

Agile Project Management



Managing Projects under Uncertainty

Session 11

Nariman Dorafshan, 2023

Part 2.3

Kanban Method

Summary: Key Metrics to Understand System behavior

Lead Time

Delivery Rate

Flow Efficiency

Failure Demand

Initial Quality

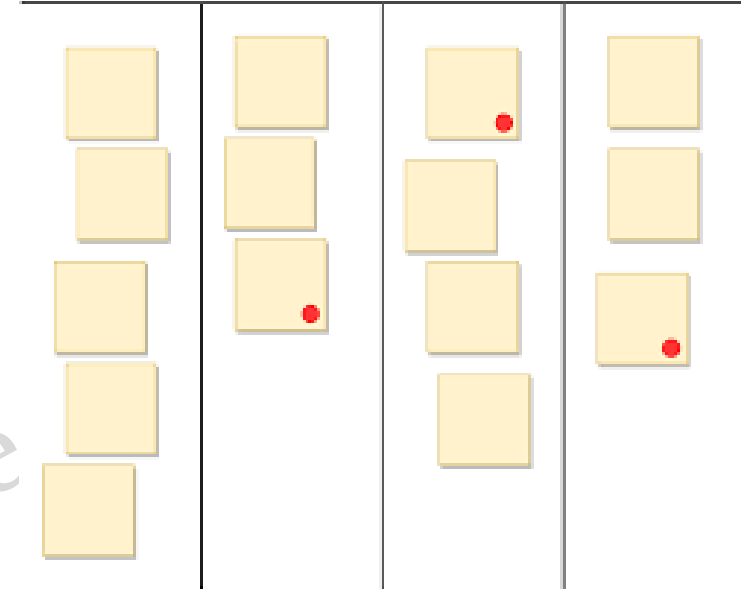
Blockers

WIP Levels

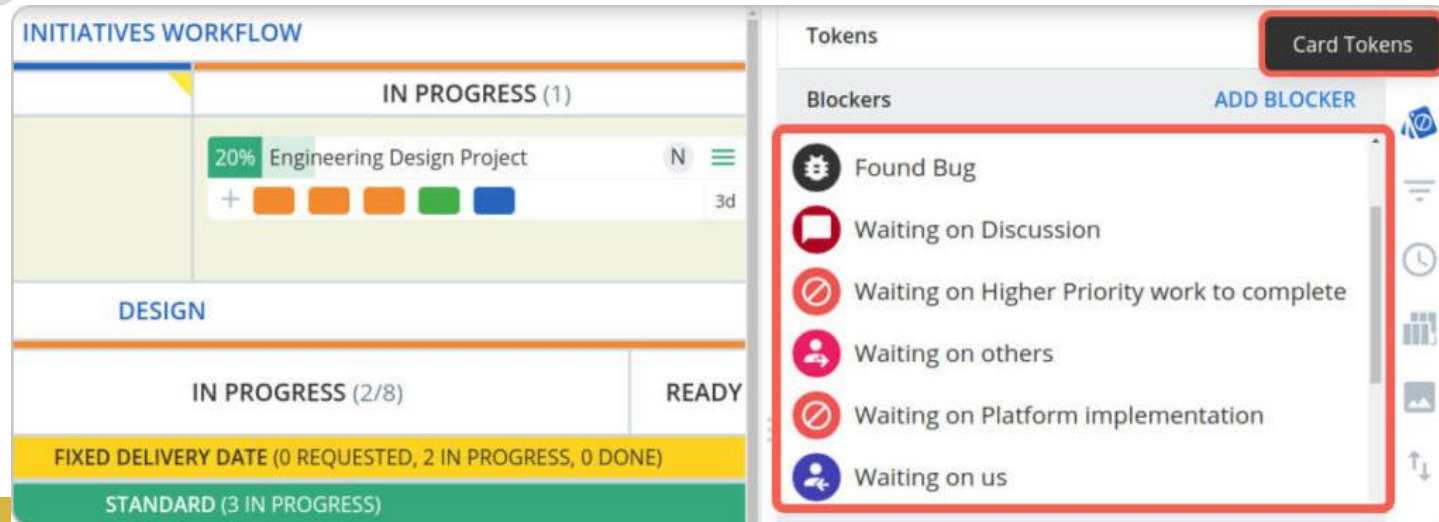
Discard / Abort
Rates

Blockers

- We can add stickies to blocked cards to show they are Blocked
- Or attach a sticky to it, describing what caused it to be blocked, and How Many Days it's been like this



- Blocker Clustering Cadence



Further Considerations

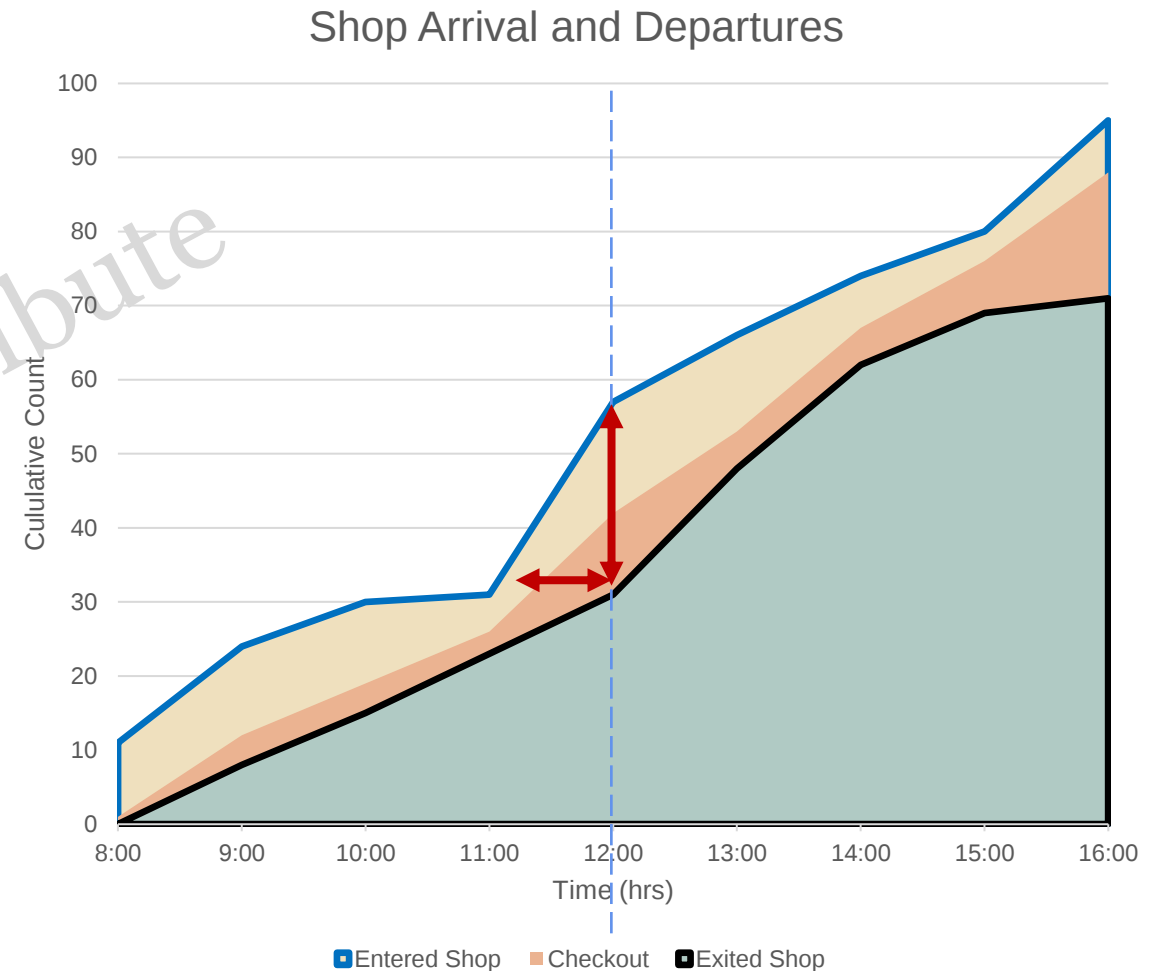
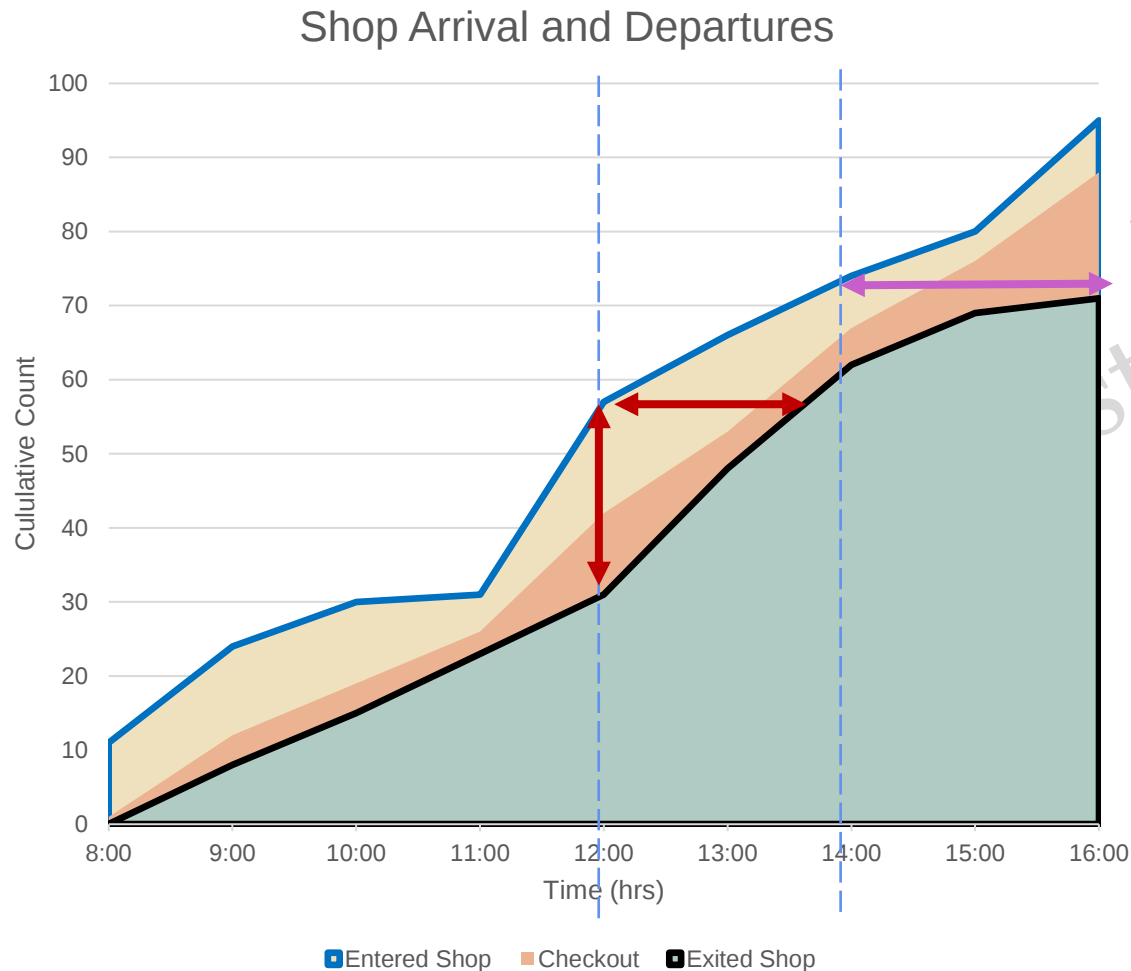
Do not Distribute

WIP Item Age

- Time passed since commitment point for work items that hasn't been done
- Monitoring different thresholds for work-stages
- Calendar Day? Working Day? Whose working day? Week? Month?

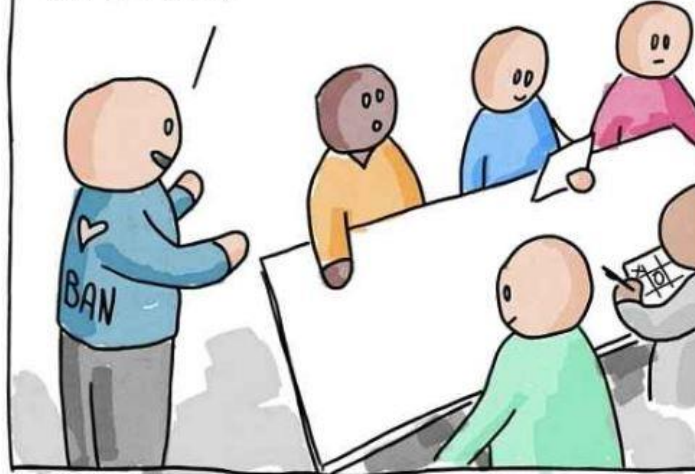
Do not Distribute

How are these two different?



Comic Agilé

So, by using Kanban principles, we can thoroughly inspect our progress and identify problems in our flow of delivery.

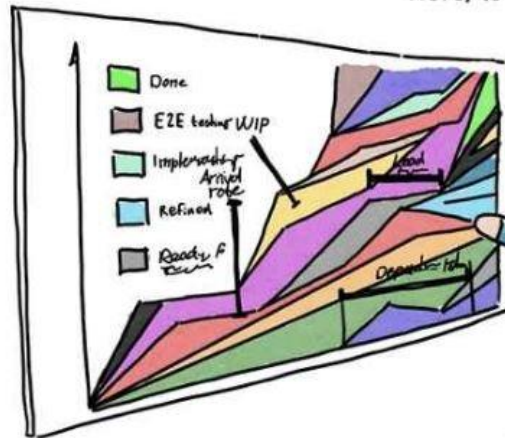


For example, with a Cumulative Flow Diagram, we can see our work-in-progress across stages and how it fluctuates over time, how the departure rate and arrival rate compare to each other, and how the increasing cycle time impacts the total lead time of our story delivery and time-to-market.



Don't be like this!

Here, let me show you



So, how're we doing?



