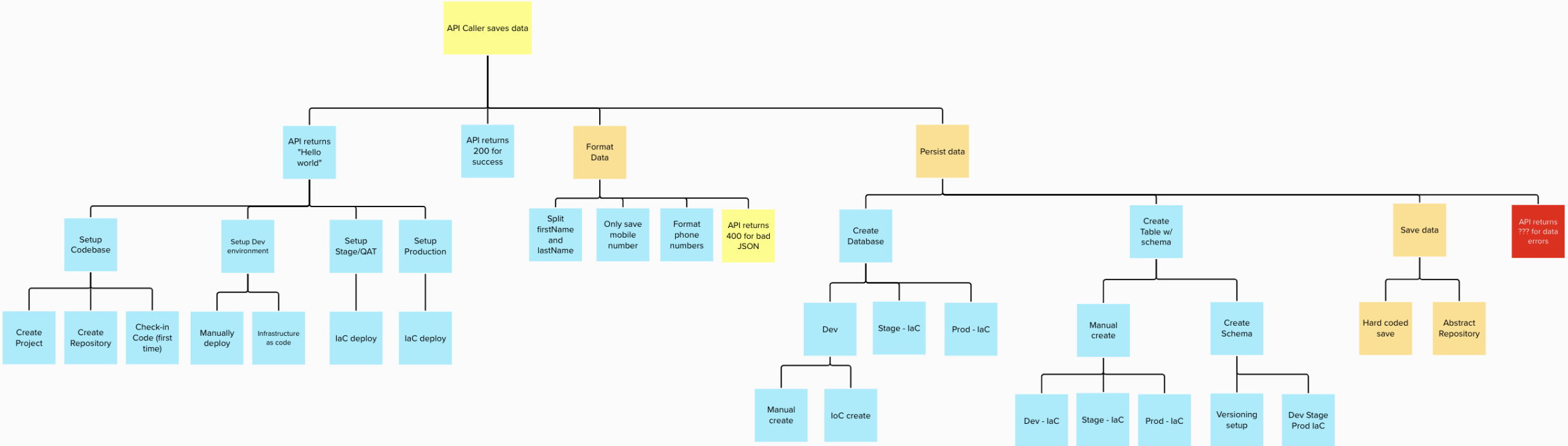
6:23 PM · Jul 15, 2016 · Twitter for iPhone

The tree can grow to n x n dimensions
and is not restricted by common agile conventions for defining work…



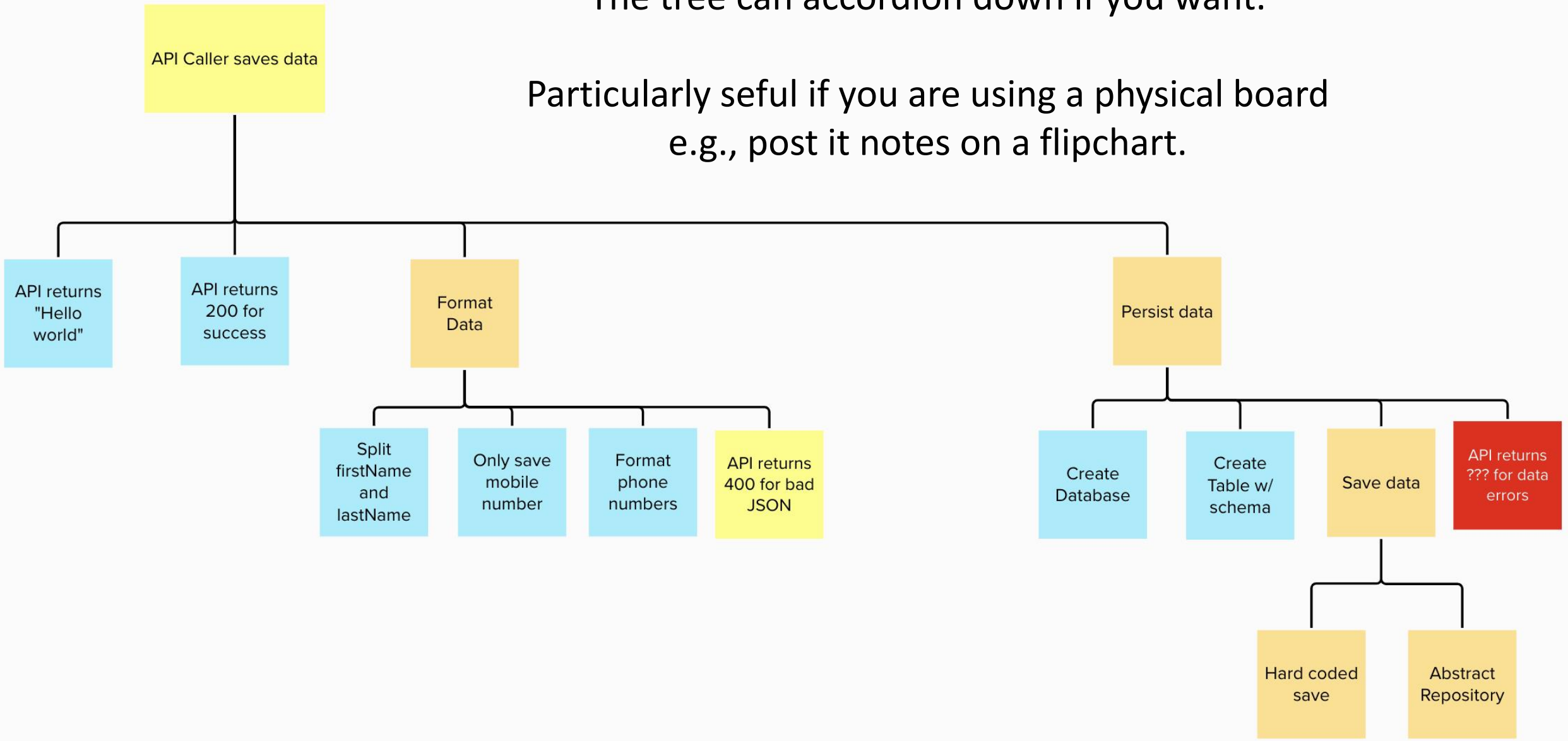
The tool gives you freedom to manage whatever meta-data you want, including color coding.

