

Agile ORLANDO 2023

JULY 24-28

PRESENTED BY



JOIN US TODAY!

#AGILE2023

LEARN MORE



Tim Ottinger

More Than “Agile”



Agile ORLANDO
JULY 24-28 **2023**

Tim Ottinger

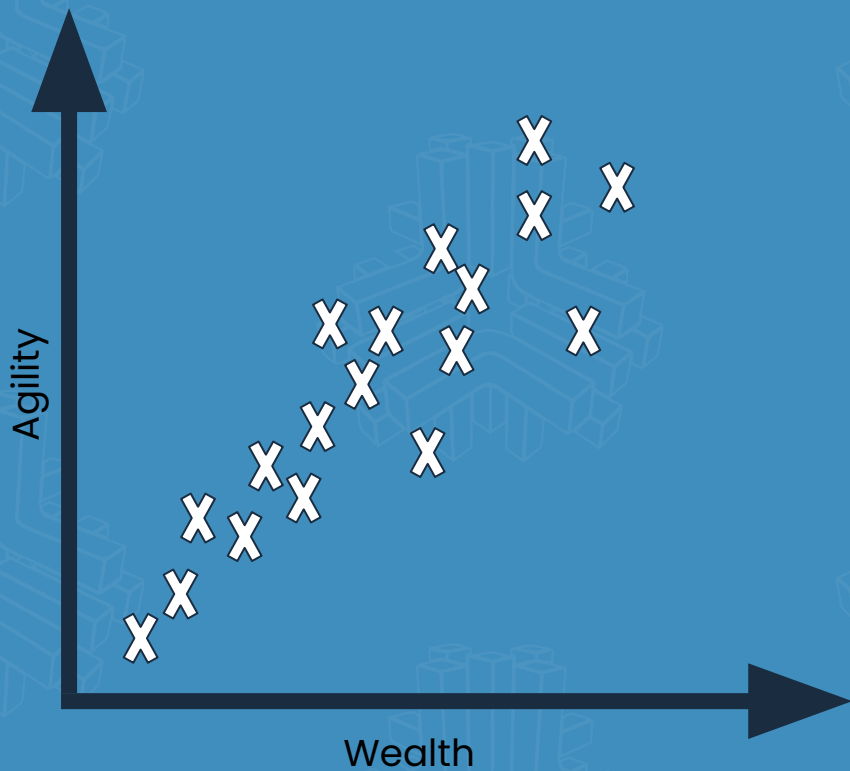
More Than “Agile”

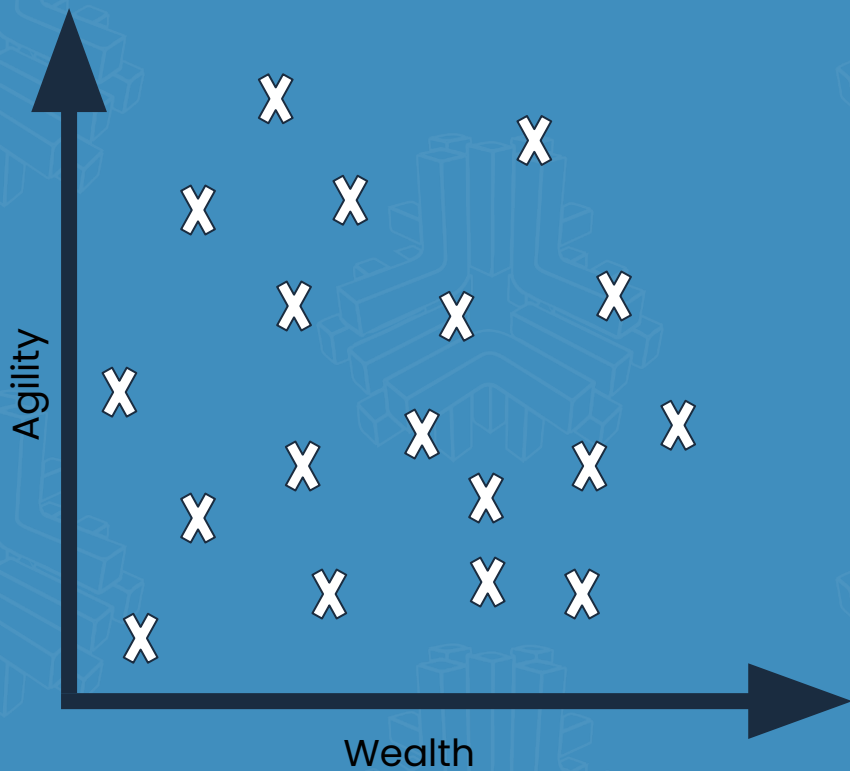


PRESENTS

Agile ORLANDO
JULY 24-28 **2023**

#AGILE2023





“Why do the wicked prosper?”