Created by Luxshan Ratnaravi & Mikkel Noe-Nygaard

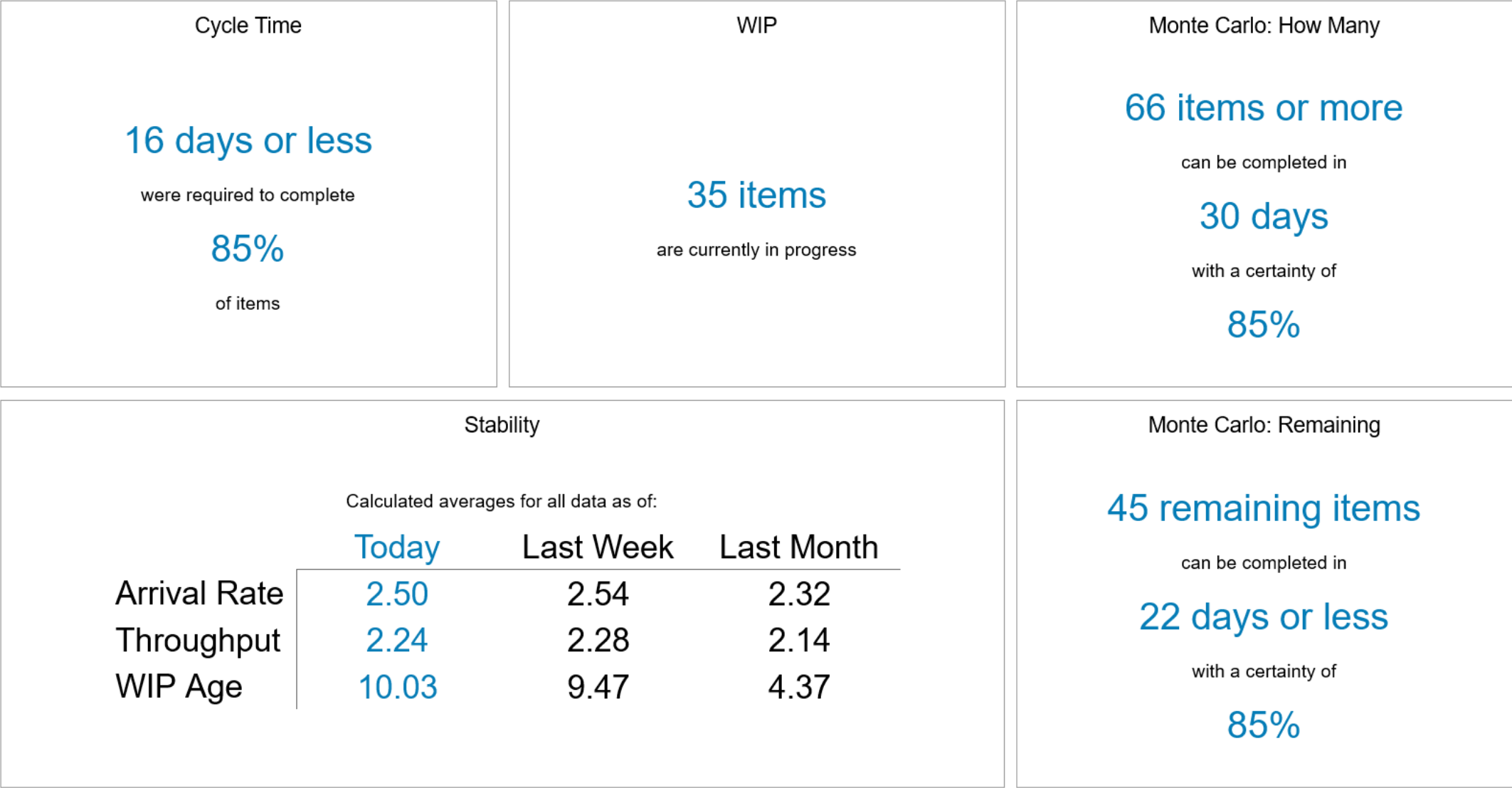
Let's look at

ActionableAgile Project Demo and its reports

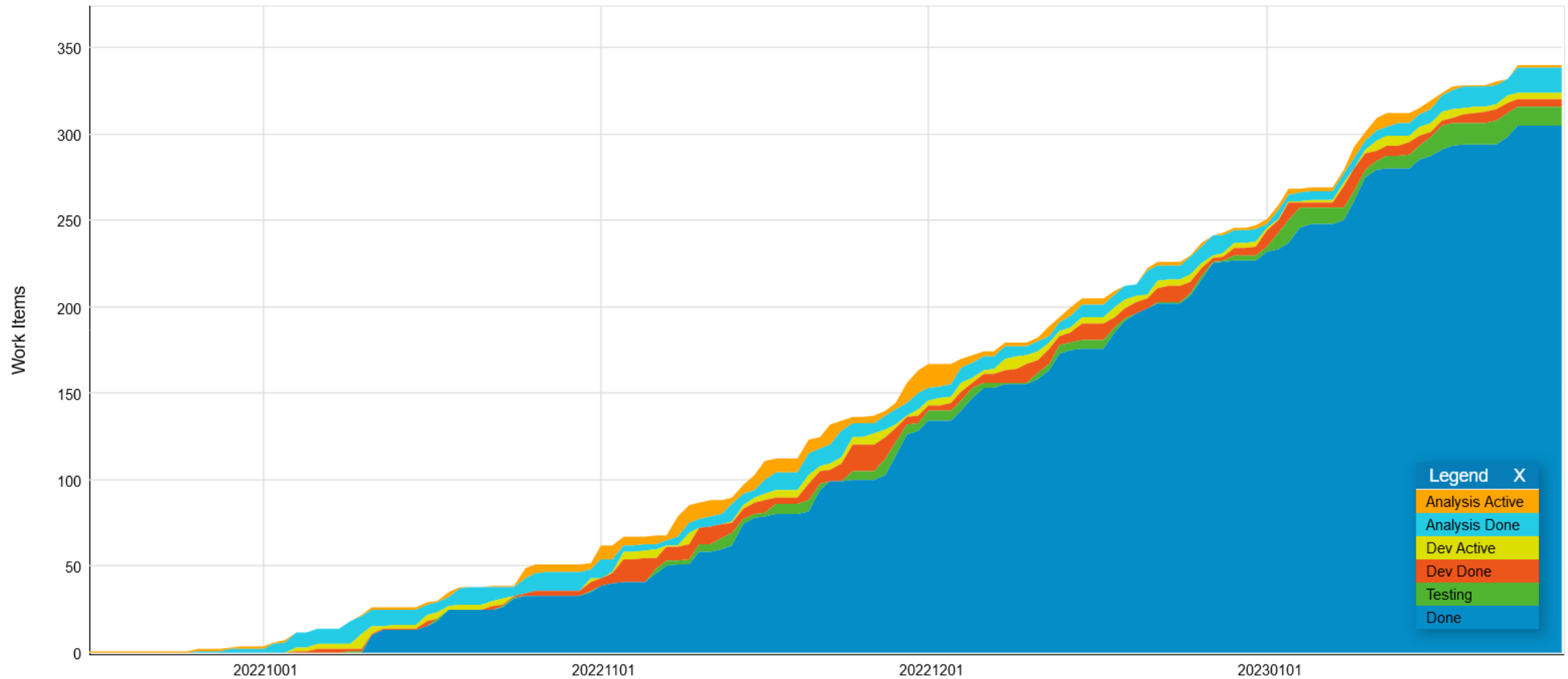
demo.actionableagile.com

Do not Distribute

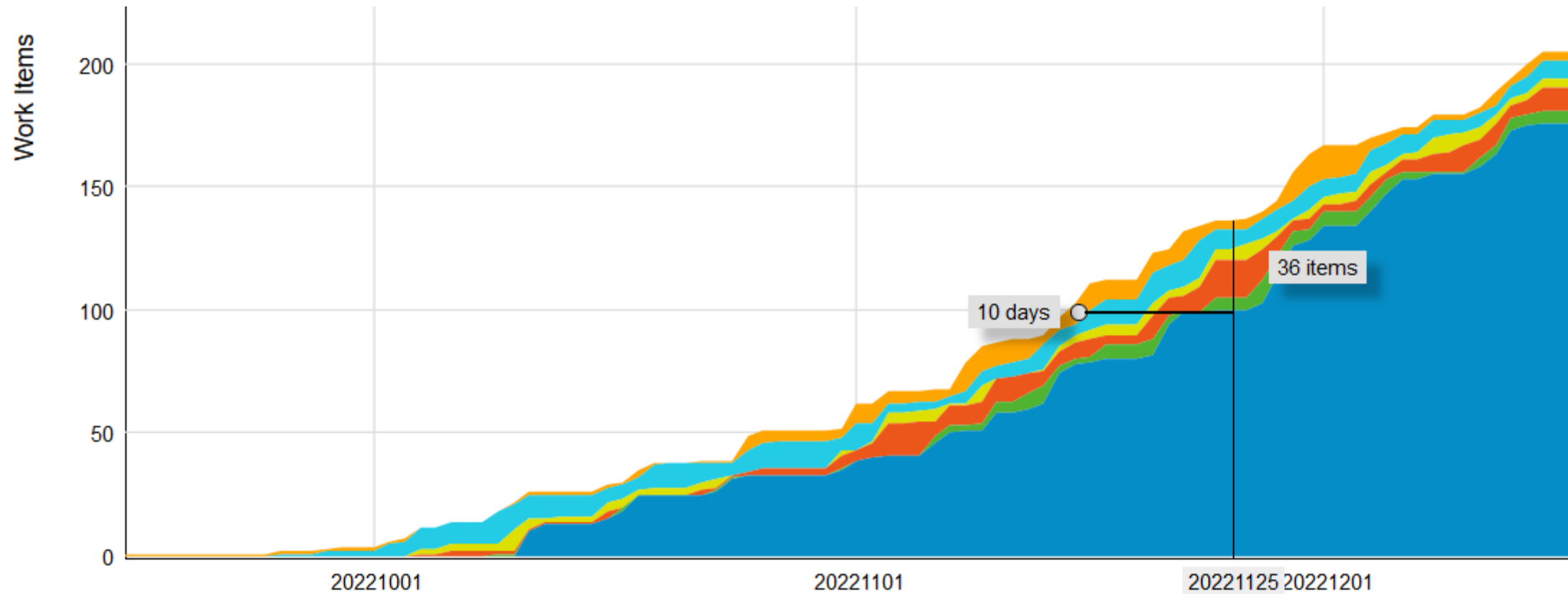
Dashboard



Cumulative Flow Diagram



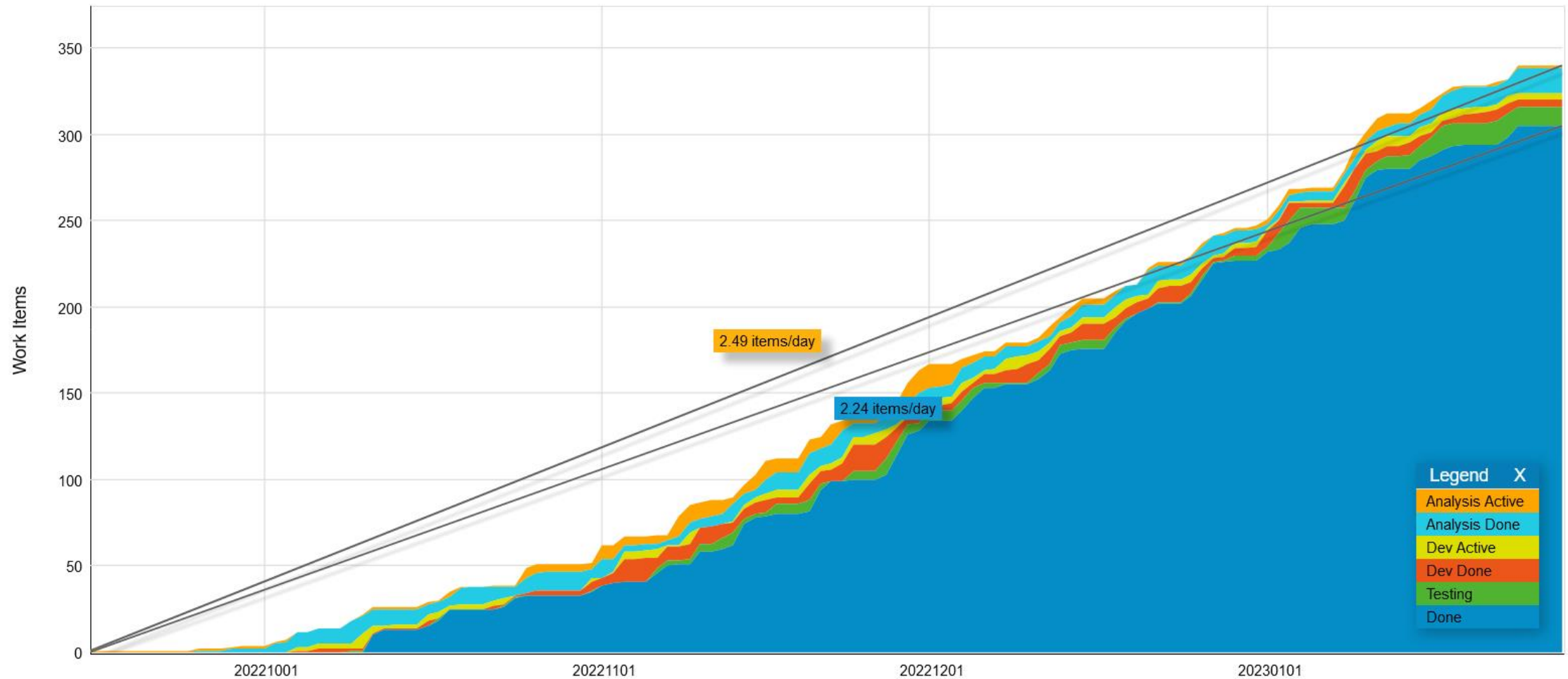
Compare how they've implemented Little's Law



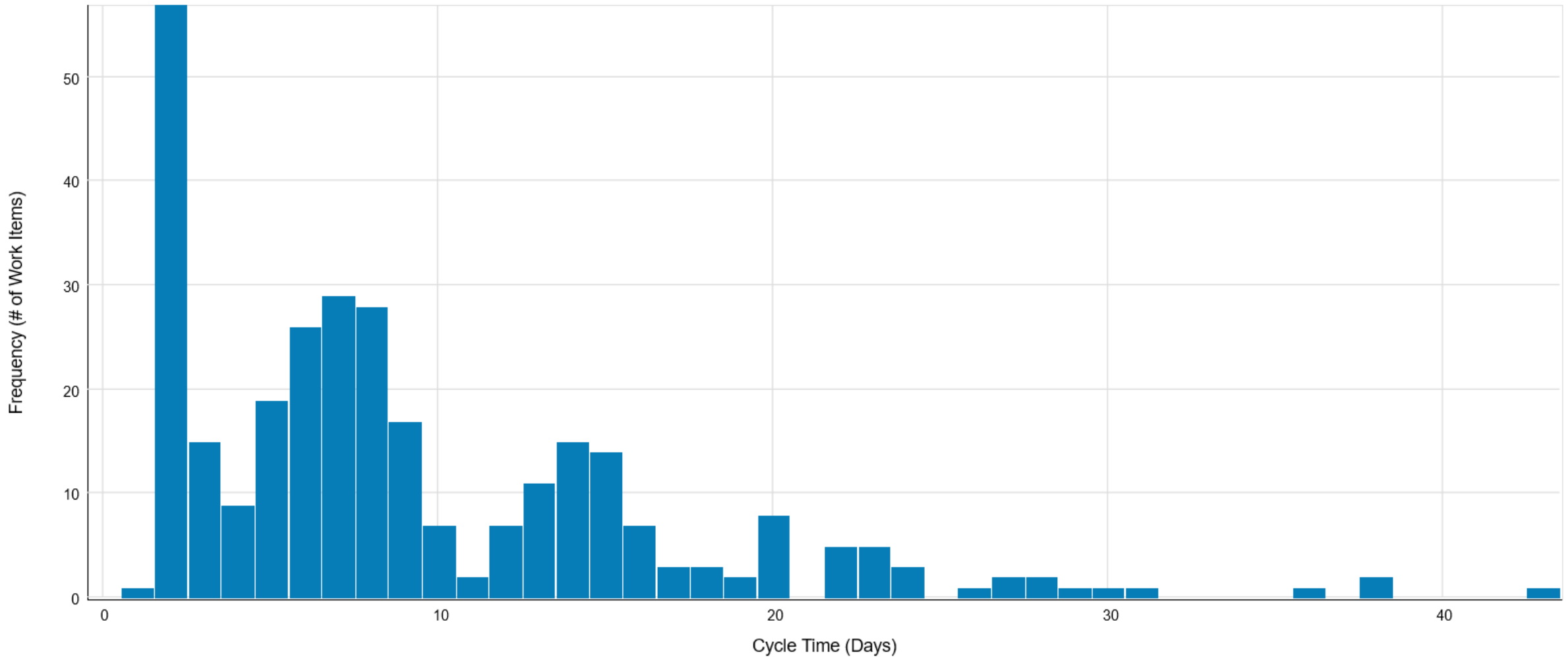
Summary Statistics

Summary Statistics				
20220915 - 20230128 (136 days)				
Note: All values are calculated averages	Arrival Rate (items/day)	Throughput (items/day)	Daily WIP (items)	Cycle Time (days)
System	2.50	2.24	21.62	9.37
Analysis Active	2.50	2.49	3.49	1.38
Analysis Done	2.49	2.38	7.11	2.68
Dev Active	2.38	2.35	2.69	1.05
Dev Done	2.35	2.32	4.76	1.92
Testing	2.32	2.24	3.57	1.18
Done	2.24	----	----	----