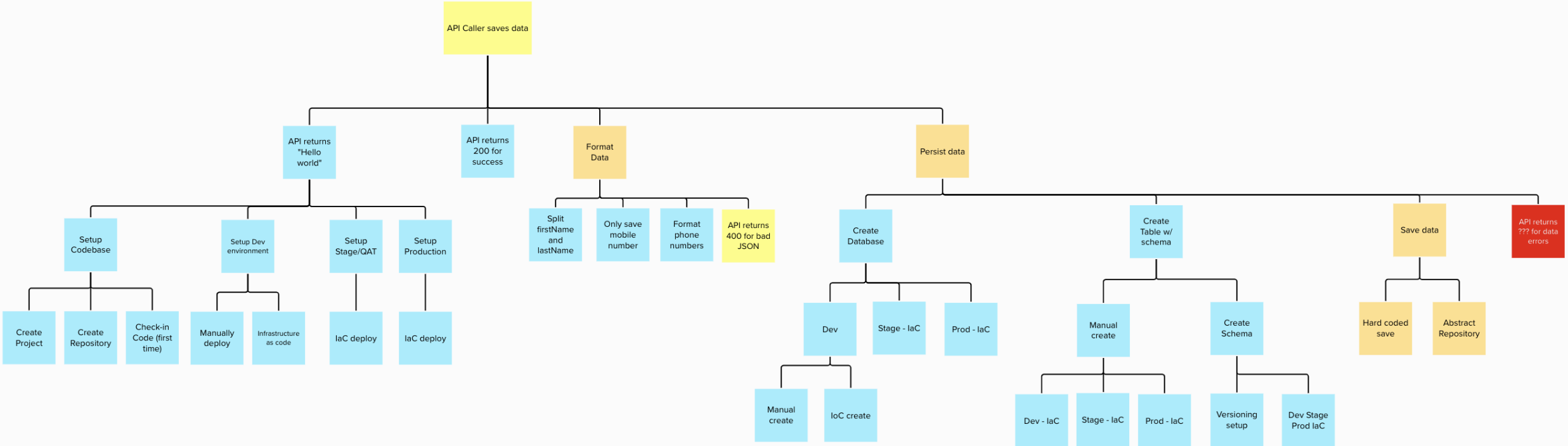



API Caller saves data
API returns "Hello world"
API returns 200 for success
Format Data
Persist data
Setup Codebase
Setup Dev environment
Setup Stage/QAT
Setup Production
Split firstName and lastName
Only save mobile number
Format phone numbers
API returns 400 for bad JSON
Create Database
Create Table w/ schema
Save data
API returns ??? for data errors
Create Project
Create Repository
Check-in Code (first time)
Manually deploy
Infrastructure as code
IaC deploy
IaC deploy
Dev
Stage - IaC
Prod - IaC
Manual create
Create Schema
Hard coded save
Abstract Repository
Manual create
IoC create
Dev - IaC
Stage - IaC
Prod - IaC
Versioning setup
Dev Stage Prod IaC

The tree can accordion down if you want.

Particularly seful if you are using a physical board
e.g., post it notes on a flipchart.




API Caller saves data
API returns "Hello world"
API returns 200 for success
Format Data
Persist data
Setup Codebase
Setup Dev environment
Setup Stage/QAT
Setup Production
Split firstName and lastName
Only save mobile number
Format phone numbers
API returns 400 for bad JSON
Create Database
Create Table w/ schema
Save data
API returns ??? for data errors
Create Project
Create Repository
Check-in Code (first time)
Manually deploy
Infrastructure as code
IaC deploy
IaC deploy
Dev
Stage - IaC
Prod - IaC
Manual create
Create Schema
Hard coded save
Abstract Repository
Manual create
IoC create
Dev - IaC
Stage - IaC
Prod - IaC
Versioning setup
Dev Stage Prod IaC

Working With Discovery Trees

Steve Kuo

24 May 2022 7 min read facebook twitter linkedin

Agile Agile Development Behavior Skills Coaching Culture Development Estimates Process Improvement Product Software Design

Steve Kuo

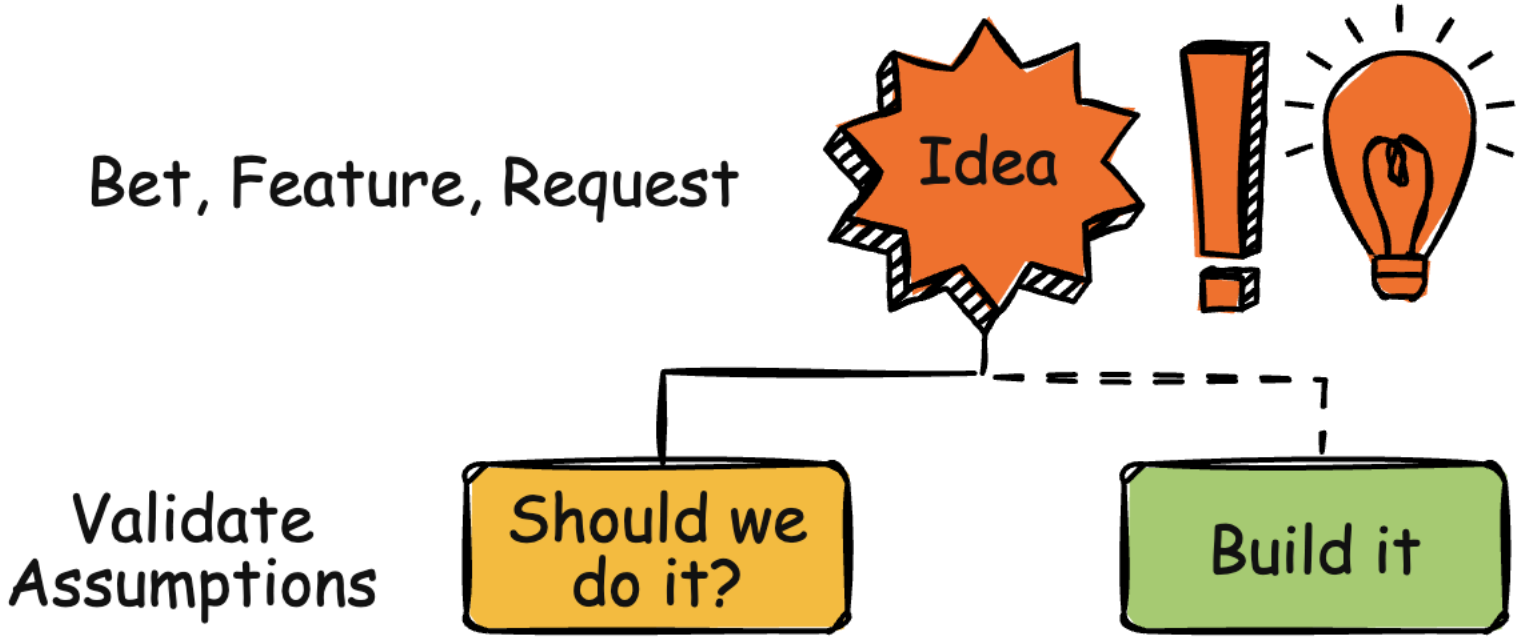
Senior Consultant
Seattle, WA - USA

Are you struggling with tracking work through your system? Finding it difficult to get the “big picture” view of how complete a feature is? Wrestling with thin

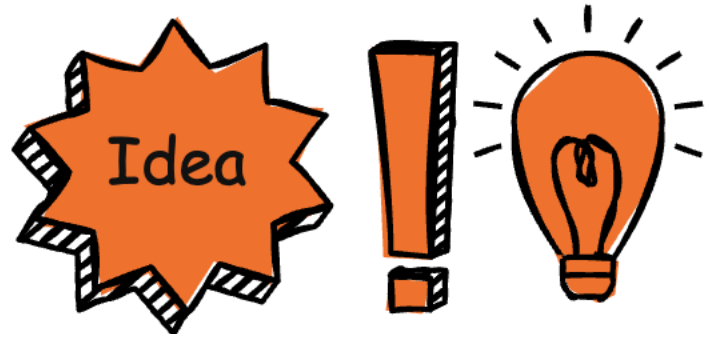
https://www.industriallogic.com/blog/discovery-trees/