“It rains on the just and the unjust”

Note: where and when this was written, rain was considered a blessing rather than a fun-spoiler.



But agility is super-important!



That's just,
like, your
opinion, man.

Was it Agile?

Do we know of cases where it definitely was **agile practices *alone*** that decided success or failure of a company or product?



(I suspect not; explanations forthcoming)

Correlation is not Causation

**We are
successful and
we do X**

*Because X?
or
Despite X?*

Fertile Curiosity Space

contractinginsights

The Three Elements Of A Successful Business

Marketing expert Seth Godin, puts it this way 'A brand's value is the sum total of how much extra people will pay, or how often they choose, the expectations, memories, stories and relationships of one brand over the alternatives.'

Company	Is There a "Central" Methodology?	What Project Management "Methodology" Is Typically Used for Engineering Projects?	Who Typically Leads Engineering Projects?
Amazon	No, teams can choose	Plan (6-pager)->Build (iterate)->Ship	Tech lead
Apple	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead
Datadog	No, teams can choose	Plan (RFC)->Build (iterate)->Ship	Tech lead or an engineer
Facebook	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead or an engineer
Google	No, teams can choose	Plan (Design Doc)->Build (iterate)->Ship	Tech lead or an engineer
Netflix	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead or an engineer
Shopify	No, teams can choose	GSD (Get Shit Done, 6-week cycles)	Tech lead or an engineer
Spotify	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead or an engineer
Uber	No, teams can choose	Plan (ERD)->Build (iterate)->Ship	Tech lead or an engineer

11 Reasons Why Companies Succeed and 5 Reasons Businesses Fail



by Yun-Fang • July 29, 2022 RUNNING A BUSINESS

SHARE



Hero Executives



Vision

Drive

Ambition

Genius

Imagination

Character

Boldness

Leadership

Inspiring

Planning

Ethics

Governance

But then...

Al Lewis

CEO performance doesn't matter very much

Published: May 22, 2013 at 9:31 a.m. ET

By Al Lewis

59

Commentary: Paying for 'performance' is crazy, professor says

CEO effect on firm performance mostly due to chance

Date: October 22, 2015

Source: Texas A&M University

Summary: A researcher calls into question the common notion that CEOs have a large effect on firm performance. Instead, he suggests that most of the performance attributed to CEOs could actually be due to chance.

Many business scholars have grown skeptical of the idea of chief executive as superhero. Cutting-edge research reveals that while some CEOs clearly do make a big difference, many are merely the most visible cogs in complex machines.

Forbes

were in accounting, operating, and stock price. The study also found that supposedly important measures like product service and quality, customer service, workplace safety, and innovation *aren't included in more than 95% of CEO evaluations.*

But that's EXECUTIVES not Product Managers.

Project Managers are as Likely to Fail as Succeed

There has already been a lot of controversy about the project manager's role in agile projects. In a related post, the [Myth of the Agile Project Manager](#). Most serious Agile and Scrum project managers add unnecessary overhead and create waste.

[Home](#) > [Agile Methodology](#) > [Project Managers Fail to Help Software Projects \(Standish Group Chaos 2020\)](#)

Project Managers Fail to Help Software Projects (Standish Group Chaos 2020)

We have our opinions, of course...

Our success is due to our choice to use

Informix 4GL!

SW Director for a company whose income came almost entirely from a COBOL mainframe app, and whose front-end developers had largely converted to C and C++.

True Story



Leadership

Finance

Market Fit

Competitive Forces

Marketing

Reputation

User Experience

Network Effect

Halo Effect

The MAGIC COOKIE...

Our recipe genius has invented a cookie that has:

- A pleasing appearance
- Agonizingly delicious aroma
- A flavor like a baby's laughter
- Raving fans



The MAGIC COOKIE...