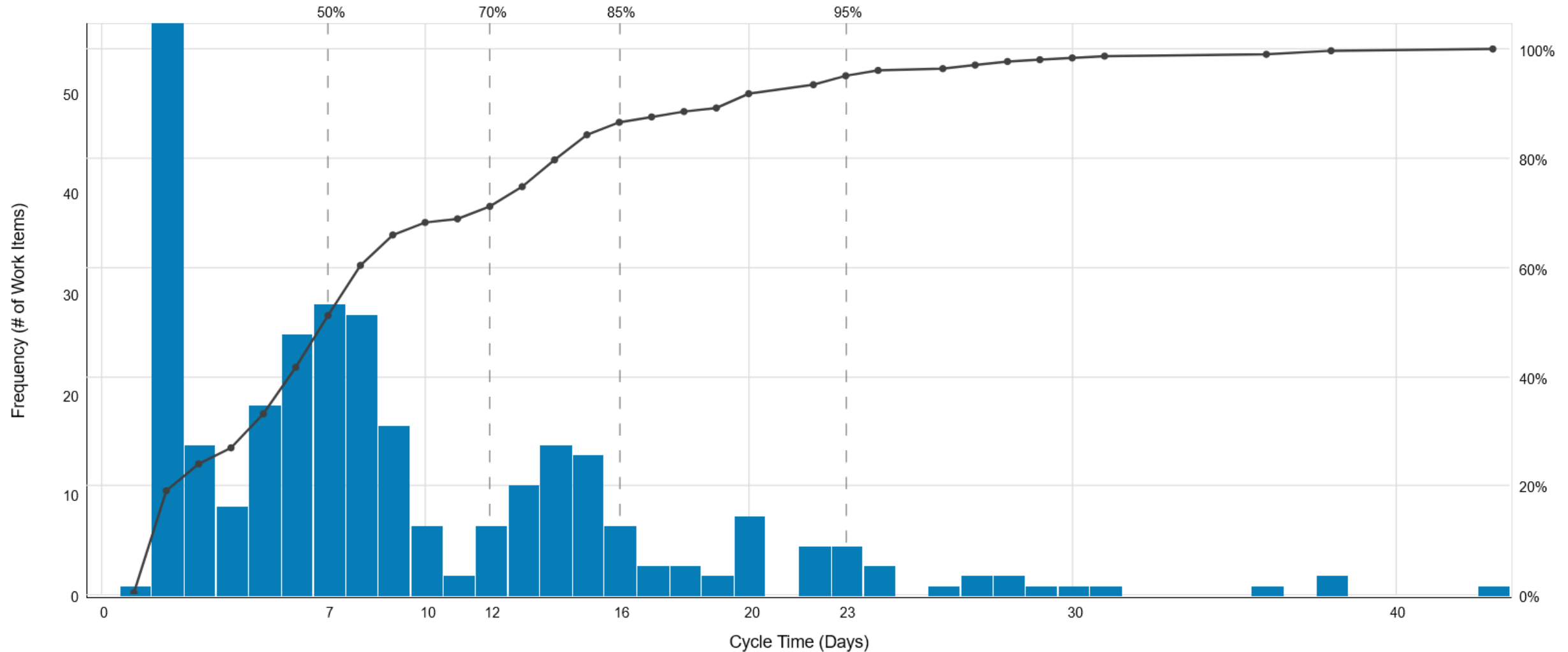
Arrival and Departure Rates



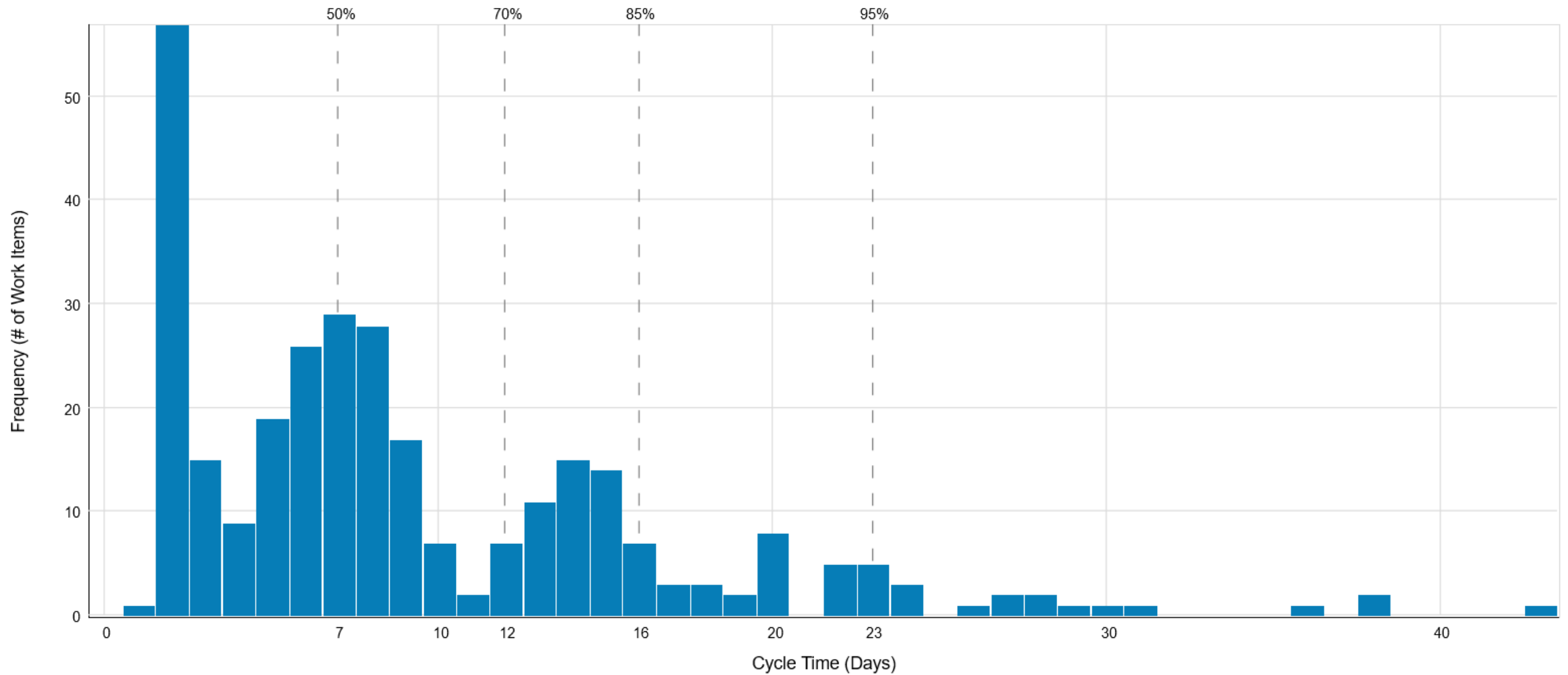
Cycle Time Histogram



Cycle Time Histogram with Cumulative Values



Cycle Time Histogram with Confidence Levels

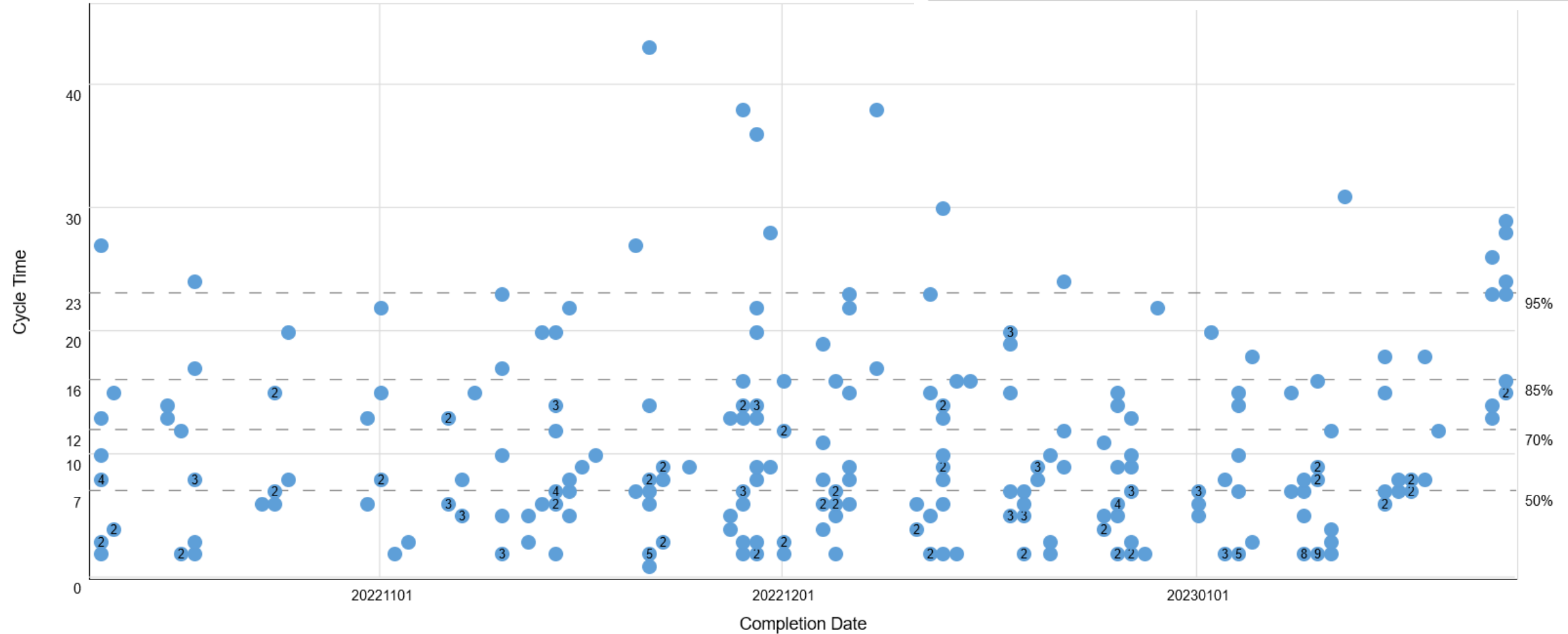


Cycle Time Scatterplot

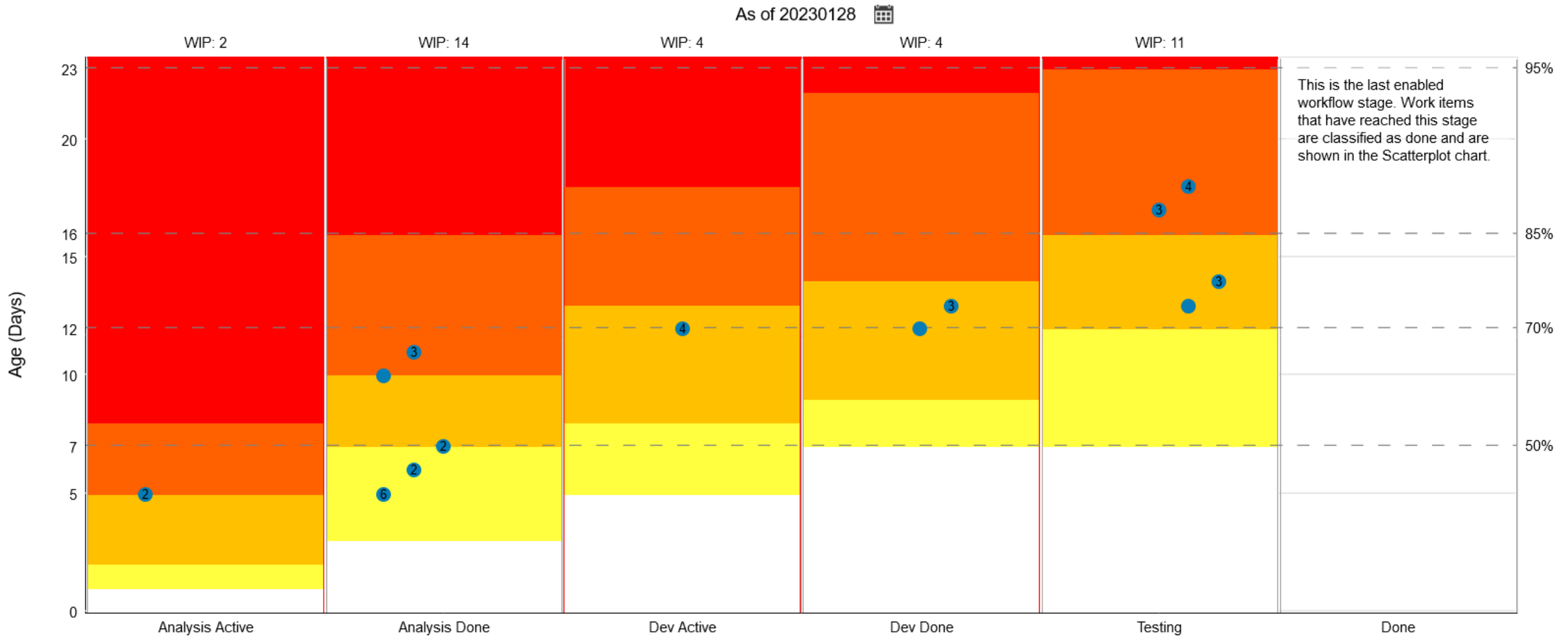
305 Work Items Completed

20221011 - 20230124 (106 days)

Percentile Range	0-50%	50-70%	70-85%	85-95%	95+%
Work Item Count	152	61	46	30	16
Work Item Total	152	213	259	289	305
Cycle Time (Days)	7	12	16	23	43



WIP Item Age

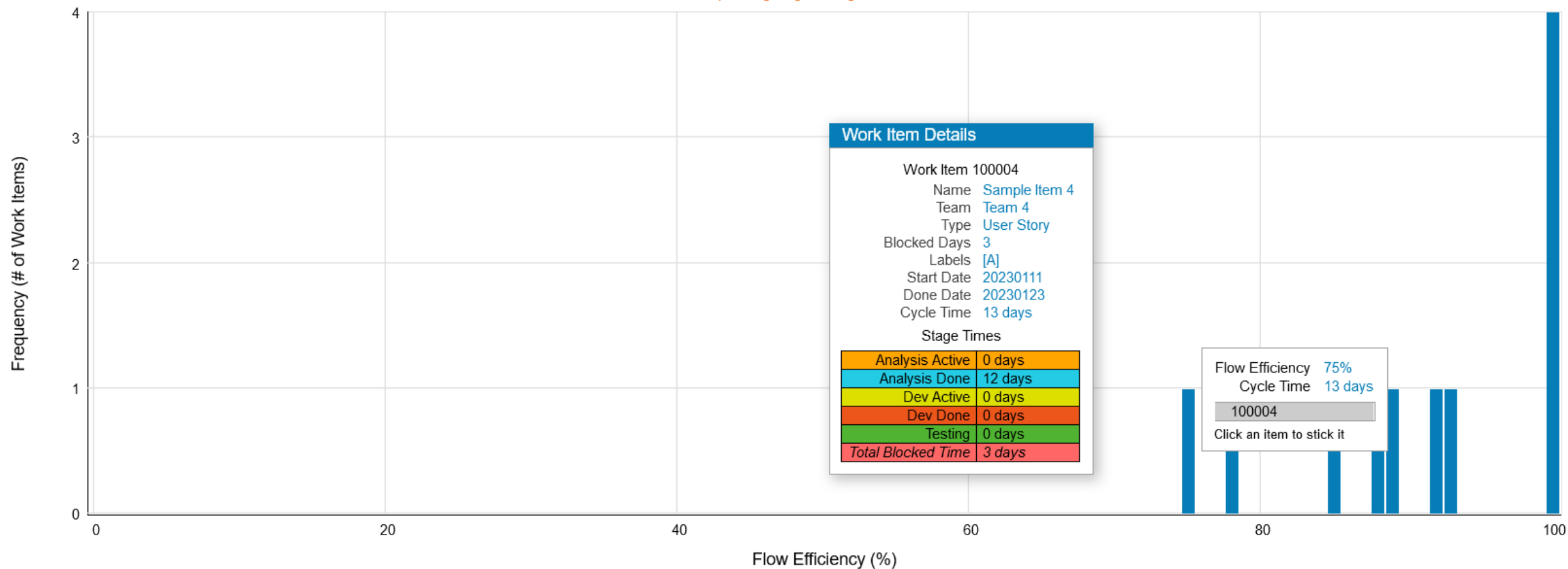


Flow Efficiency

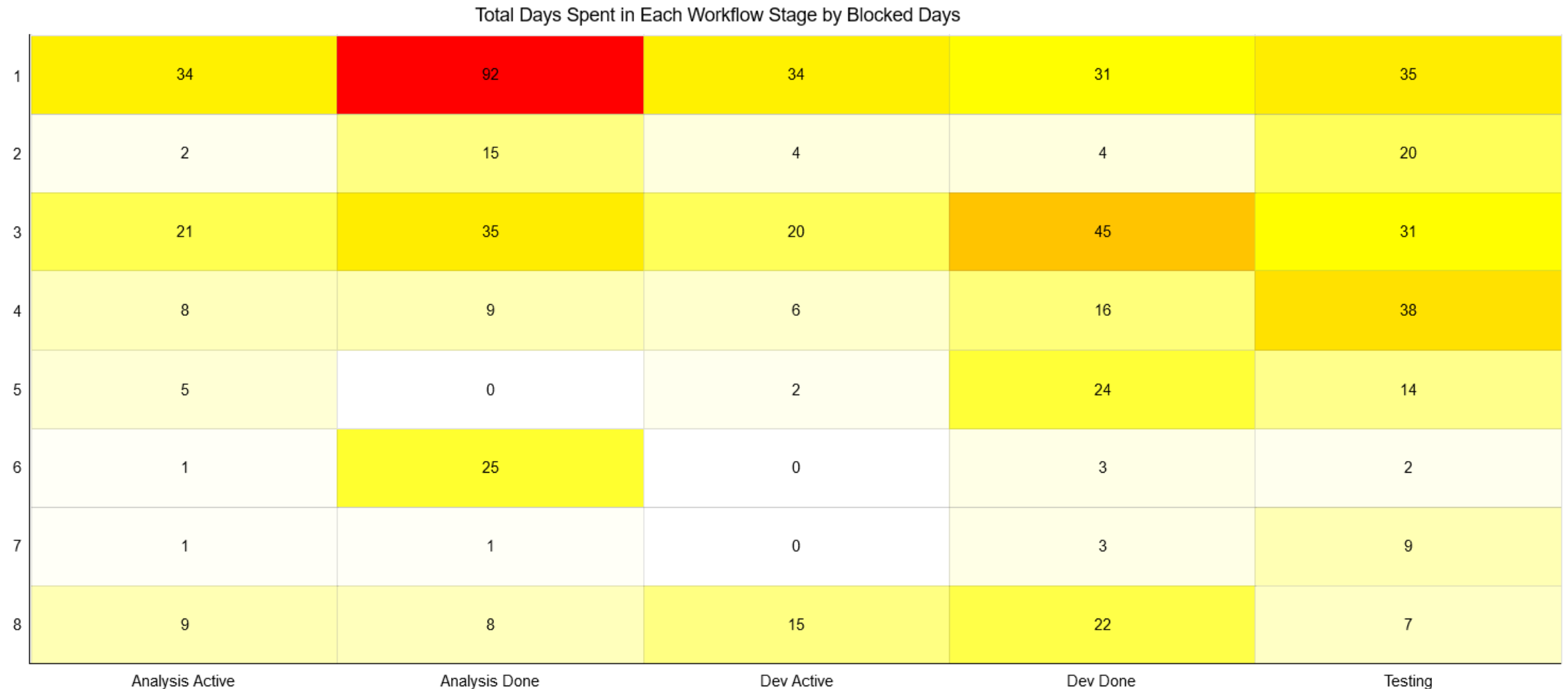
90.7%*

Overall Average Flow Efficiency

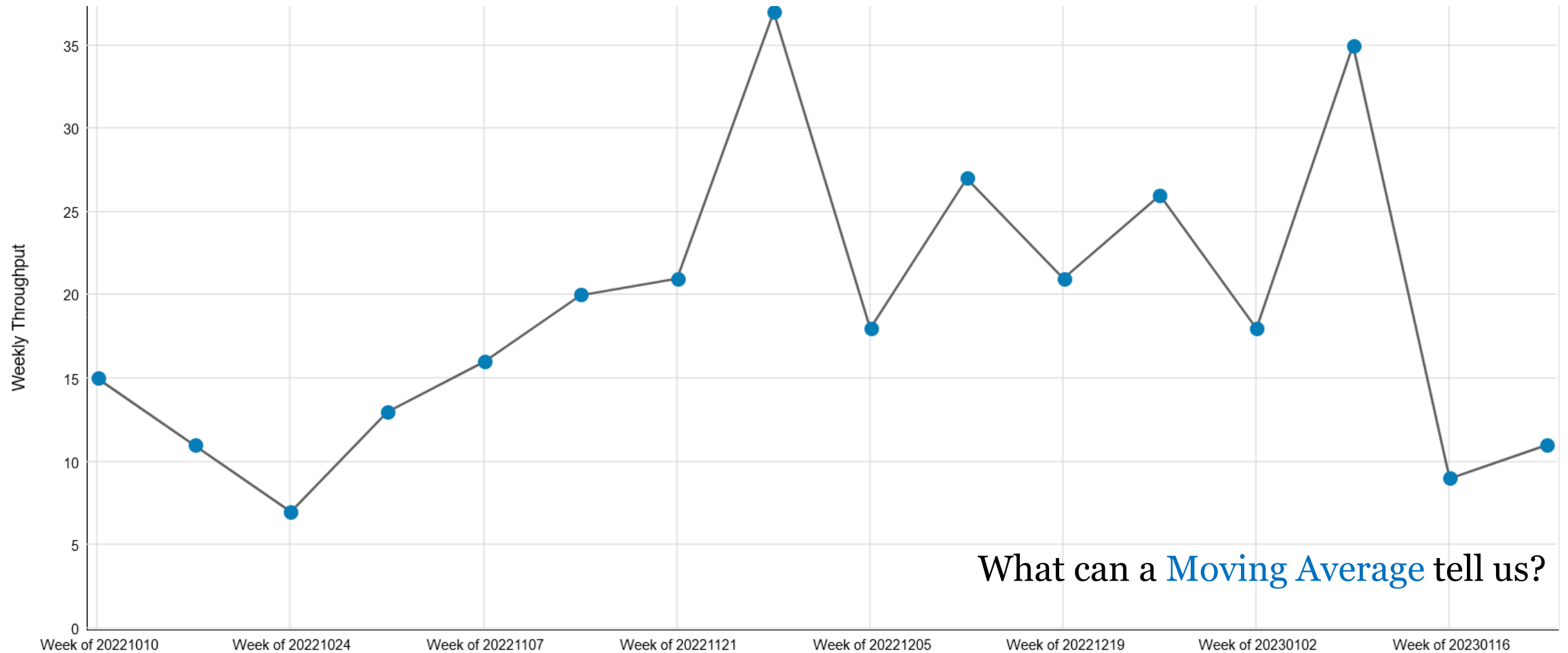
*No queueing stages designated



Heat Map – Blocked Days by Work Stages

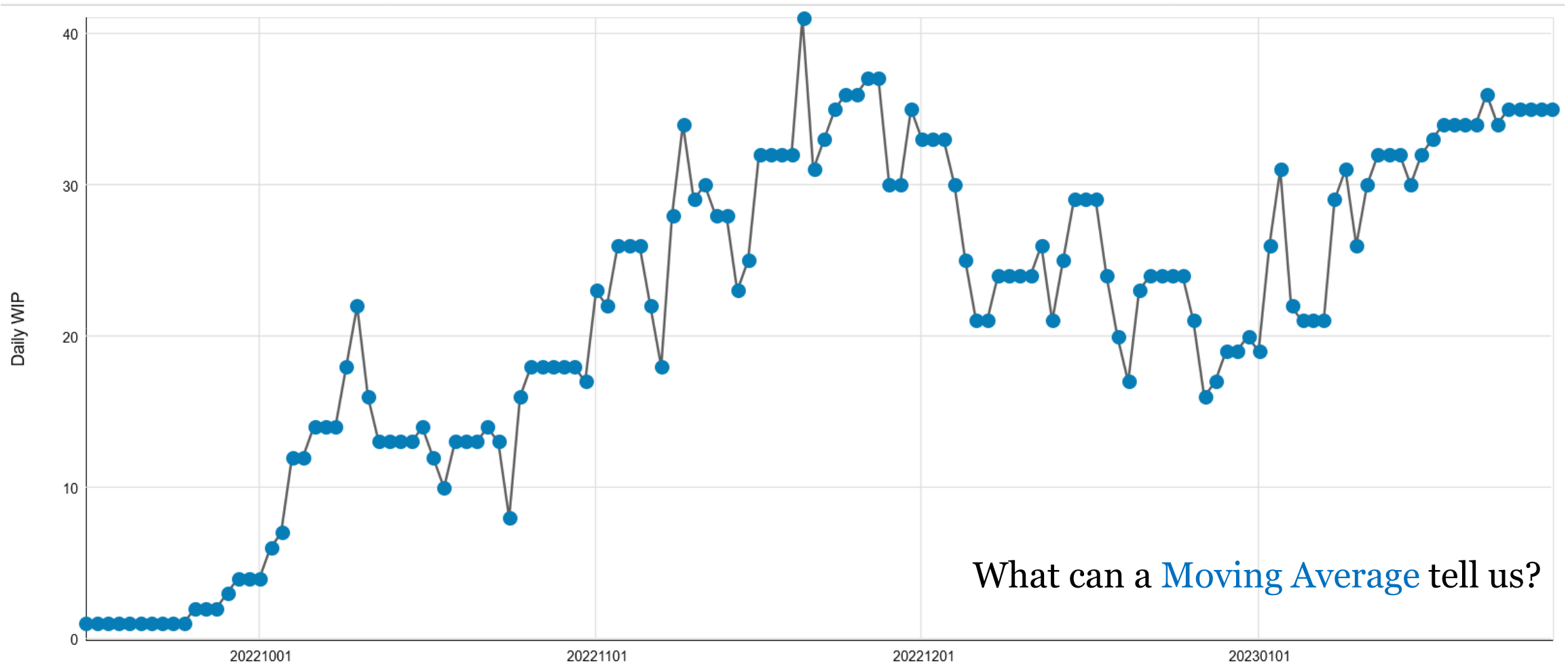


Throughput Run Chart – Weekly



What can a [Moving Average](#) tell us?

WIP Run Chart – Daily WIP



Make Policies Explicit



A process expressed as workflow and policies

- Creates constraints in action,
- Is empowering within constraints,
- And results in emergent characteristics than can be tunes by experiment.

Do not Distribute

Classes of Services (CoS)

- Defines how categories of items are treated in a system, a set of policies is defined that apply to certain work type(s)
- Recognize different kinds of customer expectation
- Treat of work items in a differentiated manner to better manage lead times and risks – be able to respond at reasonable cost
- Can impact order of selection and delivery – qualitative categories before any quantitative ranking

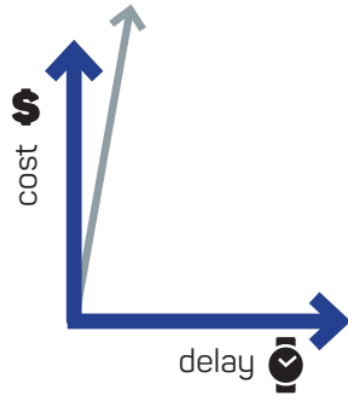
CoS - Cont

- Make explicit both internally and externally
- Can be indicated with colors, shapes, stickers, swim lanes, etc.
- ... forming the basis for reliable commitments to stakeholders!

Do not Distribute

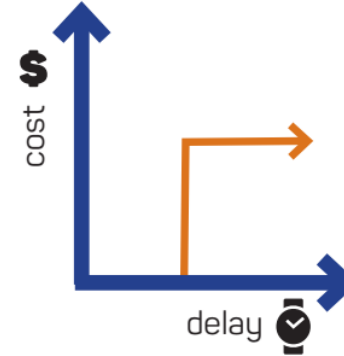
Classes of Services mapped to Cost or Impact of Delay

Called also
Service Level Expectation



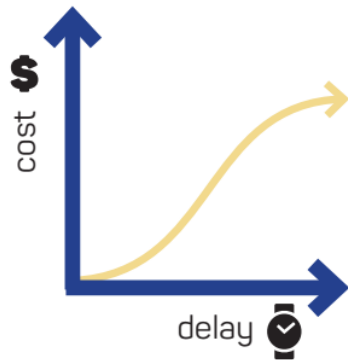
EXPEDITE

Critical and immediate cost of delay; it can exceed other kanban limit (bumps other work).