Attribution

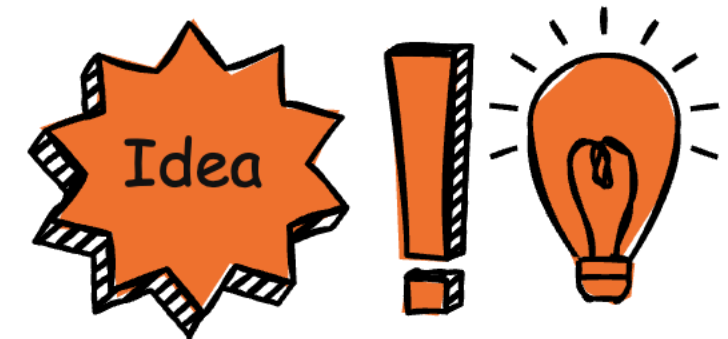
Example Discovery Tree : Discovery



Bet, Feature, Request



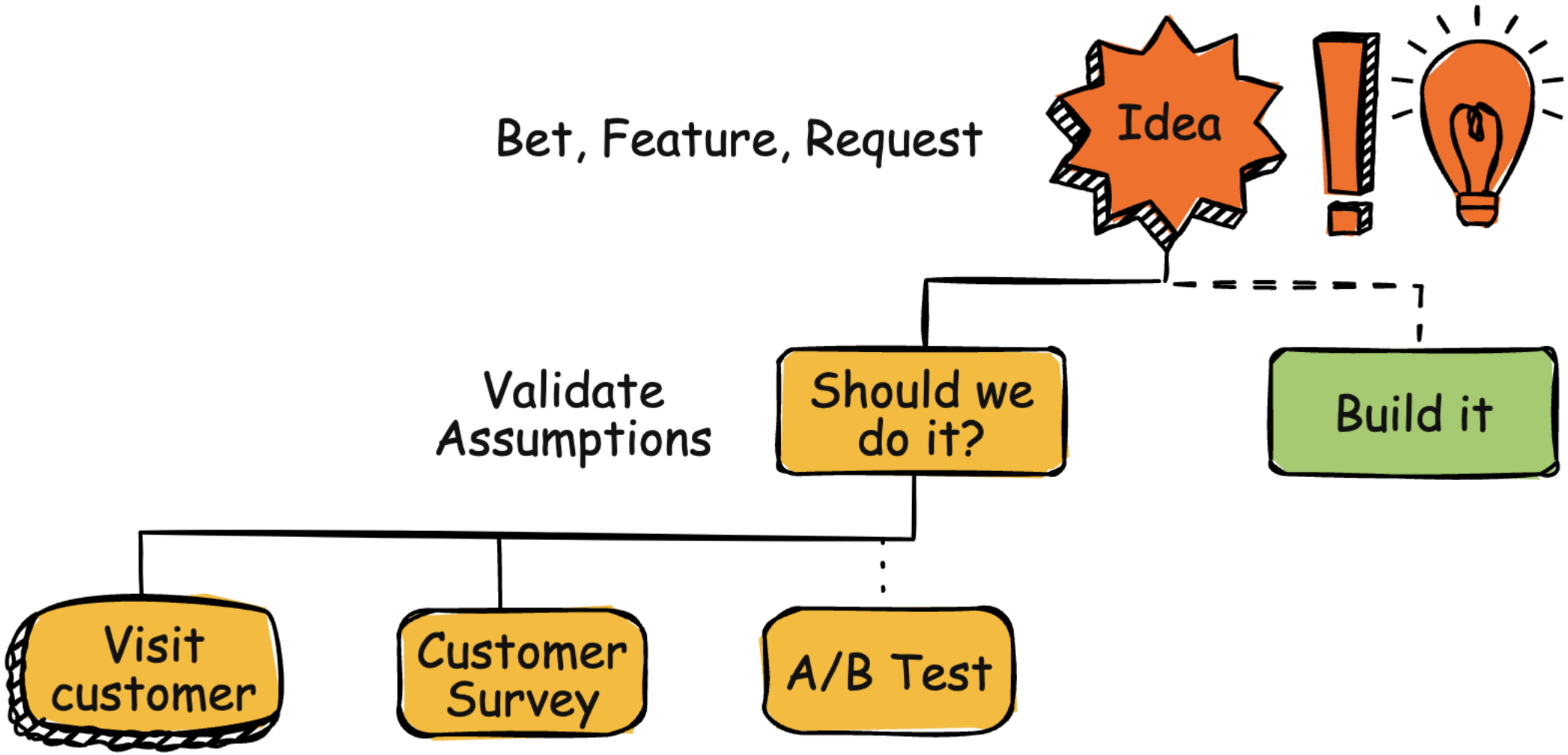
Bet, Feature, Request

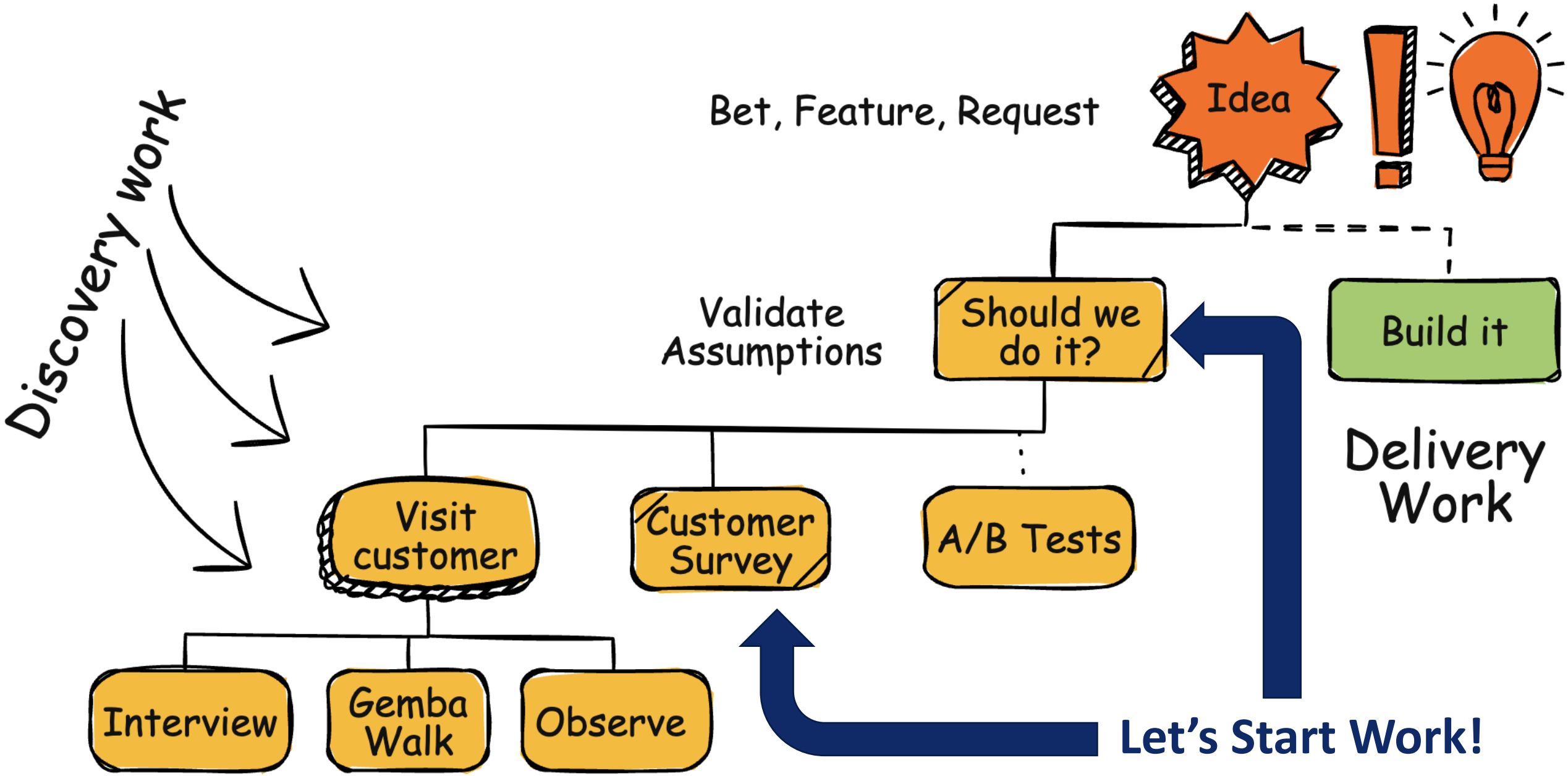


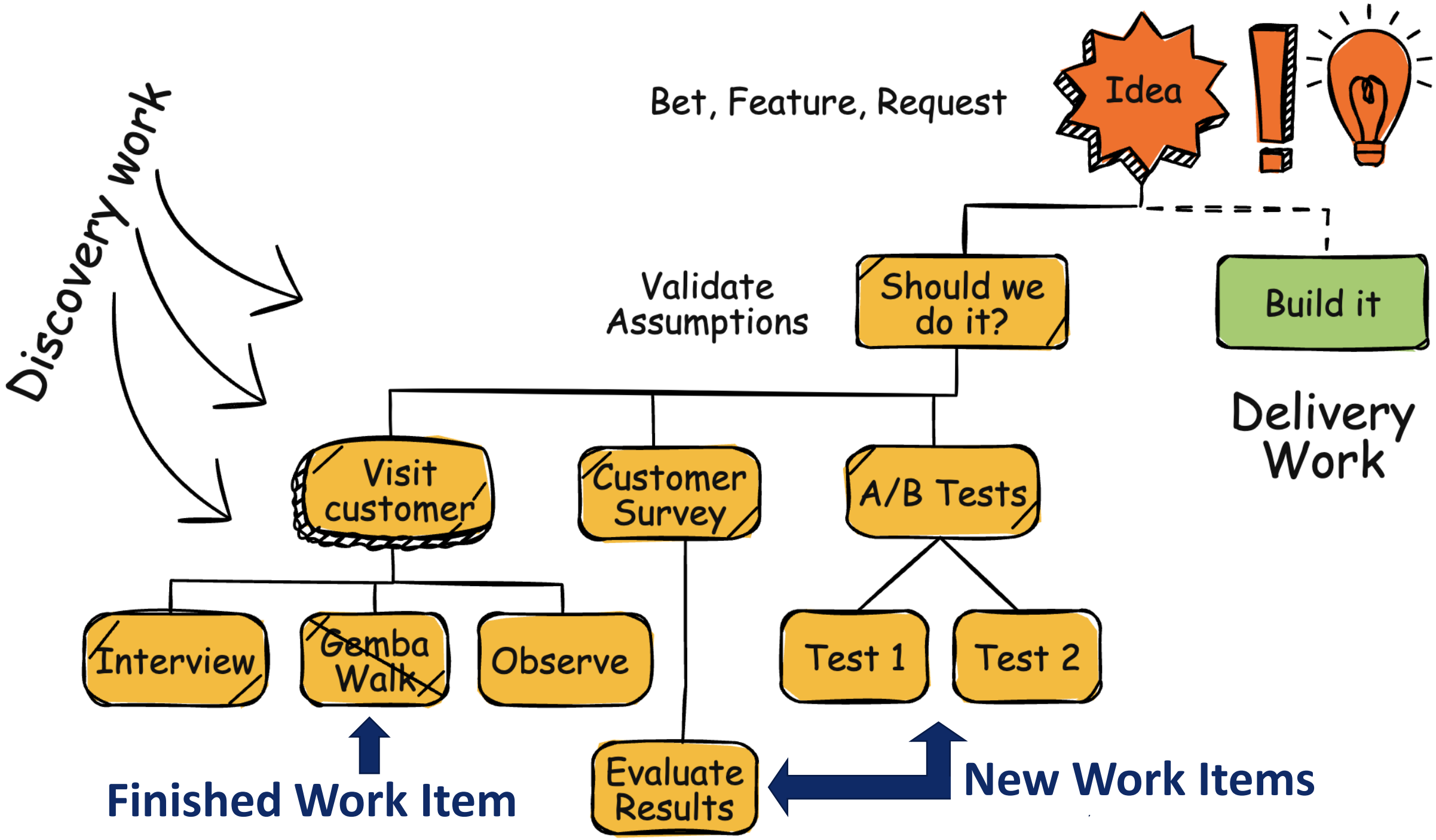
Validate Assumptions

Should we do it?

Build it


Bet, Feature, Request
Idea
Validate Assumptions
Should we do it?
Build it
Visit customer
Customer Survey
A/B Test


Discovery work
Bet, Feature, Request
Idea
Validate Assumptions
Should we do it?
Build it
Delivery Work
Visit customer
Customer Survey
A/B Tests
Interview
Gemba Walk
Observe
Let's Start Work!


Discovery work
Bet, Feature, Request
Idea
Validate Assumptions
Should we do it?
Build it
Delivery Work
Visit customer
Customer Survey
A/B Tests
Interview
Gemba Walk
Observe
Test 1
Test 2
Evaluate Results
Finished Work Item
New Work Items

Summary of Benefits – Discovery Trees

- Enables emergence
- Encourages Lean with Just enough, Just in time
- All work items have context
- Not stuck in agile naming conventions
- Shows progress at a glance
- Can be used for rapid forecasting
- Supports rich meta data and linking back to more detail when needed
- Enabler for dynamic reteaming
- Can represent both discovery and delivery work – together or separate

Visualizing Complexity

Just
Enough

A tree structure makes it easy to communicate the big picture while also diving into the details when needed.

Teresa Torres

Thinking Big
While Working Small

thinking **Big** while working **small** =

Product Mapping (big) +

Discovery Trees (small)

Came from FAST Agile(Fluid Scaling Technology for Agile)

Discovery Work vs Delivery Work

Discovery Work Item Examples

- Direct customer interactions
- Customer labs
- A/B testing
- Getting out of the office
- Analyzing results
- Making decisions
- Testing assumptions

Delivery Work Item Examples

- Design / Architect
- Engineering
- Testing quality
- Testing functionality
- Deployment/delivery
- Monitoring, inc. Synthetic
- Stress/load/defect testing

Discovery Trees

Discovery and delivery work can both be represented – and happening at the same time in the same tree!