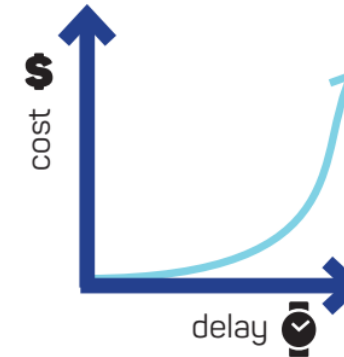
FIXED DATE

Cost of delay goes up significantly after deadline.



STANDARD

Increasing urgency, cost of delay is shallow but accelerates before levelling out.

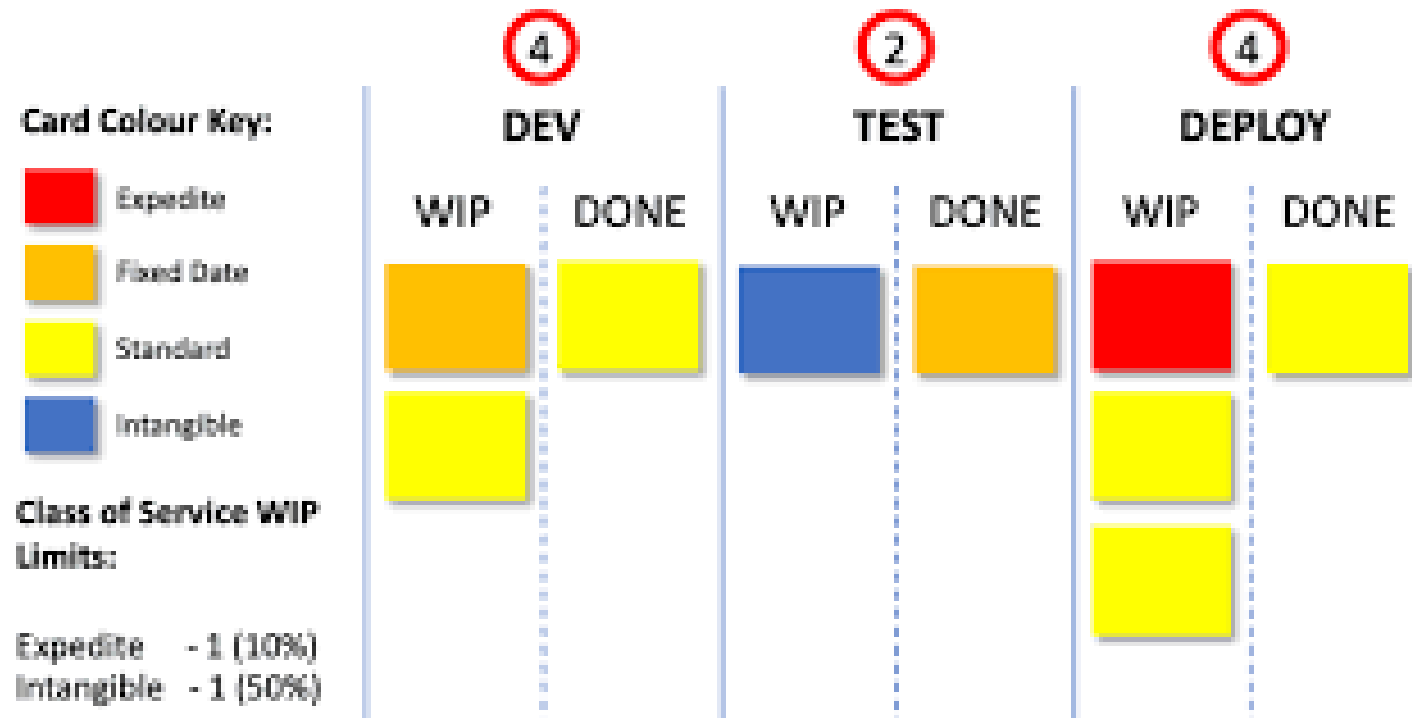


INTANGIBLE

Cost of delay may be significant but is not incurred until significantly later (if at all).

What will happen if
you don't finish the item
at that time?

CoS in different colored cards



CoS in in different Swimlanes

 **Class of service** ?

23 items total, 23 loaded items: 19 open, 4 closed | Ends in: --

Plan Details Edit

Planned Items ? [Links](#) [Snapshots](#) [Dashboard](#) [Notes](#)

☐ View As: **Kanban Board**        Exclude (1 item excluded) |

         Add Work Item

expedite
Closed Items: 2 | Open Items: 7

New 8/6 In Progress 11/8 Done 4/4

In Development In Test

237 As a Firefox user, I must have the ability
Unassigned
Blocked

239 As a mobile customer, I must
Deb
Blocked

244 As a mobile banking customer, I must
Deb
Blocked

238 As a mobile customer, I must
Marco Perez
Blocked

240 As a mobile customer, I must
Deb
Blocked

184 As a mobile customer, I must
Unassigned
Blocked

241 As an iPhone user, I want to be able to
Deb
Blocked

242 As a Windows mobile device user, I
Marco Perez
Blocked

243 As a mobile banking customer, I must
Marco Perez
Blocked

fixeddate
Closed Items: 1 | Open Items: 6

New 8/6 In Progress 11/8 Done 4/4

In Development In Test

250 As a banking customer, I would
Marco Perez
Blocked

255 As a banking customer, I want my
Deb
Blocked

183 As a mobile customer, I must
Unassigned
Blocked

251 Simulate extra load to verify performance
Curtis
Blocked

252 Prototype caching for special loan
Marco Perez
Blocked

253 Estimate number of net new loan traffic
Curtis
Blocked

254 Estimate number of net new loan traffic
Marco Perez
Blocked

standard
Closed Items: 1 | Open Items: 5

New 8/6 In Progress 11/8 Done 4/4

In Development In Test

185 As a smart phone user I need a
Unassigned
Blocked

226 As a user, I must be able to chat with an
Unassigned
Blocked

227 As a mobile customer, I must
Marco Perez
Blocked

228 As a mobile customer, I must
Deb
Blocked

229 As a mobile customer, I must
Deb
Blocked

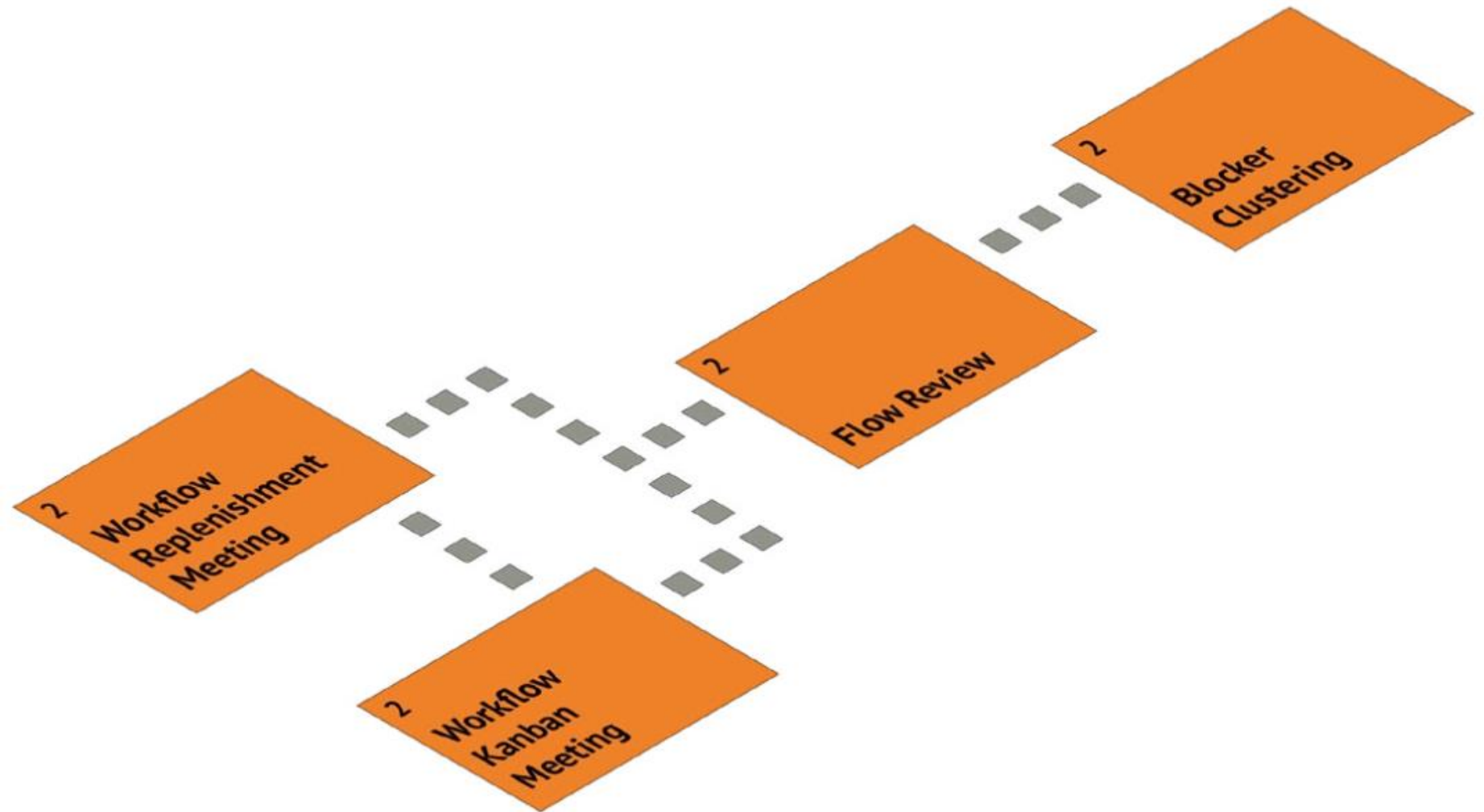
182 As a user, I must be able to chat with an
Unassigned
Blocked

Implement Feedback Loops



- Feedback loops are an essential part of any controlled process
 - Are especially important for evolutionary change.
 - Kanban defines specific feedback opportunities called “cadences”.
- Do not Distribute
- Cadence: the beat, time, or measure of rhythmical motion or activity

Service-Oriented Cadence



Workflow Kanban Meeting



Involve the service teams in a collaborative conversation about the status of the work, queues between stages of the workflow and problems that affect the flow of work.



- Commitments
- Decisions about new initiatives and capacity allocation
- Policy changes

- Shared understanding of how to best to flow current work
- Improvement suggestions



Immediate service group or team doing the work (4-50 people)



Scope: End-to-end workflow of a service
The identified issues are resolved after the meeting by team members with relevant knowledge and skills



Weekly, depending on coordination cost



10-20 mins

How to Run a Workflow Kanban Meeting

- One person facilitates
- **“Walk the board”** from right to left:
 - Go through the tickets from right (most close to completion) to left (most recently started)
- Remember to **manage work, not workers!**
- The facilitator may solicit a status update on a ticket or simply ask if there is additional information that is not on the board yet.

How to Run a Workflow Kanban Meeting – cont.

- More mature organizations may only discuss blocked items, and not discuss every ticket on the board
- There will be discussion of pullable tickets in “Done” queues. Which ticket should be pulled next and by whom?

Do not Distribute

Flow Review



Develop an initial quantitative understanding of the delivered service, facilitating work planning and improving predictability.



- Progress and system capability data from Kanban Meeting and tracking system
- Actions from Risk Review

- Defined actions to improve capability
- Communication of decisions



(parts of the) service delivery team, optionally team leads



Compare current capability with defined target conditions
Discuss possible improvement actions



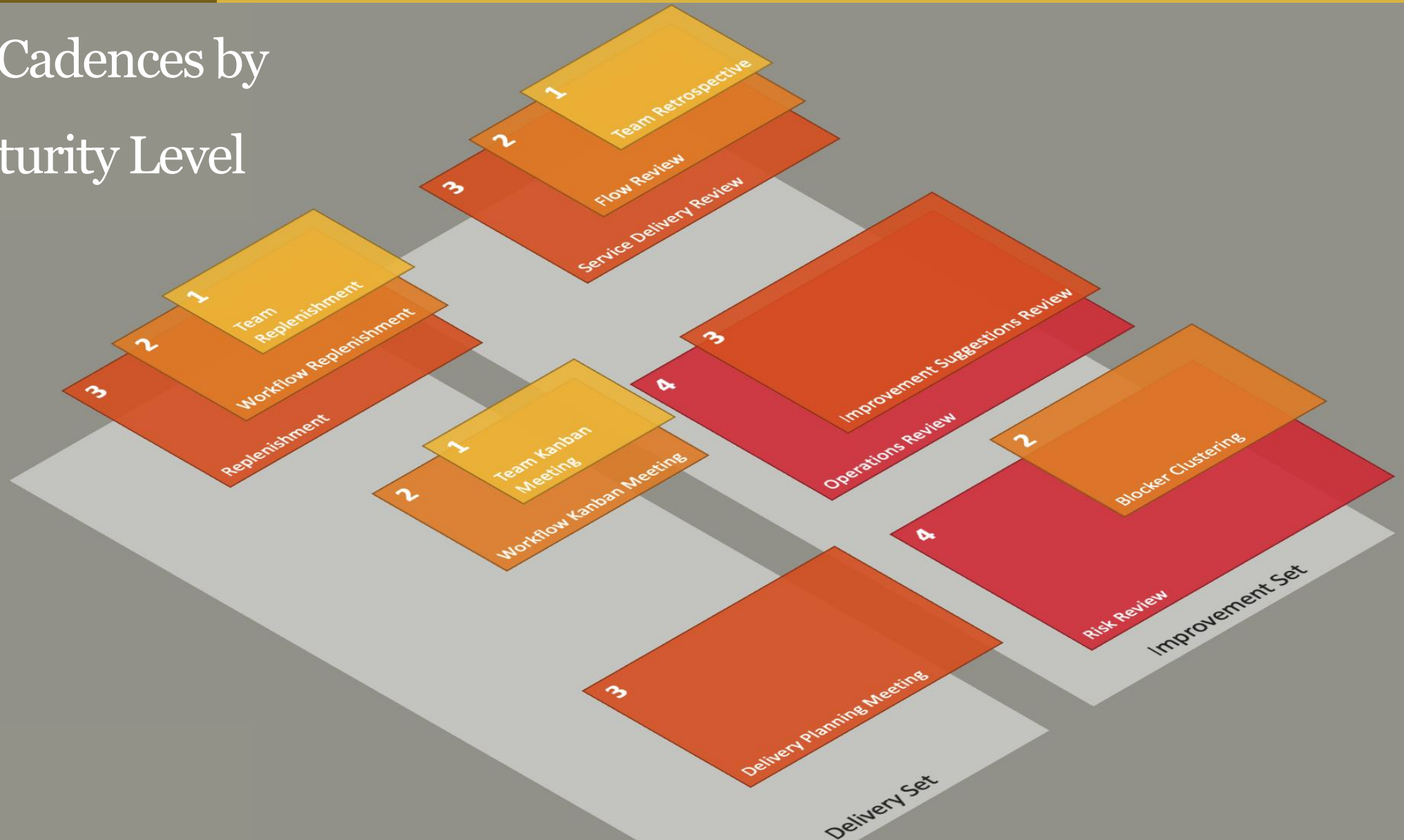
Twice a month 30 min



Typical Data Prepared for a Flow Review

- **Distribution of the demand** per work type
- **Range of Blocker time** per blocker cause
- **Range of defect and rework time** per cause
- Work items with **high age**
- Average and distribution of:
 - **Customer Lead Time** per type of work
 - **Arrival rate** of the demand
 - **Throughput** per type of work for a period
 - **System lead time** per type of work

All Cadences by Maturity Level



Official Licensed Material, Copyright © 2021 Kanban University

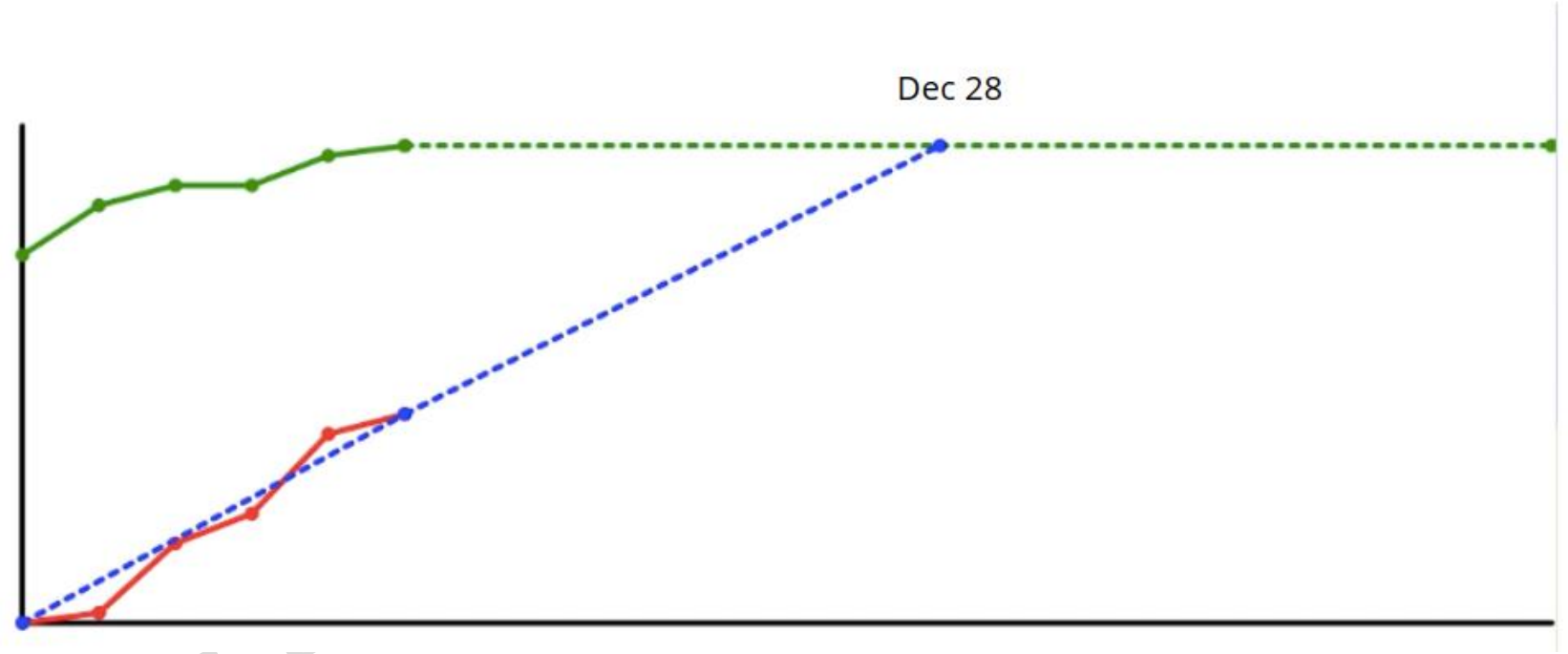
88

Using Models to Forecast

an short introduction of Monte Carlo Simulation

Do not Distribute

How to forecast?



- A simple regression to forecast the finish date
- But is it really this deterministic?

Vacanti's Verity

When making a forecast (predicting the future), you have to accept that there is more than one possible outcome.

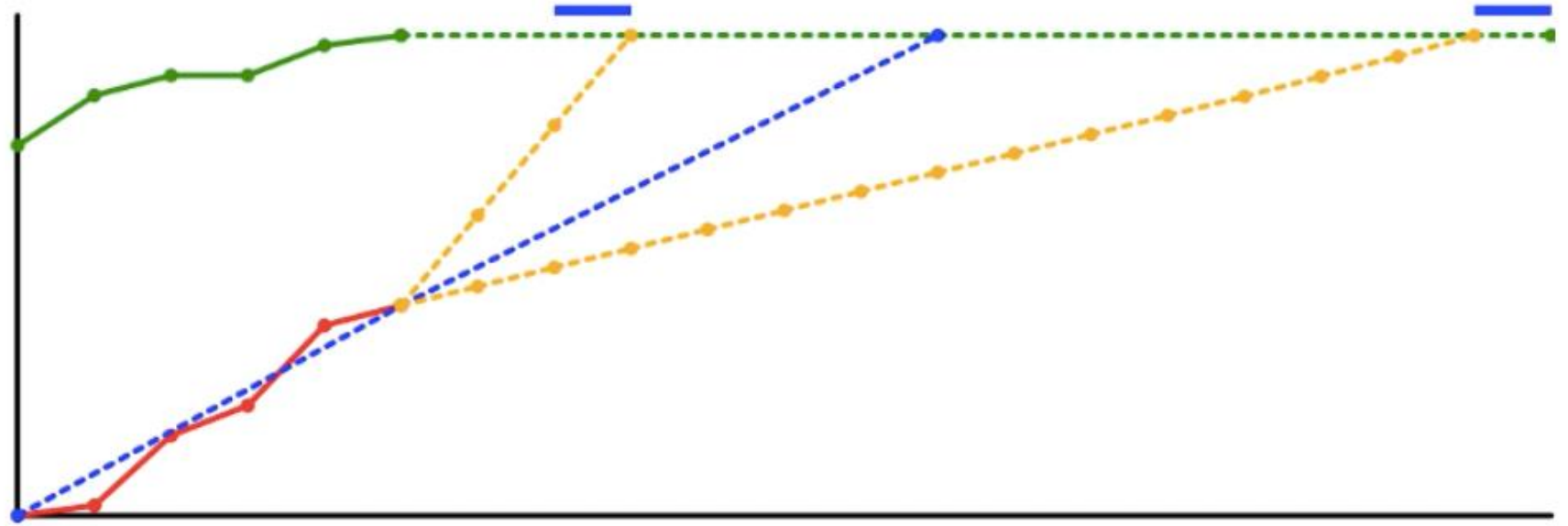
Therefore ... a forecast is a calculation about the future that includes both a range and a probability of that range occurring.

- ~ Danicel Vacanti

- Don't just use Average!!

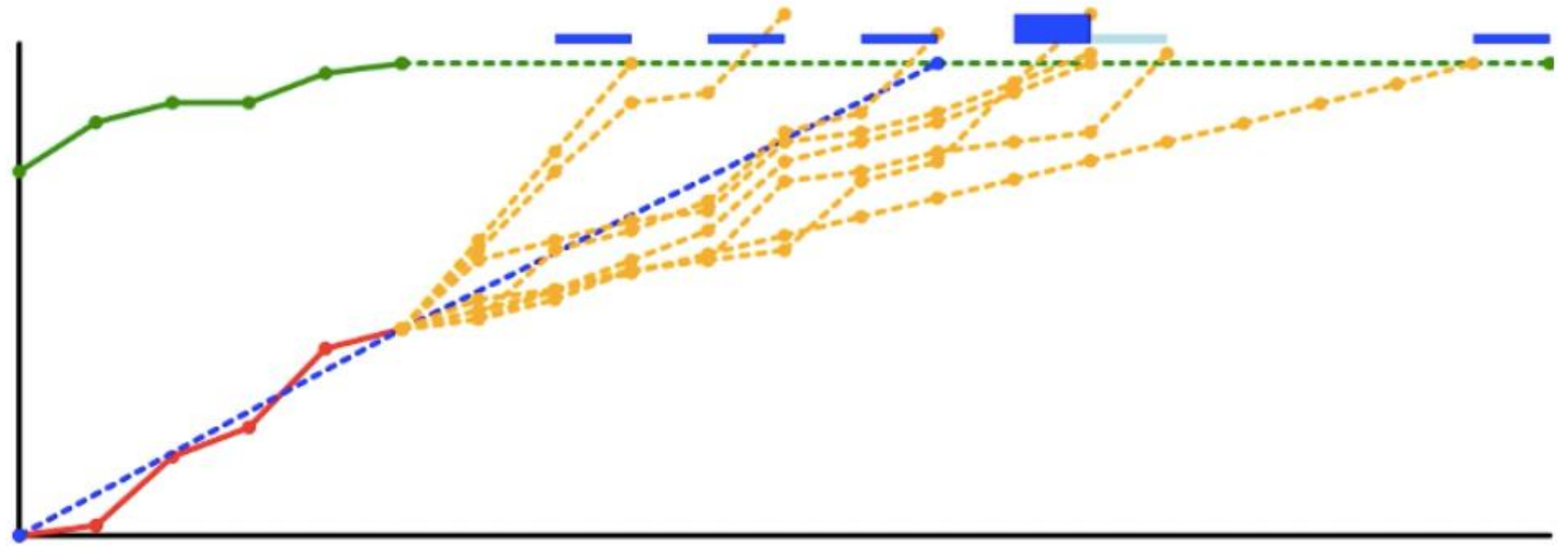
Do not Distribute

How to forecast?



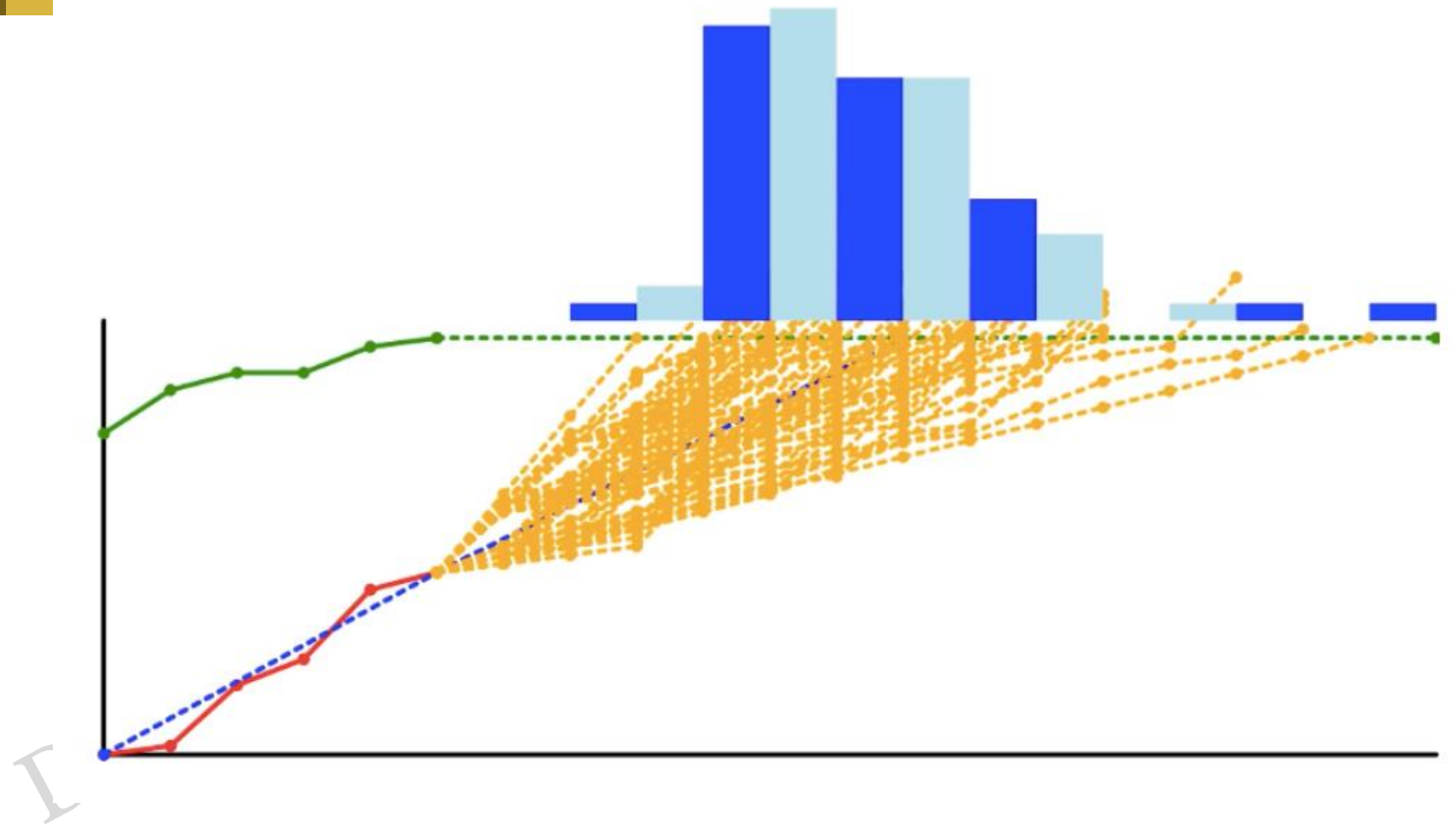
- Is it just enough to find best and worst scenarios?

How to forecast?



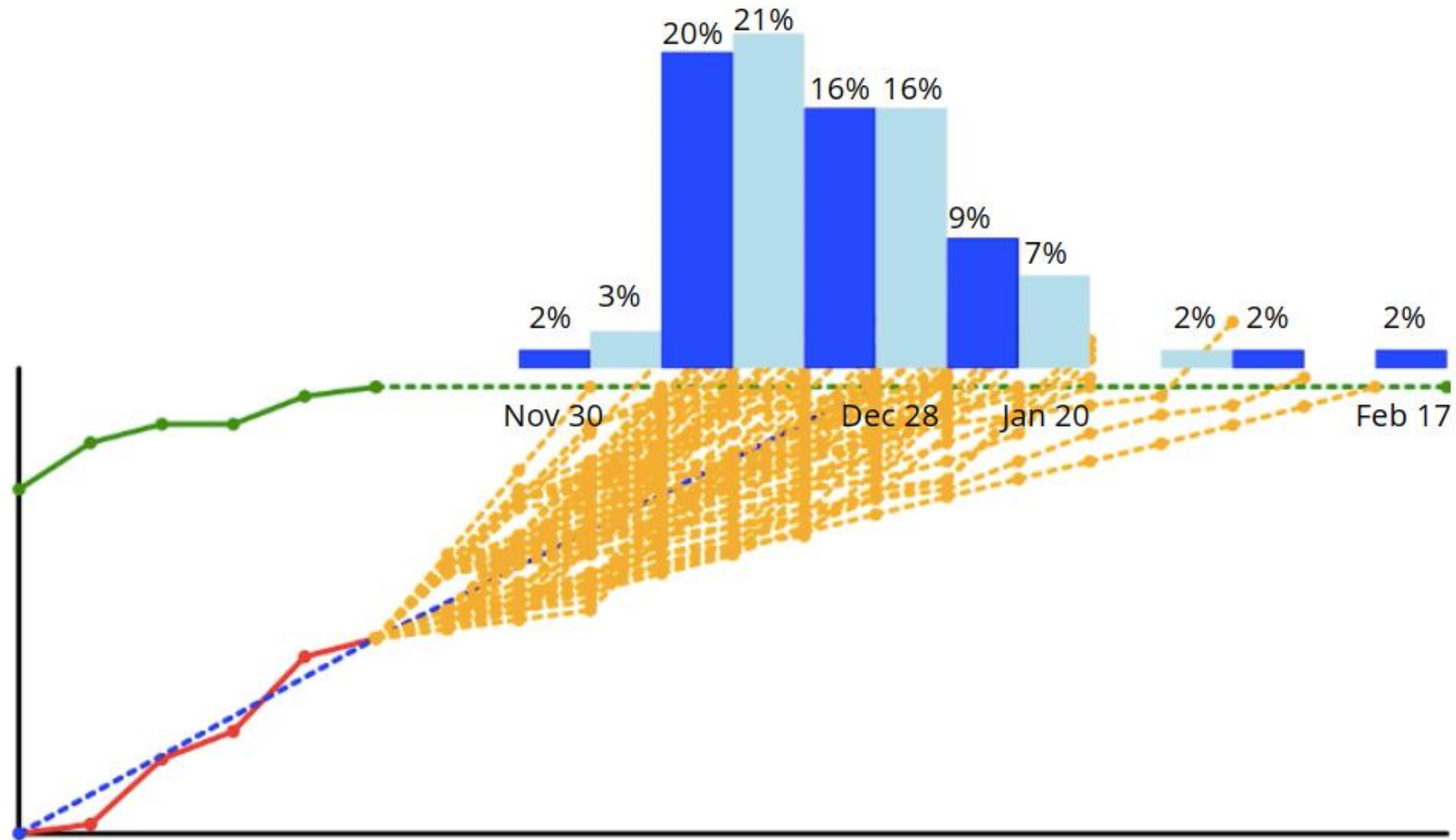
- A range of possibilities

How to forecast?

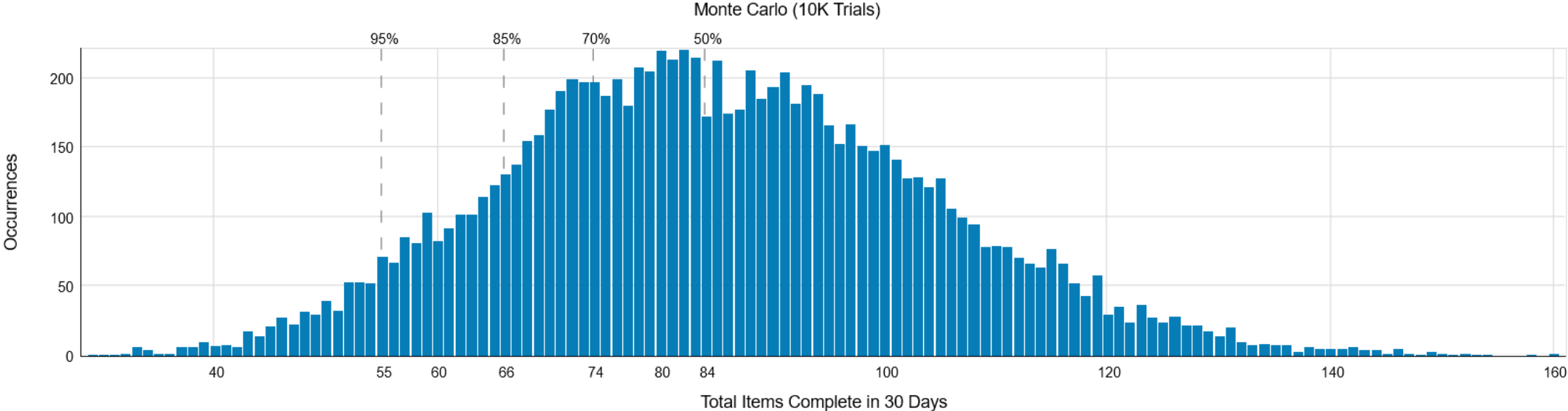


- A range of possibilities, generated by thousands of simulations
- To find the whole range of possibilities and they probability

A Monte Carlo simulation forecast

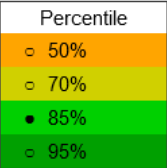


Monte Carlo Simulation – How Many

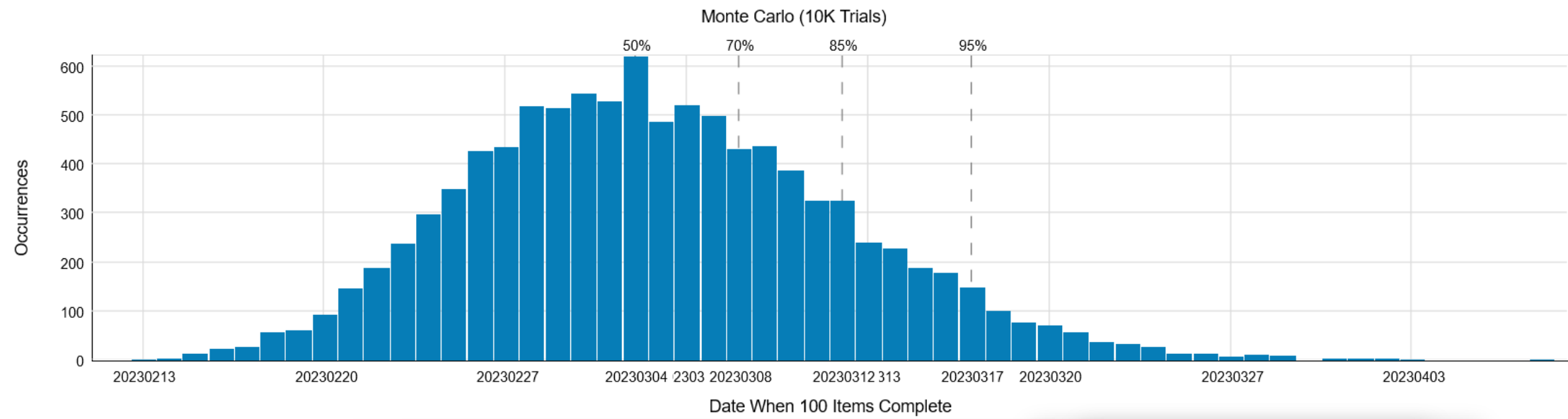


Jan 2023						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
2	4	6				

Feb 2023						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
			8	11	13	15
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				
29	30	31				
1	2	3				
4	5	6				
7	8	9				
10	11	12				
13	14	15				
16	17	18				
19	20	21				
22	23	24				
25	26	27				
28	29	30				
31						



Monte Carlo Simulation – When



Jan 2023						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Feb 2023						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

Mar 2023						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Apr 2023						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Legend	
0-50%	
50-70%	
70-85%	
85-95%	
95+%	

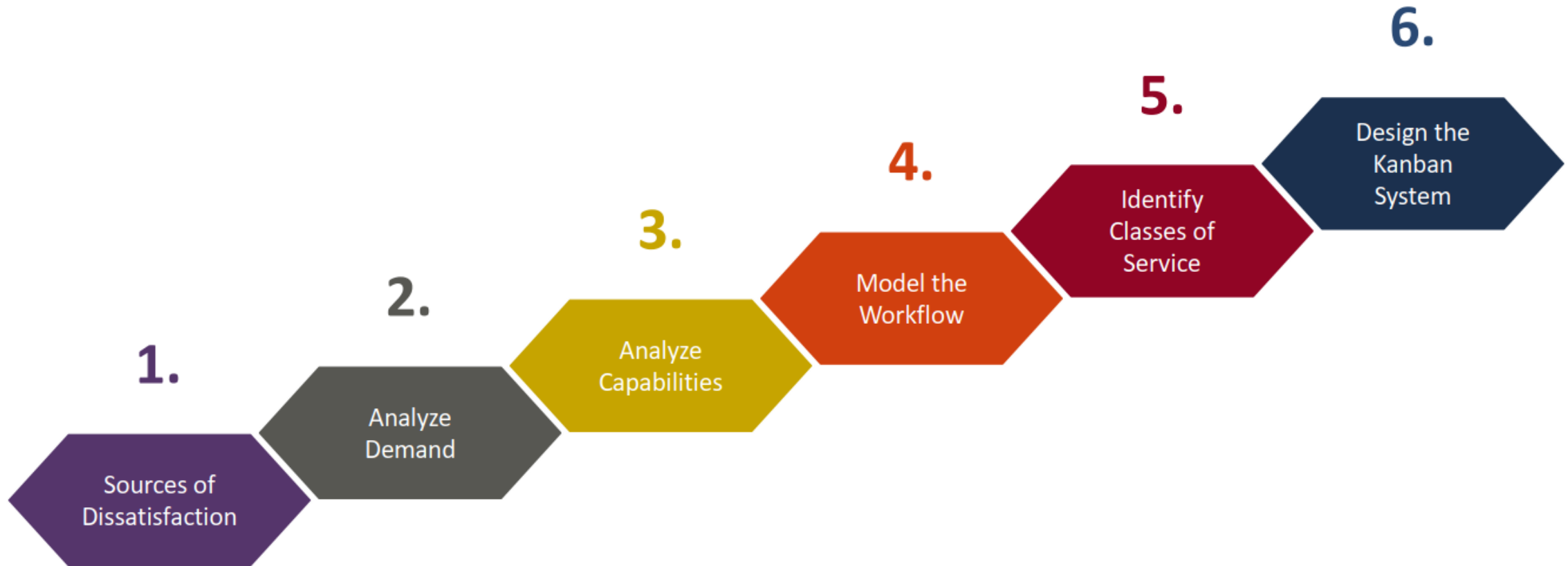
To read more on Monte Carlo Simulation

- If you want to read more on Monte Carlo Simulation for Knowledge work
 - Vacanti, “[Actionable Agile Metrics for Predictability](#)”, ActionableAgile Press, 2015
 - Vacanti, “[When Will It Be Done](#)”, ActionableAgile Press, 2018
 - Magennis, “Lean Forecasting”, 2012
 - Magennis, [Focused Objective website](#) Workshops and Video Tutorials