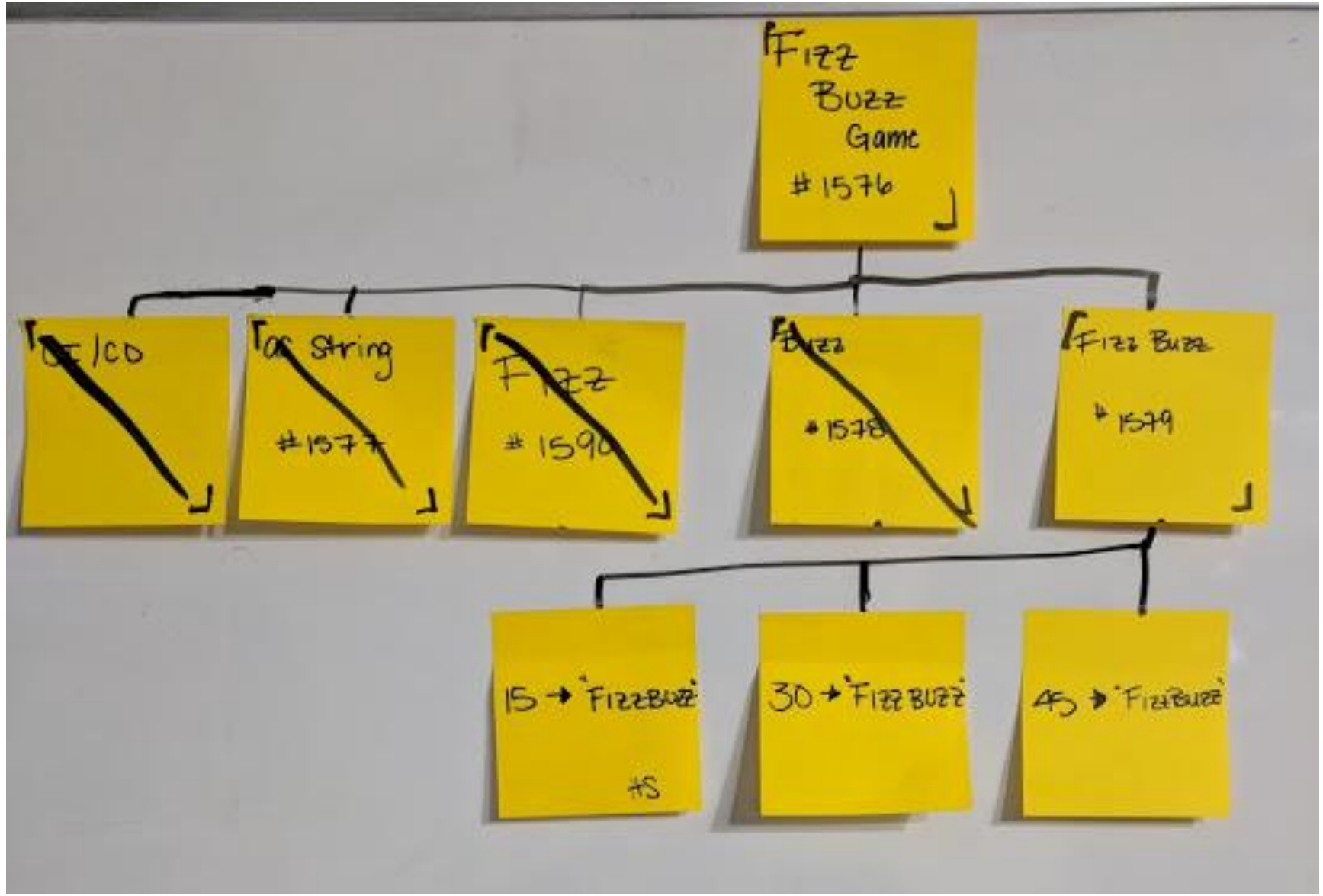
Real World Examples

Of Discovery Trees

DEPLOY TO WINDOWS STORE ~~ACCT~~ ∝ #792
~~GIT REPO UWP IN USTS~~
~~CREATE PROJECT 10586 → CREATORS~~
BUILD IN USTS
RELEASE ~~to~~ USTS IN TO STORE
~~APP PROJECT~~
~~PROJECT UNIT TESTS W/TEST~~
~~UI TESTS PROJECT W/TEST~~
CREDENTIALS TO WIN STORE ACCT
~~FIX BUILD ERROR~~
~~UPDATE BUILD HOSTED FOR USWEST~~
RUN UNIT TESTS
RUN UI UNIT TESTS
SIGNING CERT
CHECKMARX
GET
SECURELY STORE
~~EMAIL ARHIE FOR "HOW"~~
~~EMAILED OPS FOR SUGGESTIONS~~

CHATBOT FEATURE BOARD
Data
APIs
BoT Plat
Android Client
CI CD
Postify Internal Calls
Web Socket Proxys
UID PSP Calls
Versioning Strategy
Productify Code
Scrub Proprietary Code
SBT up VSTS Release To Resource File
mom Edge
mom PSP
mom SSO
BUGS GO Here
R-Next
?
Data Research

UWP FEATURE BOARD
NEXT UP
MISC TASKS
MRE PINNING
PROD BUILD #
ADOBE ANALYTICS #
LOGIN PAGE CLEAR STATIC CACHES
TO DO THE TODO'S
Security Pass
LIBRARY PULLED IN
CONFIG THE CONFIG
MIM VIA CONFIG
Cost and Savings #886
MRE
#885 ID CARD
Claims #888
Cost & Savings Blade
Network Backend
Create UI
CODE
Automation
Dashboard Widget
NO CLAIMS
REFACTORS
Memory issue
Navigation
Create UI
Automation
CODE
Back Stack
All Claims Page
Now to defend All claims
Now to specific Claim on All claims
CODE
UI
Automation
Claim Details Blade
Service Details Blade
UI
Automation
CODE
UI
Automation
CODE
ACTIVITY TIMER
MOBILE CONFIG FILE
LOGIN SCREEN
Disable login when no internet connection
CERT PIN SSO
FIND CARE
FIND CARE WIDGET FOCUS INTO LOCATION BUG
Find care widget Black Headed
DEPLOY TO WINDOWS STORE #792
CODED UI RUN ALWAYS WHEN NOT REMOTED
UI Tweaks
BLOCKED
LIFEWISE MEMBER ?
Remember User Name does not work on Premera Machines



Attributions to Quinn Gil - https://quinngil.com/2020/02/03/feature-mapping/

Article

Discovery Tree - A Simple and Natural Way of Story Creation, Project Tracking and Shared Ownership

Quinn Gil, Ron Quartel
03 Feb 2020 • 18 min read

Attributions to Quinn Gil - https://quinngil.com/2020/02/03/feature-mapping/

Retrospective

What did you learn?

Discovery

Discovery feeds delivery, and delivery feeds discovery. They aren't two distinct phases. You can't have one without the other.

- Teresa Torres

Delivery

Benefits, Summary, Closing

Summary of Benefits
Tooling
Similar Concepts
Finding Out More - References
Staying in Touch - Links

Woody Zuill
@WoodyZuill

As I say: "it's in the doing of the work that we discover the work we must do", (or the point where balance is achieved)

6:23 PM · Jul 15, 2016 · Twitter for iPhone

Summary of Benefits

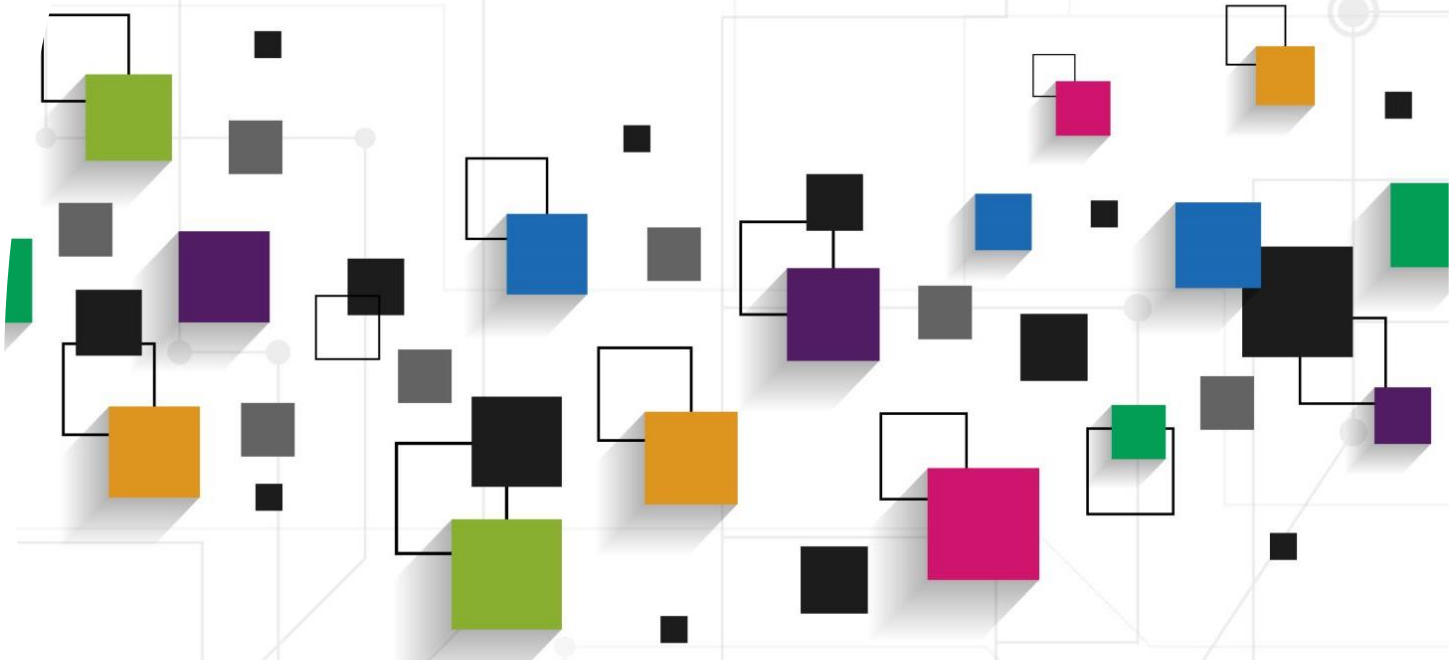
Tool for PM and Devs	Supports Dynamic Reteaming (and FAST Agile)	Lightweight	Highly adaptive – supports new discoveries	Discovery and Delivery supporting (not just Delivery)	All the information one needs can be found at a glance
Work items always have context	A tool to aid in splitting down work	Supports lightweight forecasting	Visibility into what is started, not started, done, blocked etc.	Supports any amount of meta information needed (colors, markings etc.)	Maintains the complex nature of work (to n dimension)
Bypasses the confusion that agile industry have made around naming conventions of work items	Helps with flow	Supports and encourages “Just enough and Just In Time” view of work, and breaking down of work	Simple	Easy to use	Map of Maps
Enabler for Self-organization	Enabler for Thinking Big while Working Small	Information Radiator	No need for velocity or story pointing	Acceptance criteria only when useful/needed	Can see the wood from the trees (solves ALM tooling issues)