STATIK method

Do not Distribute

Overview of STATIK Steps



First, let's read the Case Study

KANBAN "BLUE BOOK" EXTRACT

**The Microsoft
XIT Sustaining Engineering Story**

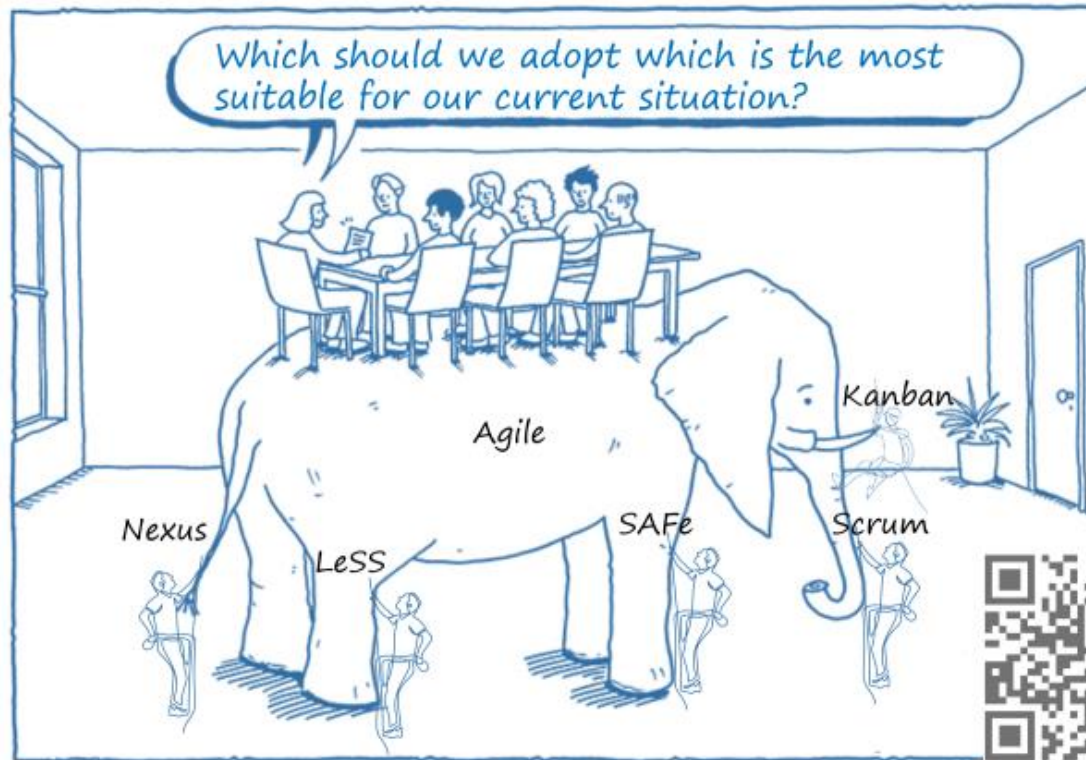
Do not Distribute

Part 2.4

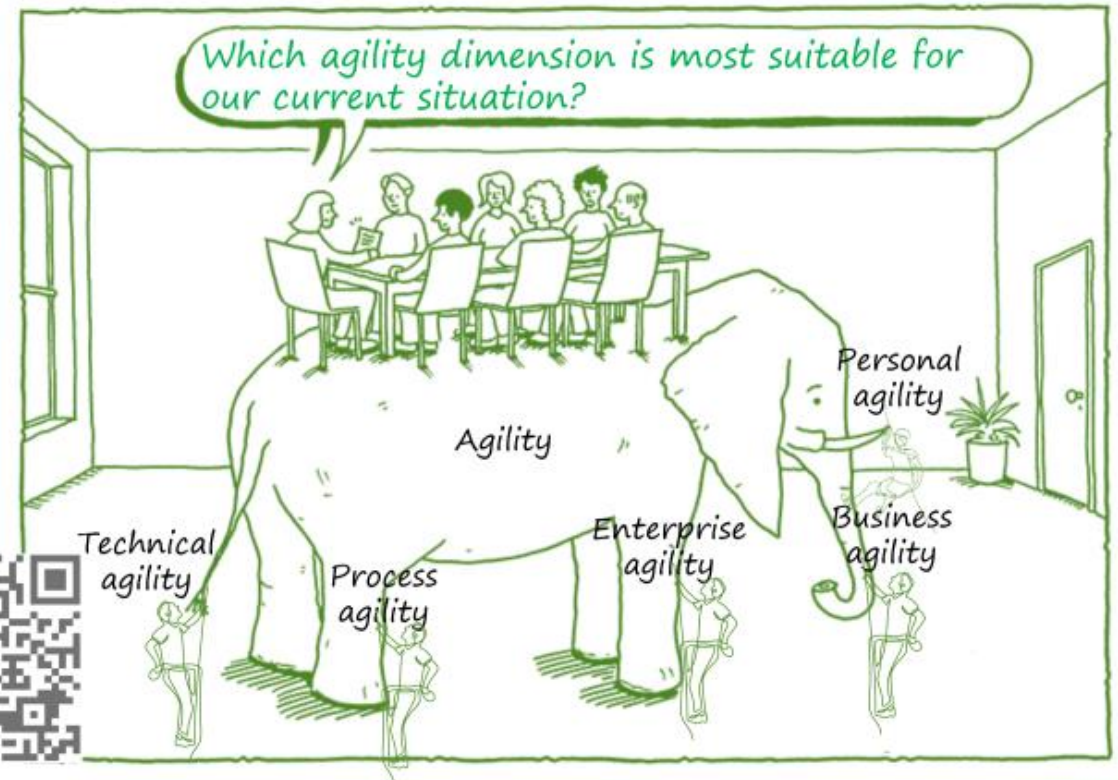
Agile

What is Agile?

Past or Present

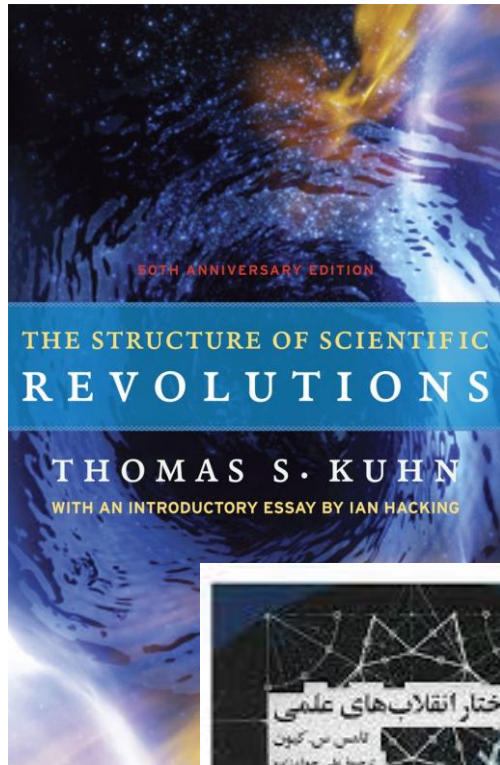


Current or Future



Not Just Removing The Elephant From The Room, We're Removing Our Inner Elephant

A Constructivist View | بر ساخت گرایانه



- Paradigms are Constructed by people
- Paradigm Shift
- But what is Paradigm?
- Motamem's Article ([link](#))



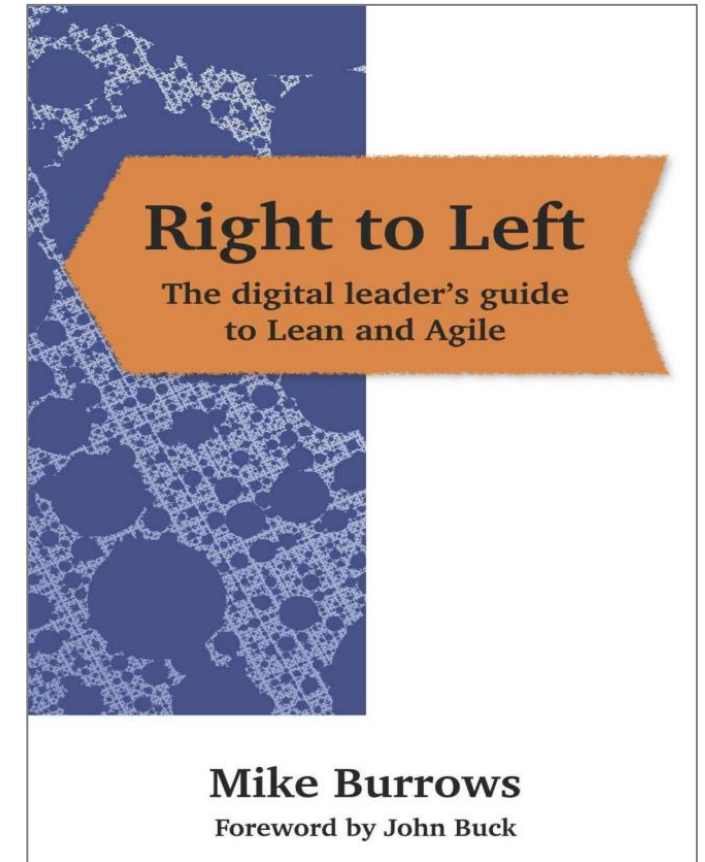
Thomas S. Kuhn
(1922-1996)

پارادایم چیست ؟ پارادایم شیف به چه معناست؟

Agile:

People collaborating over
the rapid evolution of working solution
that is already beginning to meet needs

Working Solution, instead of Working Software



Mike Burrows, *Right to Left*,
New Generation Publishing, 2019

Agile:

People bringing their various skills to bear on the rapid evolution of working solution that is already beginning to meet needs, working in teams that place high value on collaboration and adaptation.

Working Solution, instead of Working Software

Mike Burrows, *Right to Left*,
New Generation Publishing, 2019

Manifesto:

Ideological statement of how things should be, or more frequently how they should not be

Do not Distribute

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.

12 Principles of Agile Software Development (1/4)

- *Principles behind the Agile Manifesto*
- We follow these principles:
 - 1)** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
 - 2)** Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

12 Principles of Agile Software Development (2/4)

- 3) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4) Business people and developers must work together daily throughout the project.
- 5) Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

12 Principles of Agile Software Development (3/4)

- 6)** The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 7)** Working software is the primary measure of progress.
- 8)** Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

12 Principles of Agile Software Development (4/4)

- 9) Continuous attention to technical excellence and good design enhances agility.
- 10) Simplicity--the art of maximizing the amount of work not done--is essential.
- 11) The best architectures, requirements, and designs emerge from self-organizing teams.
- 12) At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The Declaration of Interdependence (2005)

1. We increase return on investment by making **continuous flow of value** **our focus**.
2. We deliver **reliable results** by engaging customers in frequent interactions and shared ownership.
3. We expect uncertainty and manage for it through iterations, anticipation, and adaptation.
4. We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
5. We boost **performance** through group accountability for results and shared responsibility for team effectiveness.
6. We improve **effectiveness and reliability** through situationally specific strategies, processes and practices.

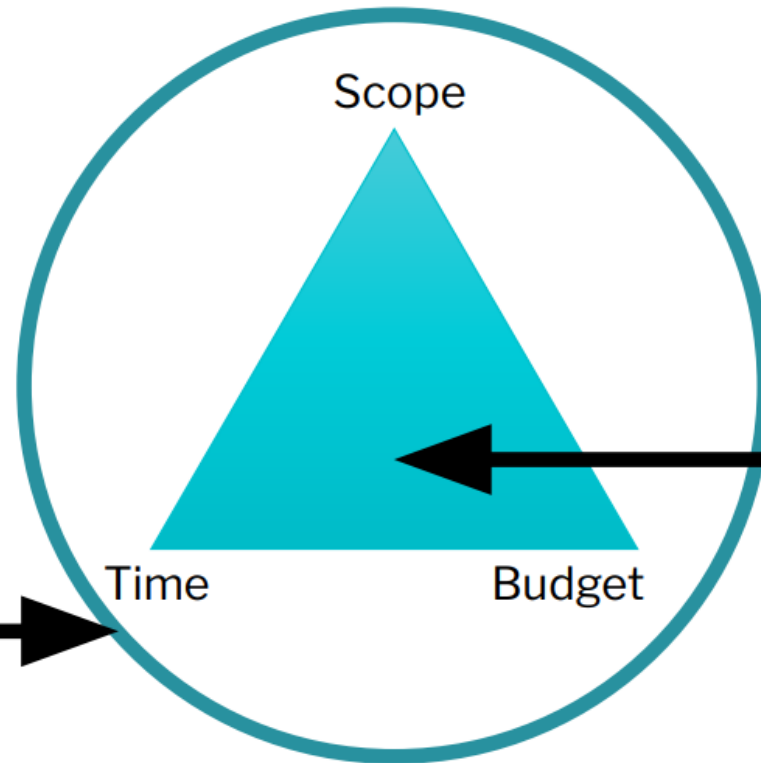
Project Mindset vs. Product Mindset

Product Mindset

Success *continuously* driven by *business metrics* outside in:

- User adoption/retention
- Revenue
- Cost savings per feature

Leads to less waste, more creativity, and more releases.



Project Mindset

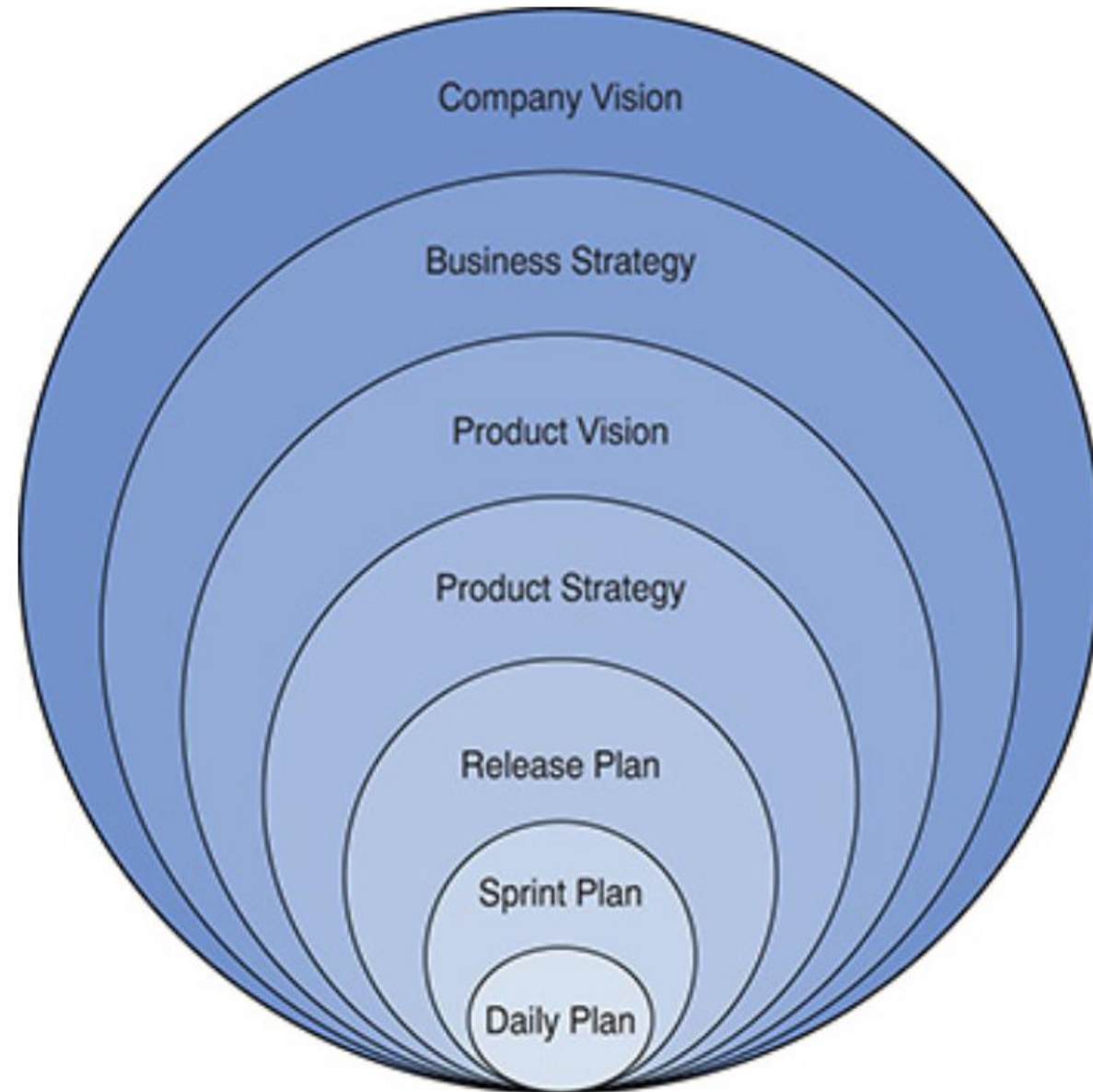
Success *upfront* defined inside out:

- Scope
- Time
- Budget

Leads to less business involvement, more task management.

Different Layers of Planning

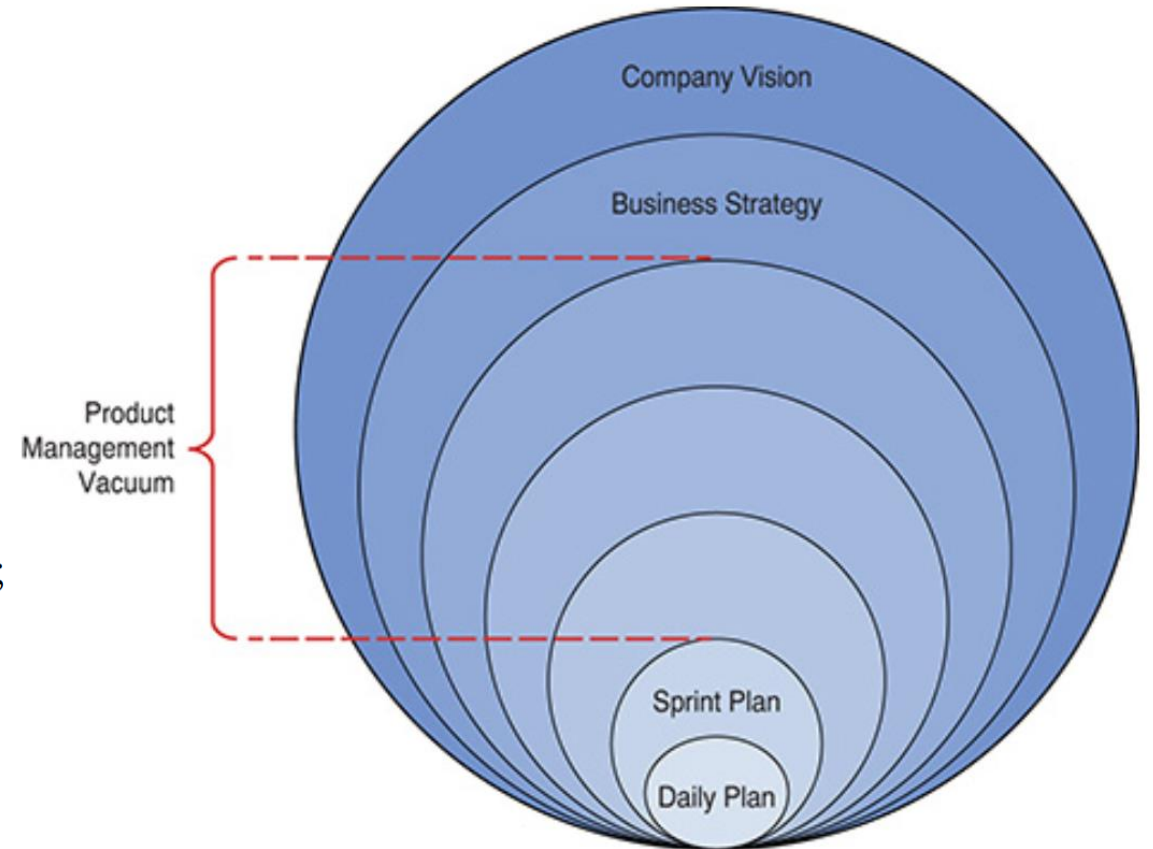
Do r



Product Management Vacuum

The bigger the vacuum:

- the more disconnected the technology groups are from the business;
- the less engaged the people on the ground become;
- the more reliance there is on project and task management;
- the more hierarchies and handoffs emerge;
- the more complexity is introduced;
- the harder it is to shift directions when the business climate changes;
- the more “busy work” is created;
- the more waste and rework occurs; and
- the less value is delivered to customers.