Tooling

Are there tools available?

What Tools are Available?

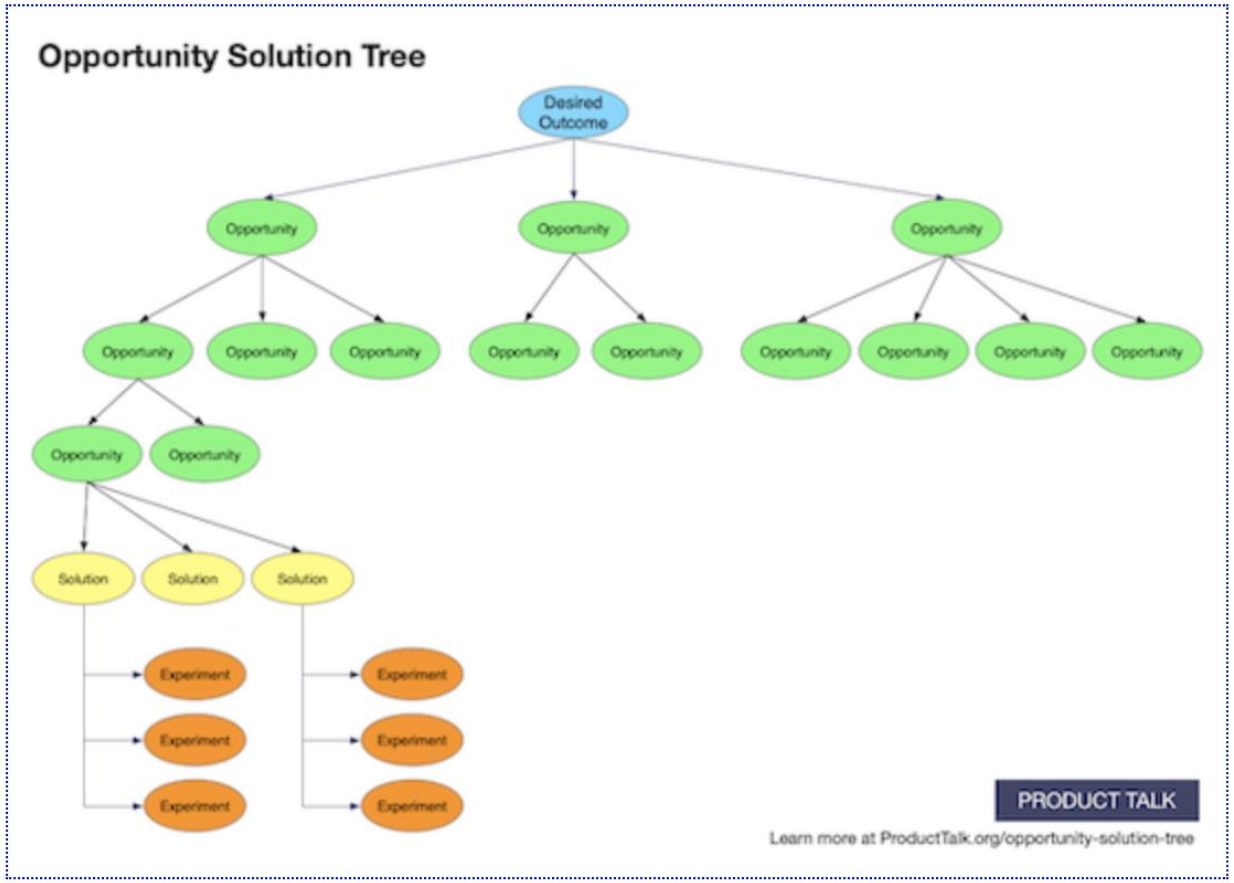
- Mind mapping tools
- Miro, Mural, others of this ilk
- If the number of dimensions you need matches the number made available by your ALM tool, you can use that
- Post it notes



Similar Concepts

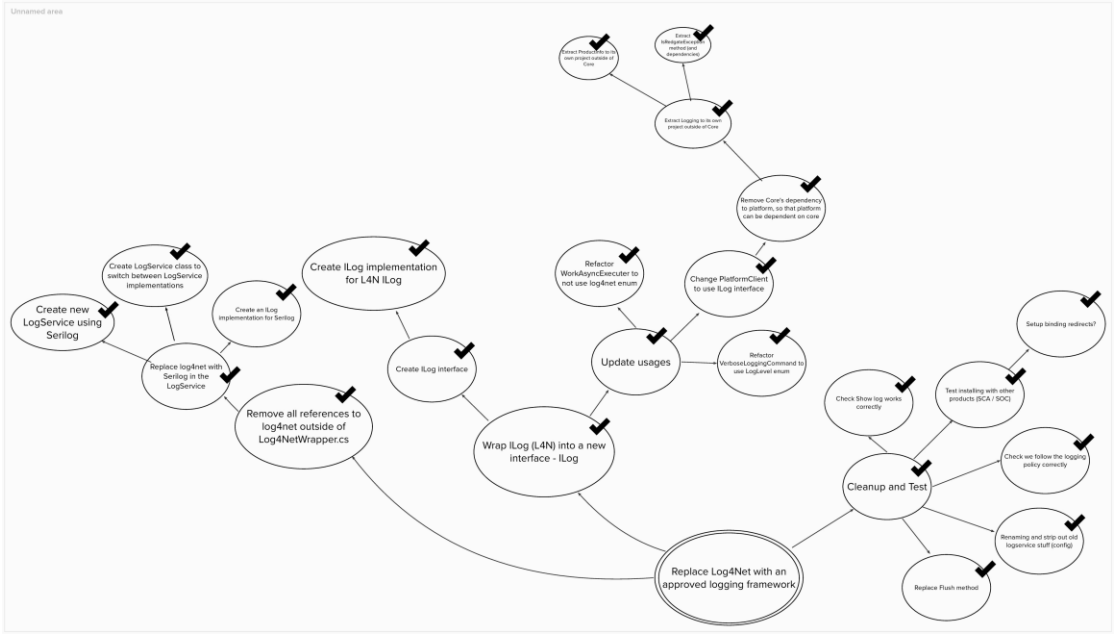
Opportunity Mapping (Teresa Torres)
The Mikado Method