Liedtka's Strategic Thinking Model

معرفی یک مدل تفکر استراتژیک | الگوی پنج بخشی لیدکا

Intelligent
Opportunism

Knowing Trends

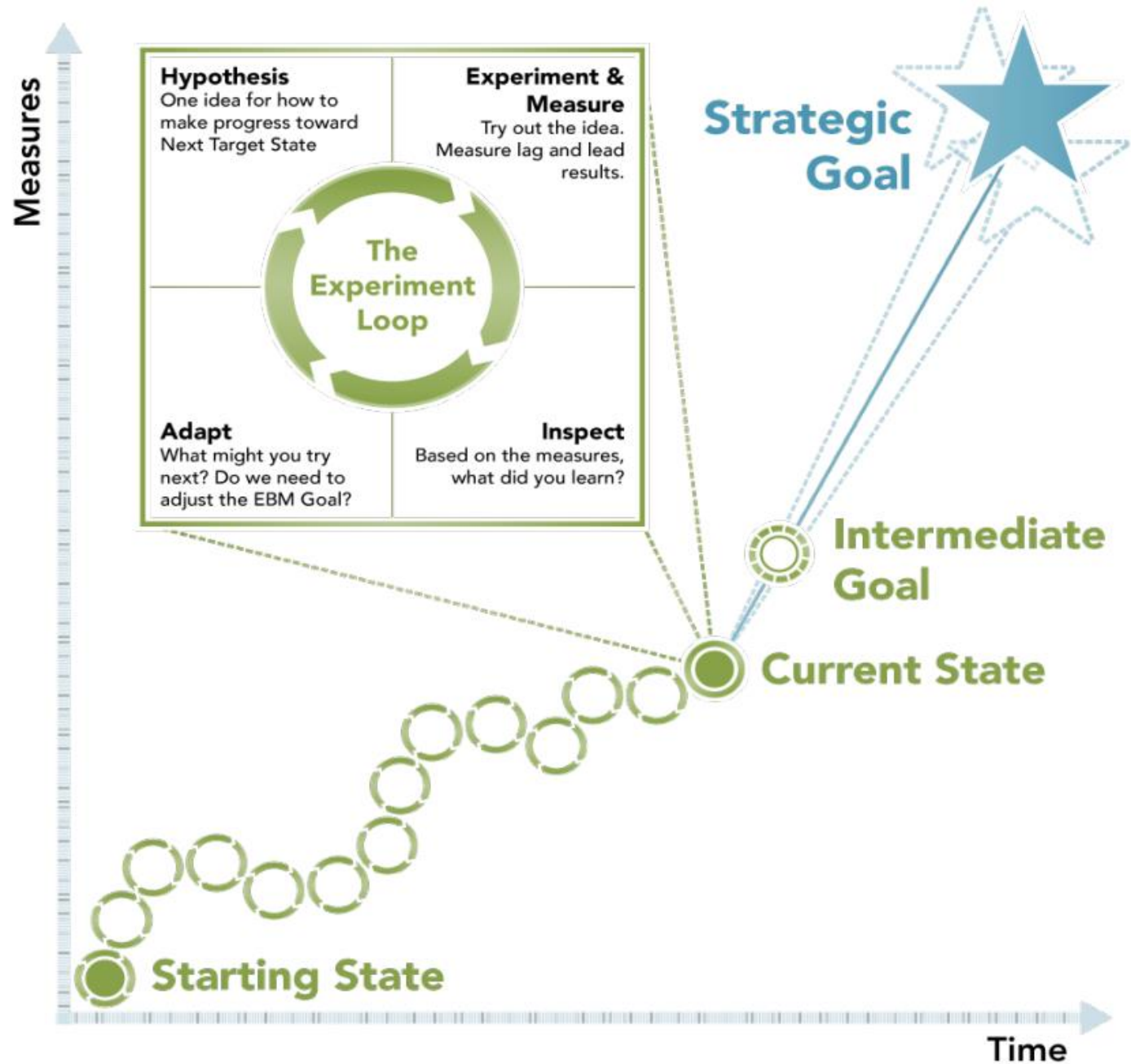
Hypothesis-driven
Thinking and
Experimentation

System Thinking

Strategic Intent

How to measure
where we are?

What to measure?



A Mindset and Behavioral Shift for Management

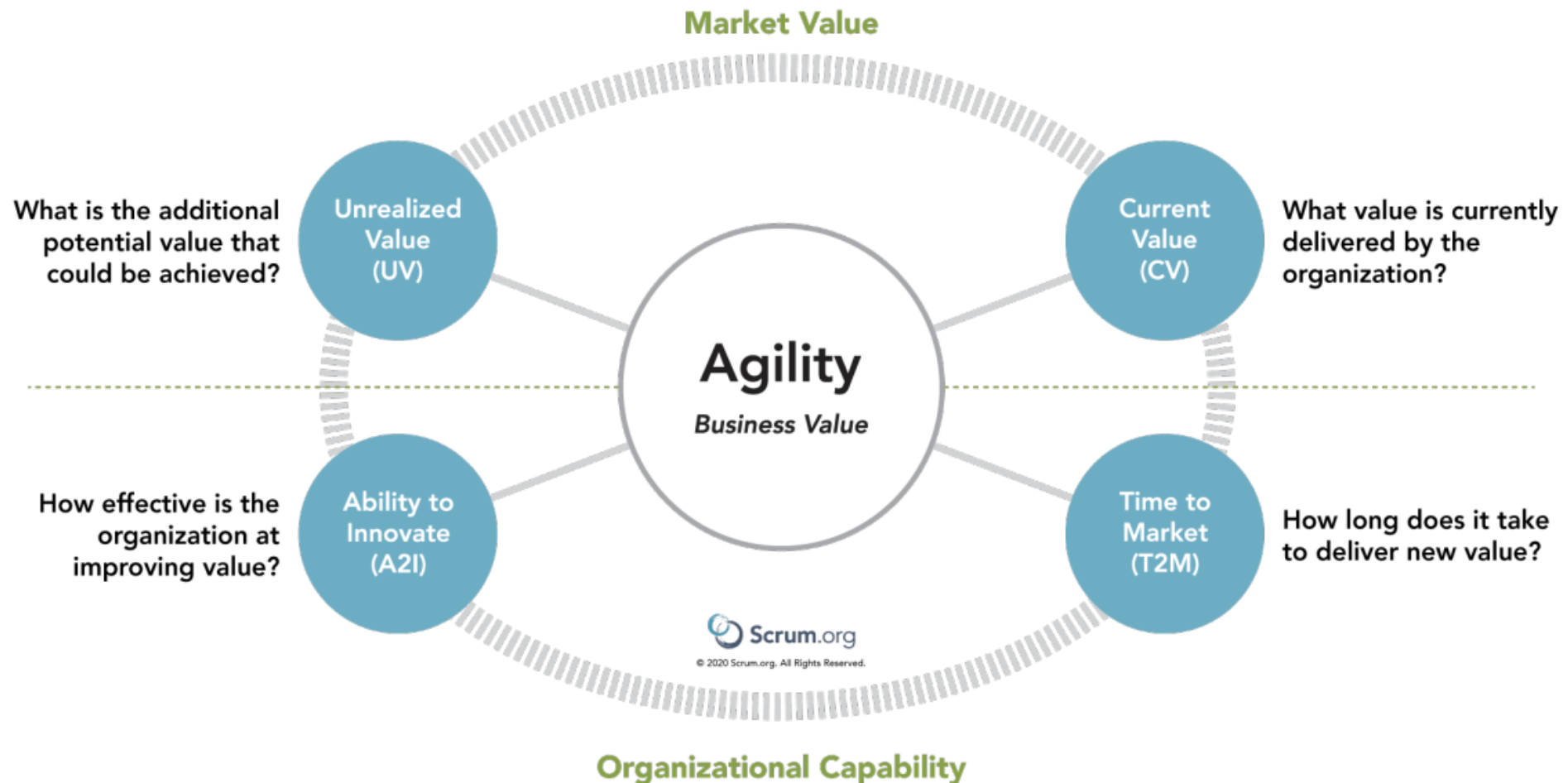
Predictive Management

- Long-term detailed plans
- Assign and control the work
- Maximize capacity and effort
- Keep all on schedule
- Driven by meetings and reports
- Intervene to fix all problems
- Provide external motivators

Empirical Management

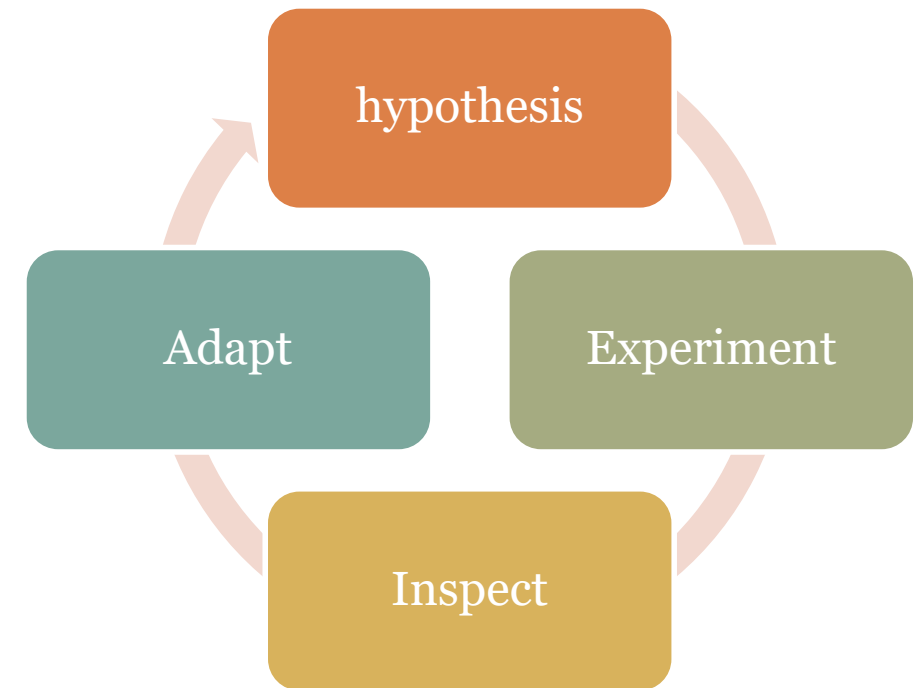
- Goals, Vision, Direction
- Foster the environment
- Help remove impediments
- Attend sprint review
- Share incremental feedback
- Manage for value
- Autonomy, mastery, purpose

Evidence-based Management Framework - KVAs



The Experiment Loop (1/2)

The Experiment Loop helps organizations move from their **Current State** toward their **Next Target Goal**, and ultimately their **Strategic Goal**, by taking **small, measured steps**, called **experiments**, using **explicit hypothesis**.



The Experiment Loop (2/2)

- Hypothesis

A proposed explanation for some observation that has not yet been proven (or disproven)

- Experiment

Make the change you think will help you to improve and gather data to support or refute your hypothesis

- Inspect

Did the change you made improve your results based on the measurements you have made?

- Adapt

Adapt your goals or your approach based on what you learned.

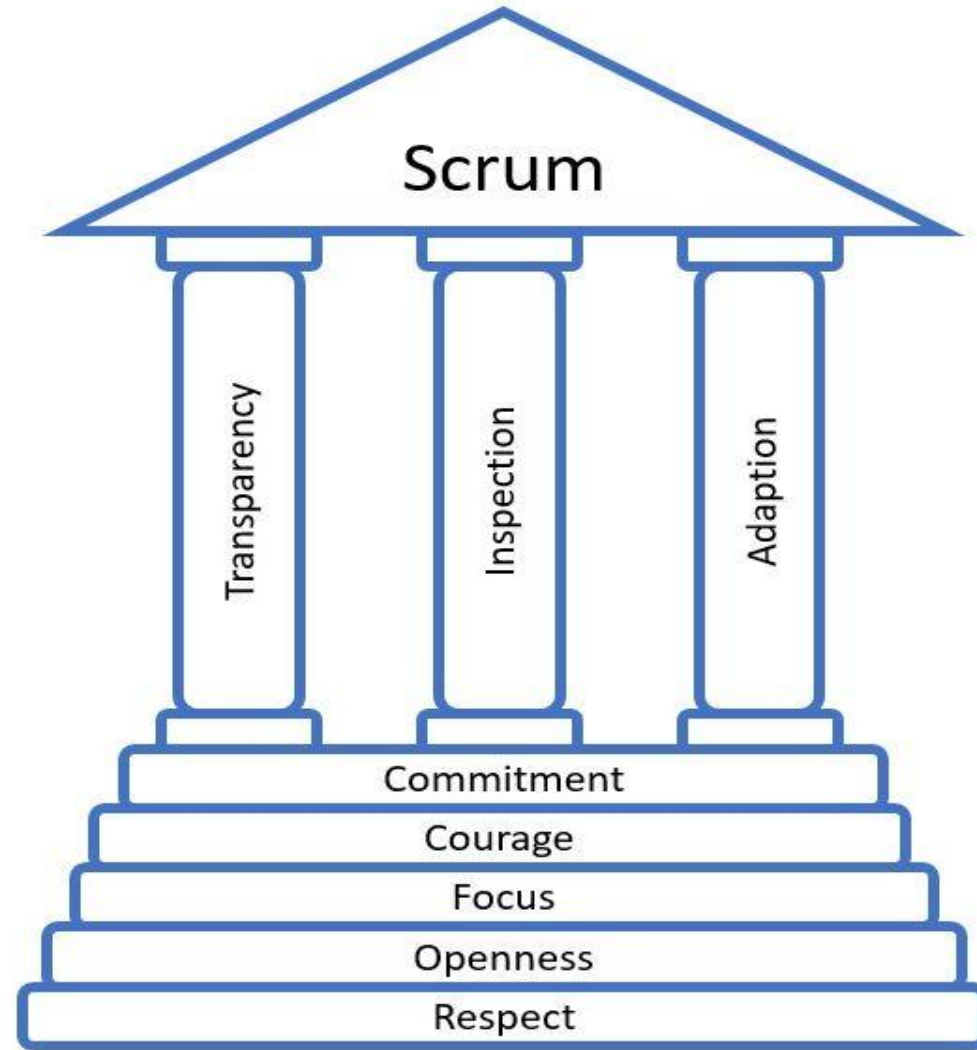
Change:	
Owner:	Mentor:
Context / scope:	Aligned to objective:

Hypothesis		Assumptions & Dependencies	
We believe that will result in If successful, we might expect to see:		Assumptions (to be validated)	Dependencies (to be resolved)
Risks		Pilot experiments (new A3s)	
Downside (to be invalidated/mitigated)	Upside (to be nurtured)		(owner)
People		Insights	
Directly impacted	Other stakeholders & influencers		



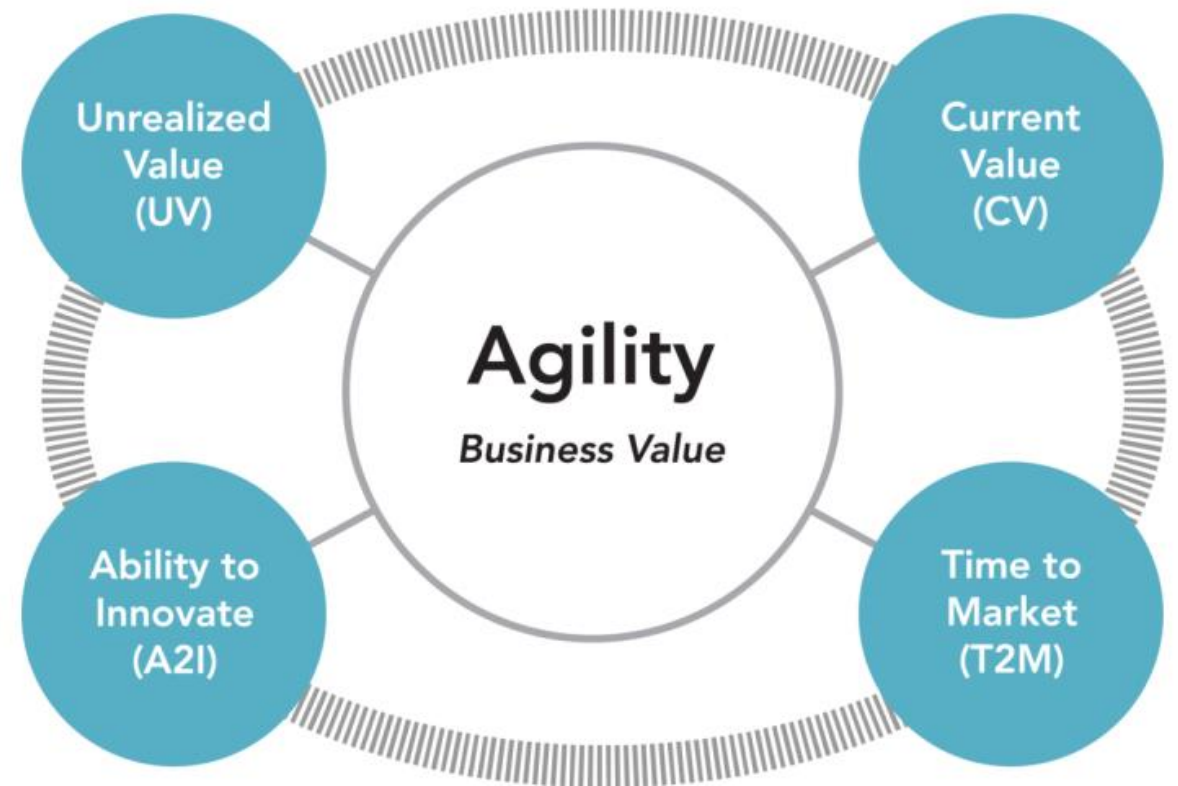
Scrum Temple

Three Pillars of Empiricism



Key Value Areas (KVAs)

- Current Value (CV)
- Time to Market (T2M)
- Ability to Innovate (A2I)
- Unrealized Value(UV)



- Which **Key Value Measures** are in KVAs?

Current Value (CV) – examples

KVM	Measuring
Revenue per Employee	The ratio (gross revenue / # of employees) is a key competitive indicator within an industry. This varies significantly by industry.
Product Cost Ratio	Total expenses and costs for the product(s)/system(s) being measured, including operational costs compared to revenue .
Employee Satisfaction	Some form of sentiment analysis to help gauge employee engagement, energy, and enthusiasm .
Customer Satisfaction	Some form of sentiment analysis to help gauge customer engagement and happiness with the product .
Customer Usage Index	Measurement of usage , by feature, to help infer the degree to which customers find the product useful and whether actual usage meets expectations on how long users should be taking with a feature.

Unrealized Value (UV) – examples

KVM	Measuring
Market Share	The relative percentage of the market not controlled by the product ; the potential market share that the product might achieve if it better met customer needs .
Customer or User Satisfaction Gap	The difference between a customer or user's desired experience and their current experience .
Desired Customer Experience or satisfaction	A measure that indicates the experience that the customer would like to have .

Time to Market (T2M) – examples (1/4)

KVM	Measuring
Build and Integration Frequency	The number of integrated and tested builds per time period. For a team that is releasing frequently or continuously, this measure is superseded by actual release measures.
Release Frequency	The number of releases per time period, e.g. continuously, daily, weekly, monthly, quarterly, etc. This helps reflect the time needed to satisfy the customer with new and competitive products
Release Stabilization Period	The time spent correcting product problems between the point the developers say it is ready to release and the point where it is actually released to customers. This helps represent the impact of poor development practices and underlying design and code base.
Mean Time to Repair	The average amount of time it takes from when an error is detected and when it is fixed. This helps reveal the efficiency of an organization to fix an error.