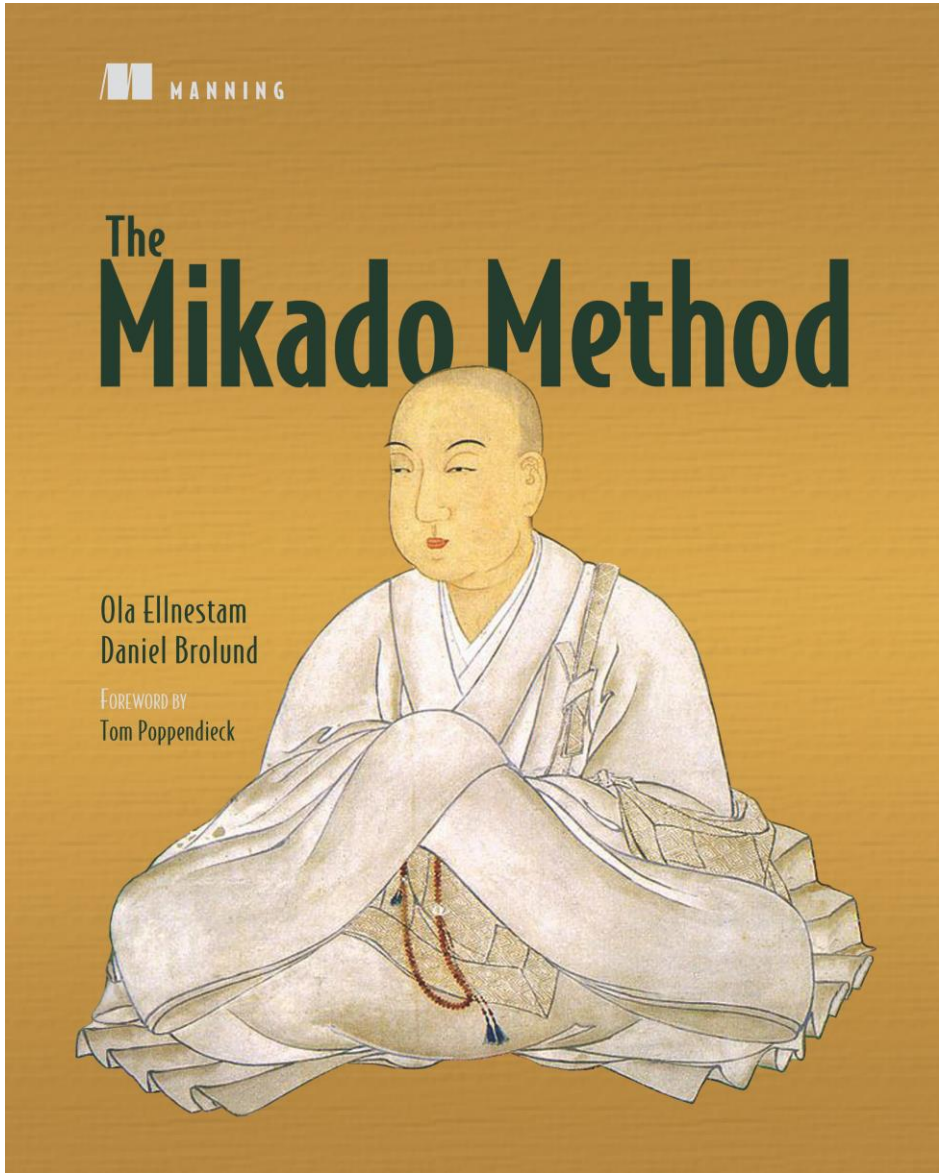
Opportunity Solution Trees: Visualize Your Thinking



Opportunity solution trees help you visualize your thinking.

Teresa Torres is an internationally acclaimed author, speaker, and coach. She teaches a structured and sustainable approach to continuous discovery that helps product teams infuse their daily product decisions with customer input. She's coached hundreds of teams at companies of all sizes, from early-stage start-ups to global enterprises, in a variety of industries. She has taught over 7,000 product people discovery skills through the Product Talk


Unnamed area
Create new LogService using Serilog
Create LogService class to switch between LogService implementations
Replace log4net with Serilog in the LogService
Create an ILog implementation for Serilog
Remove all references to log4net outside of Log4NetWrapper.cs
Create ILog implementation for L4N ILog
Create ILog interface
Wrap ILog (L4N) into a new interface - ILog
Refactor WorkAsyncExecuter to not use log4net enum
Update usages
Change PlatformClient to use ILog interface
Refactor VerboseLoggingCommand to use LogLevel enum
Remove Core's dependency to platform, so that platform can be dependent on core
Extract Logging to its own project outside of Core
Extract Product Info to its own project outside of Core
Extract IdlestgateExecuter method (and dependencies)
Replace Log4Net with an approved logging framework
Cleanup and Test
Check Show log works correctly
Test installing with other products (SCA / SOC)
Setup binding redirects?
Check we follow the logging policy correctly
Renaming and strip out old logservice stuff (config)
Replace Flush method


MANNING
The Mikado Method
Ola Ellnestam
Daniel Brolund
Foreword by
Tom Poppendieck

Find Out More

Where to go from here for more information

Resources to find out more

Untangling Confusion with a Discovery Tree ~ Jon Reid
https://www.youtube.com/watch?v=dt7JVmpw-DE&t=1300s

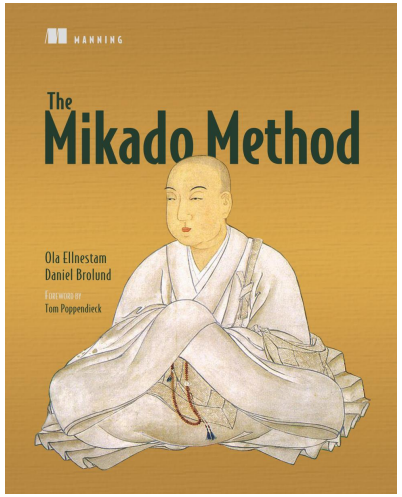
Discovery Trees - A Simple and Natural Way of Story Creation, Project Tracking and Shared Ownership ~ Quinn Gil

https://quinngil.com/2020/02/03/feature-mapping/

Working With Discovery Trees ~ Steve Kuo

https://www.industriallogic.com/blog/discovery-trees/

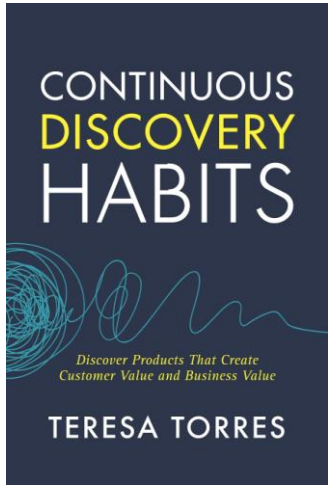
References and Resources



The Mikado Method
by Olla Ellnestan
and Daniel Brolund

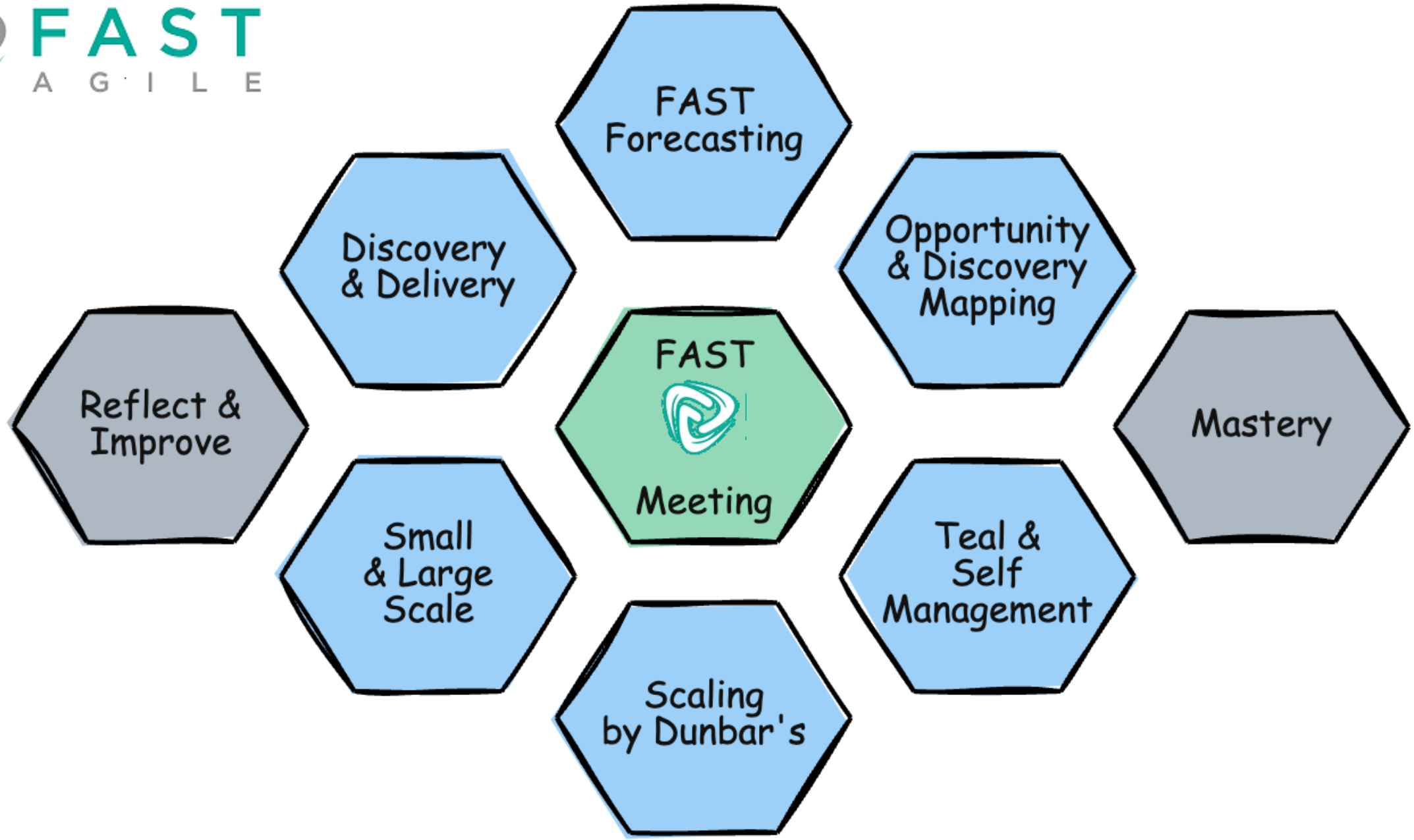
FAST AGILE

FAST Agile
by Ron Quartel
https://fluid.scaling.tech

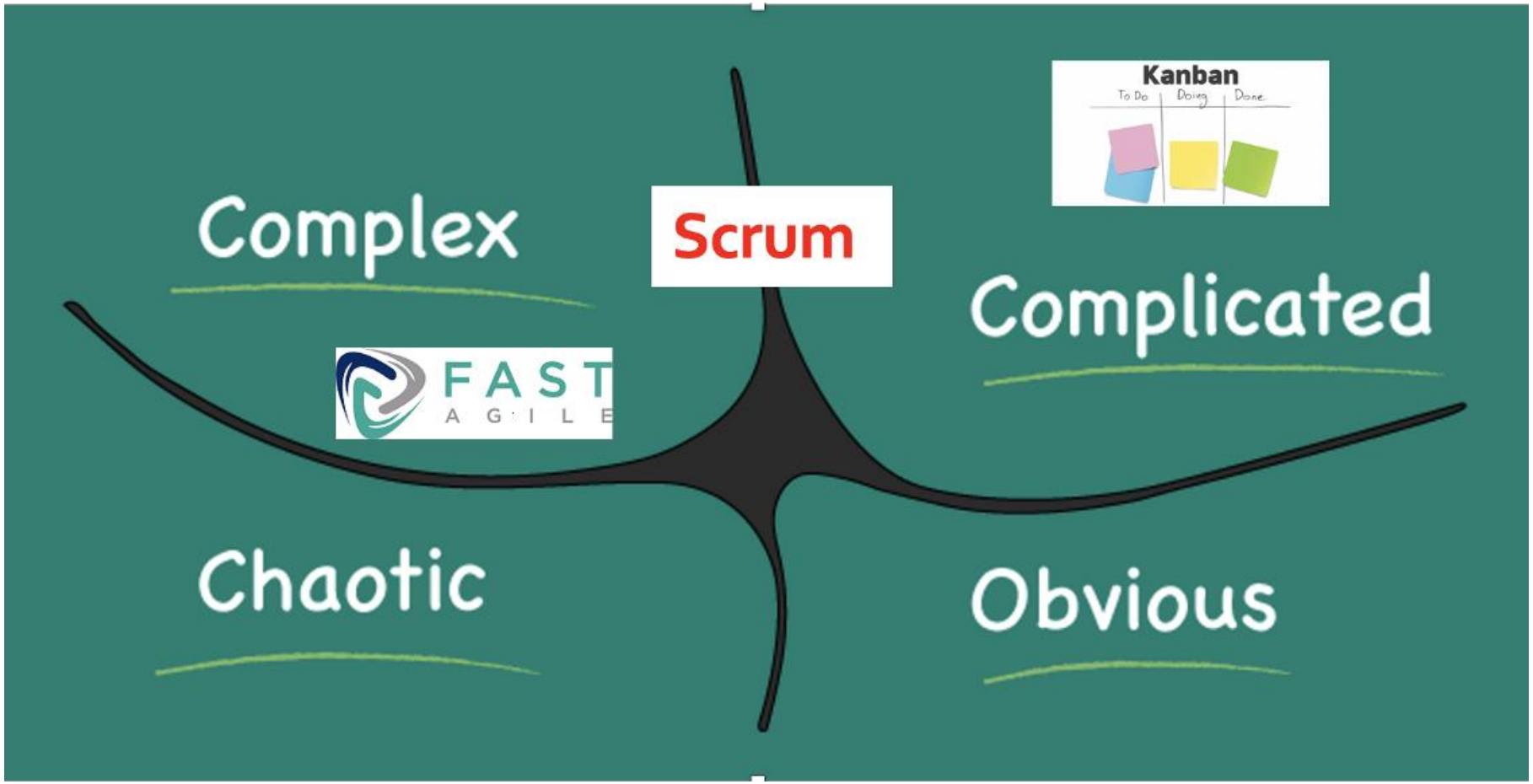


Continuous Discovery Habits
by Teresa Torres

FAST AGILE


FAST Forecasting
Discovery & Delivery
Opportunity & Discovery Mapping
Reflect & Improve
FAST
Meeting
Mastery
Small & Large Scale
Teal & Self Management
Scaling by Dunbar's

FAST is the First Purely Complex Agile Method!



Staying in Touch

Ron and Paige want to stay in touch with you

Stay in touch

Ron Quartel

LinkedIn –
https://linkedin.com/rquartel

Twitter –
@ronquartel
@fstagile

Blog –
https://medium.com/@ronquartel

Paige Watson

LinkedIn –
https://www.linkedin.com/in/paige-watson-b817564/

Twitter –
@paigeisxp