Time to Market (T2M) – examples (2/4)

KVM	Measuring
Customer Cycle Time	The amount of time from when work starts on a release until the point where it is actually released. This measure helps reflect an organization's ability to reach its customer.
Lead Time	The amount of time from when an idea is proposed, or a hypothesis is formed until a customer can benefit from that idea. This measure may vary based on customer and product. It is a contributing factor for customer satisfaction.
Lead Time for Changes	The amount of time to go from code-committed to code successfully running in production. <i>For more information, see the DORA 2019 report.</i>

Time to Market (T2M) – examples (3/4)

KVM	Measuring
Deployment Frequency	The number of times that the organization deployed (released) a new version of the product to customers/users. <i>For more information, see the DORA 2019 report.</i>
Time to Restore Service	The amount of time between the start of a service outage and the restoration of full availability of the service. <i>For more information, see the DORA 2019 report.</i>
Time-to-Learn	The total time needed to sketch an idea or improvement, build it, deliver it to users, and learn from their usage.

Time to Market (T2M) – examples (4/4)

KVM	Measuring
Time to remove Impediment	The average amount of time from when an impediment is raised until when it is resolved. It is a contributing factor to lead time and employee satisfaction.
Time to Pivot	A measure of true business agility that presents the elapsed time between when an organization receives feedback or new information and when it responds to that feedback; for example, the time between when it finds out that a competitor has delivered a new market-winning feature to when the organization responds with matching or exceeding new capabilities that measurably improve customer experience.

Ability to Innovate (A2I) – examples (1/3)

KVM	Measuring
Innovation Rate	The percentage of effort or cost spent on new product capabilities, divided by total product effort or cost. This provides insight into the capacity of the organization to deliver new product capabilities.
Defect Trends	Measurement of change in defects since last measurement. A defect is anything that reduces the value of the product to a customer, user, or to the organization itself. Defects are generally things that don't work as intended
On-Product Index	The percentage of time teams spend working on product and value.
Installed Version Index	The number of versions of a product that are currently being supported. This reflects the effort the organization spends supporting and maintaining older versions of software.

Ability to Innovate (A2I) – examples (2/3)

KVM	Measuring
Technical Debt	A concept in programming that reflects the extra development and testing work that arises when “quick and dirty” solutions result in later remediation. It creates an undesirable impact on the delivery of value and an avoidable increase in waste and risk.
Production Incident Count	The number of times in a given period that the Development Team was interrupted to fix a problem in an installed product. The number and frequency of Production Incidents can help indicate the stability of the product.
Active Product (Code) Branches	The number of different versions (or variants) of a product or service. Provides insight into the potential impact of change and the resulting complexity of work..

Ability to Innovate (A2I) – examples (3/3)

KVM	Measuring
Time Spent Merging Code Between Branches	The amount of time spent applying changes across different versions of a product or service. Provides insight into the potential impact of change and the resulting complexity of work.
Time Spent Context-Switching	Examples include time lost to interruptions caused by meetings or calls, time spent switching between tasks, and time lost when team members are interrupted to help people outside the team can give simple insight into the magnitude of the problem
Change Failure Rate	The percentage of released product changes that result in degraded service and require remediation (e.g. hotfix, rollback, patch). For more information, see the DORA 2019 report.

17 Metrics to Help Build Better Software

“CRAWL” METRICS

Deployment Frequency (DORA) P E D
What How often an organization successfully releases to production Measurement Deployments per Day Source Value Stream Delivery Platform Impact potential Increase customer value by reduced Time-to-Market
Lead Time for Changes (DORA) P E D
What The amount of time it takes a commit to get into production Measurement Lead Time in Days Source Value Stream Delivery Platform + Agile Planning Tool Impact potential Increase developer efficiency
Change Failure Rate (DORA) P E D
What The percentage of deployments leading to a degradation in service that must be addressed Measurement Number of incidents / number of deployments Source Value Stream Delivery Platform + Incident Management System Impact potential Increase customer satisfaction by decreased number of outages
Time to Restore Service (DORA) P E D
What How long it takes an organization to recover from a failure in production Measurement Restore Time in hours Source Value Stream Delivery Platform + Incident Management System Impact potential Increase customer satisfaction by decreased duration of outages
Monthly Active Users (MAU) P E D
What Adoption rates for features, products or services Measurement Monthly active users per feature, product or service Source Application Monitoring Impact potential Alignment of features and user needs

“RUN” METRICS

Flow Time C P E
What Time it takes a work item from customer request to production Measurement Time in Days Source Value Stream Delivery Platform + Agile Planning Impact potential Uncover process bottlenecks
Flow Velocity C P E
What Completed work items over a given period of time Measurement Number of work items per time period Source Value Stream Delivery Platform + Agile Planning Tool Impact potential Accelerate value delivery
Flow Load C P E
What Number of work items in progress Measurement Number of active work items Source Value Stream Delivery Platform + Agile Planning Tool Impact potential Improve over and under-utilization
Flow Efficiency C P E
What The distribution of work items in work and waiting states Measurement Active work time / total time of flow Source Value Stream Delivery Platform + Agile Planning Tool Impact potential Identify waste
Flow Distribution C P E
What Categorizing the types of work being done at any given time Measurement Percentage distribution between Features, Defects, Risks and Debts work items Source Value Stream Delivery Platform + Agile Planning Tool Impact potential Improve the level of innovation and ensure you are covering security, tech debt and other issues
Team Health C P E D
What Distribution of work among team members Measurement Work items distribution in terms of type and quantity Source Agile Planning Tool Impact potential Create an even team workload distribution
Return on Investment (ROI) C P
What Quantifiable business value produced by development efforts Measurement Expenses vs Gains over time; Number of new customers Source Business Process Software + CRM Impact potential Improve planning; Increase overall business value of spend
Number of Vulnerabilities C P E D
What Software quality from the security standpoint Measurement Number of vulnerabilities per product or service Source Application Security Impact potential Increase product or service security and limit work on Defects, Risks and Debt
Team Satisfaction C E
What Employee satisfaction Measurement Employee Net Promoter Score (eNPS) Source Employee engagement tools Impact potential Increase Developer engagement

“WALK” METRICS

Customer NPS C P E D
What Customer satisfaction with new products and features Measurement Net Promoter Score Source Customer Success Management Platforms Impact potential Delivery of the right features at the right time
Service Level Indicator (SLI) C P E
What Performance against stated Service Level Objectives (SLO) and Agreements (SLA) Measurement Actual availability Source Application Monitoring Impact potential Increased customer satisfaction and retention
Cloud Costs C P E
What Cloud spend per product or service Measurement Costs in local currency Source Cloud Provider Cost Management Impact potential Reduce run costs by increased visibility and accountability (FinOps)

D DevOps Teams C CTO / Product Operations
P Product Manager E Engineering Leader

5 Tips for using metrics successfully

- Always define your own base and target levels for metrics on product, team or company level
- Do not track metrics on a developer level
- Do not compare apples and oranges. A mobile app is not the same as a backend service
- Metrics should be used to measure the improvement of DevOps teams, not to micromanage
- Learn from top-performing teams. You can even gamify the improvement process between teams

What Key Value Measures can you define
in your industry?

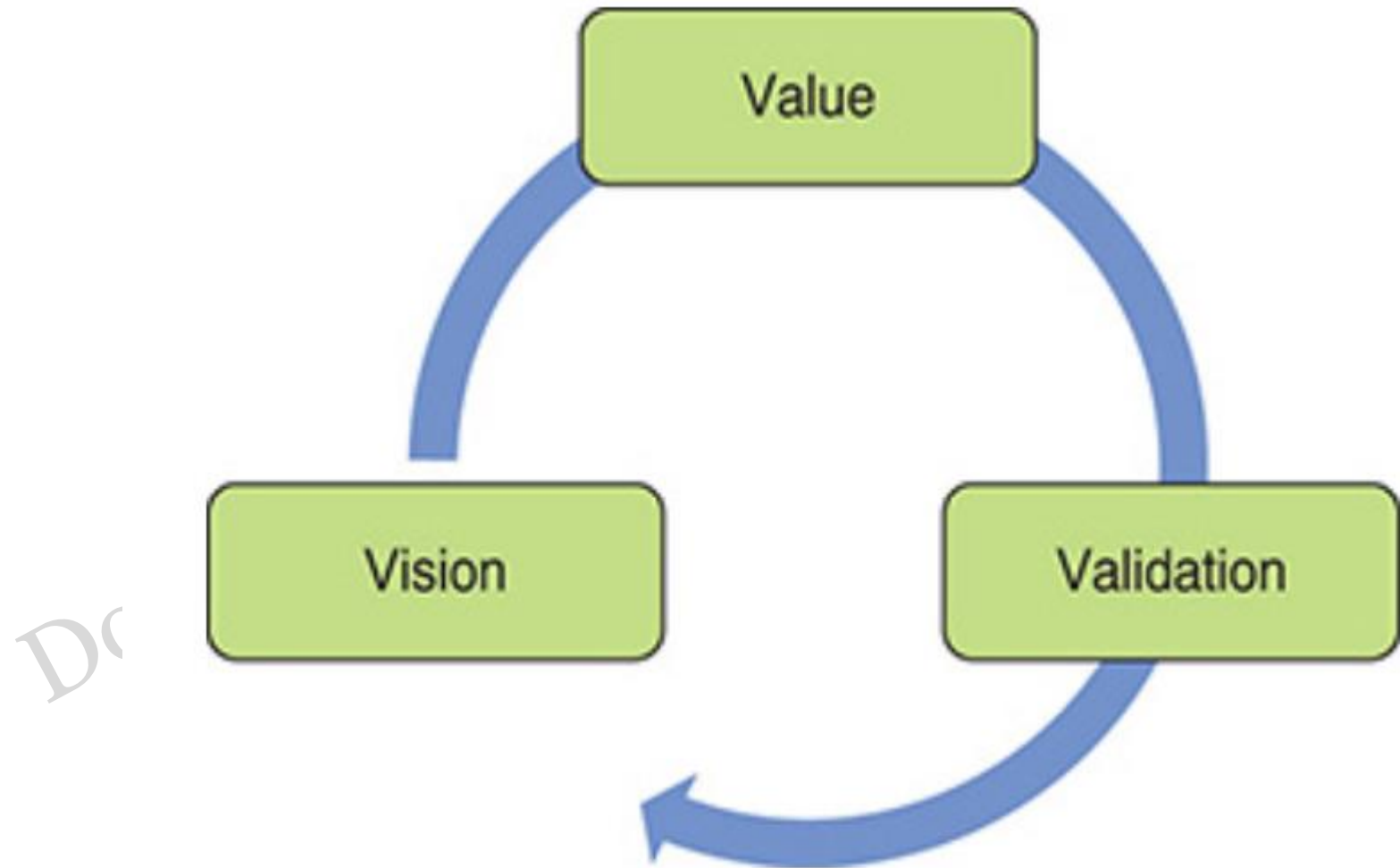
Do not Distribute

What can be found in
Evidence-Based **Portfolio** Management (EBM-PM)?

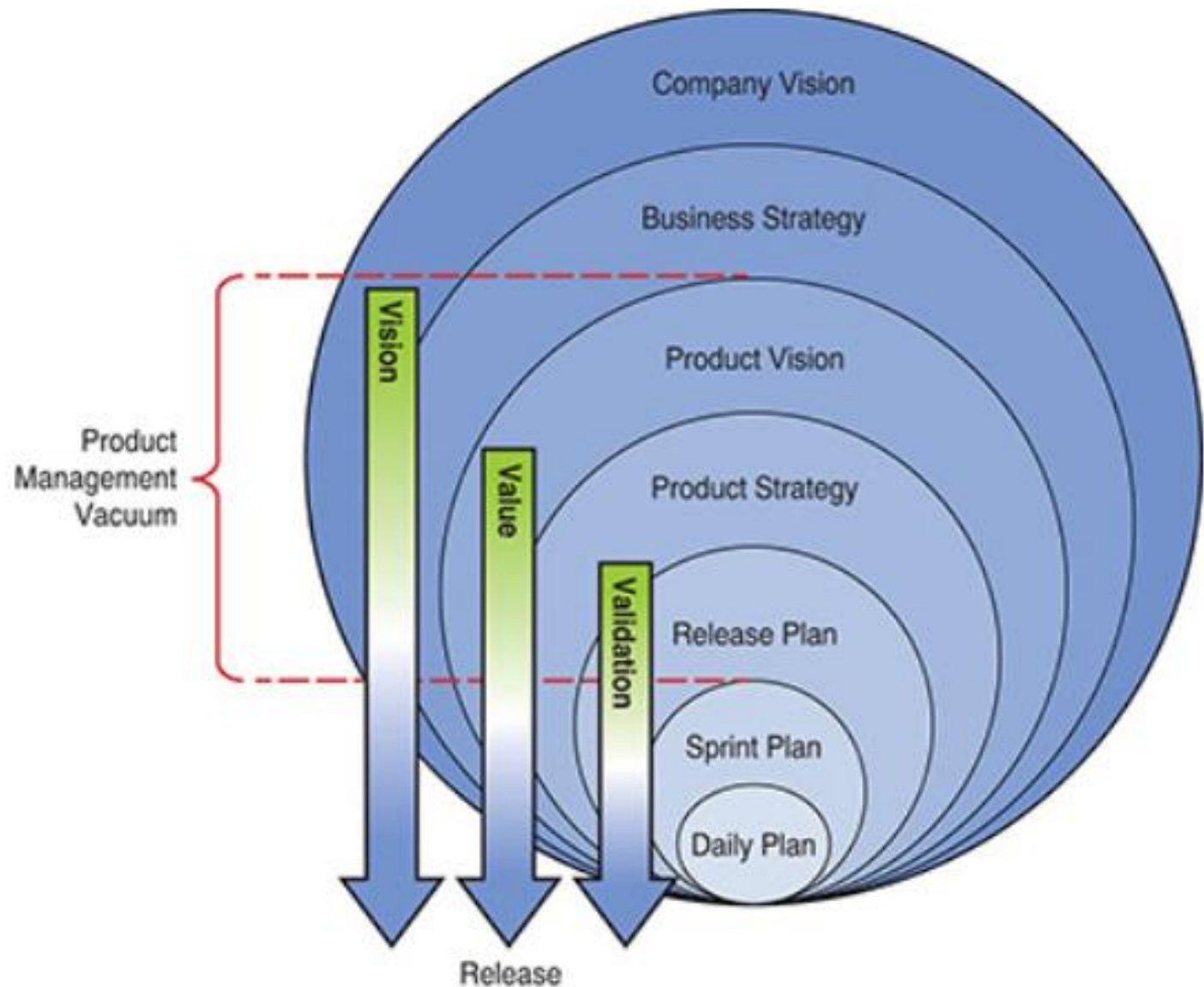
Do not Distribute

Three Vs

From the
Business-side



How Three Vs fill the Product Management Vacuum



Vision

- Vision creates *Transparency*
- The successful business representative establishes a clear product vision for her team, much like a military commander establishes clear intent for his subordinates.
- Doing so allows subordinates to act without direct orders in necessary to carry out the customers intent.

Product vision

- The product vision anchors everything in the process.
- This vision creates transparency because it forms the basis for all following conversations, leading to a common understanding for why you are building the product and what your customers' needs are.

Do not Distribute

Well-communicated vision

- A great and well communicated vision is paramount for a successful launch.
- You need to communicate the product vision again and again to keep everyone on board and honest.
- Never forget that this vision represents the voice of the customer.
- If you stop listening to your customers, the resulting product will be of less value or even alienate them.

Value

- Defining **value** provides you with something to **Inspect**.
- Imagine the vision as a long thread.
- Value is the individual pearls you attach as you progress. Vision provides a foundation and direction, but without the pearls attached to it, there is no value.
- A Team's job is to identify and then attach pearls (value) to the thread (vision).

So how do you measure value?

Do not Distribute

Prioritize

- “If you could have only one thing, what would it be?” is often a good opening question when identifying the most valuable.
- If everything is important, then nothing is.

~ *Patrick Lencioni*

Powerful Questions: importance and priority (1/5)

If we were paying for this Sprint with our own money, what work would give us the highest chance to get that money back?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

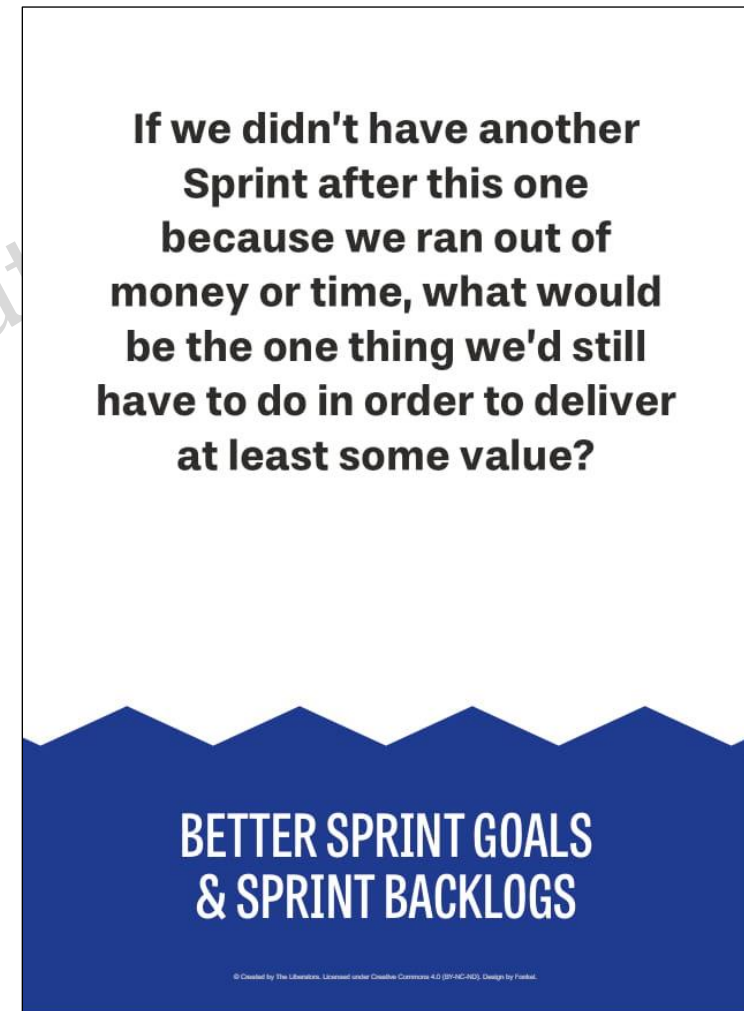
© Created by The Liberator. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

If we suddenly had half the team available and we could only do half the work required for the Sprint Goal, what should absolutely be on our Sprint Backlog to still be okay with the outcome? What could we let go of for now and return to later?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Liberator. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

Powerful Questions: importance and priority (2/5)



Powerful Questions: importance and priority (3/5)

If we just canceled the next Sprint and went on vacation, what would be inevitably lost or become much harder later?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Liberator. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

**Which steps are required to achieve this Sprint Goal?
Which are the least required or could we do without if we really had to?**

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Liberator. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

Powerful Questions: importance and priority (4/5)

If there's an 'AND' in the Sprint Goal: Which would you naturally do first if you had to choose? What is irrevocably lost if we do that thing first, and the second thing in another Sprint?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Librarians. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

What would need to happen while working on this Sprint Goal that would be cause for celebration?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Librarians. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

Powerful Questions: importance and priority (5/5)

What worry about our product is keeping you up at night? What can we build or test this Sprint to make you sleep a bit better?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Librarians. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.

In terms of value and learning about what else is needed from us as a team, what is the worst way to spend the upcoming Sprint? What should we focus on this Sprint to prevent that?

**BETTER SPRINT GOALS
& SPRINT BACKLOGS**

© Created by The Librarians. Licensed under Creative Commons 4.0 (BY-NC-ND). Design by Finkler.