- Profits are US\$3B per year and growing.
- Costs are *only* a few hundred million a year.



The MAGIC COOKIE...

You are now managing software development for

- logistics
- mobile app
- point of sale



The MAGIC COOKIE...

WHY do you think it is crucial to *agile software development*?



The MAGIC COOKIE...

Does your argument for agility
pass the magic cookie test?



The MAGIC COOKIE...

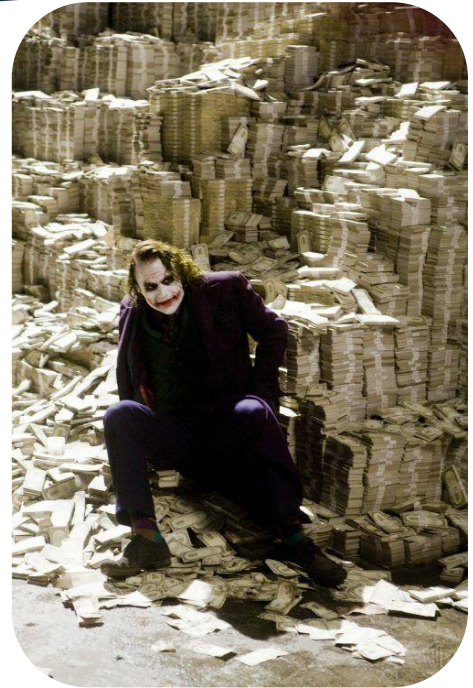
Efficiency matters most
when margins are tight.



The Dominant Player...

“If our company stopped making money tomorrow, it would be 10 years before anyone would notice”

Name withheld



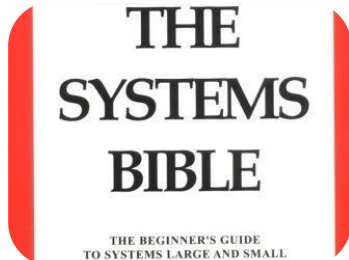
“If you’re not agile, you’ll fail!”

DENIED

- many non-agile companies are successful
- many companies had success before going agile
- being agile is not table stakes

But... the *sheer efficiency!*

Complicated systems seldom exceed five percent efficiency.



Gall, John.
SYSTEMANTICS. THE SYSTEMS BIBLE

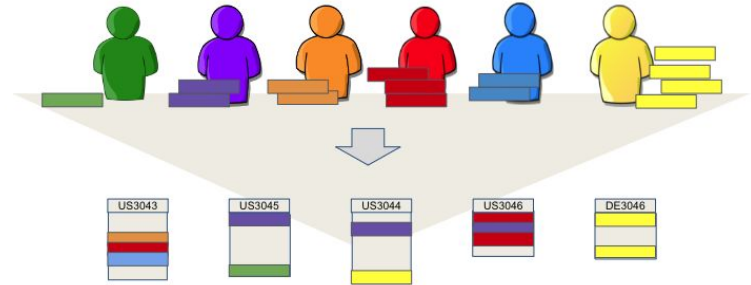
But...agile is simple (which is why it's hard)

What most of us are calling “agile” is a rather complicated system of tickets and status tracking.



Scatter-Gather

Tim Ottinger

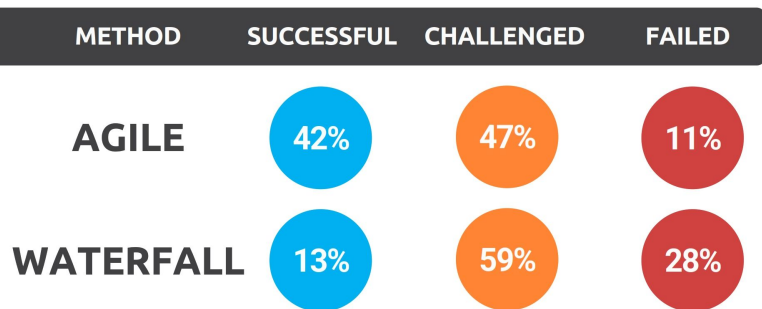


Simple!

We've run entire projects on index cards.



PROJECT SUCCESS RATES AGILE VS WATERFALL



WWW.VITALITYCHICAGO.COM

Source: Standish Group Report 2020

For some reason,
agile seems to help.

Why?
How?

Let's Dig Deep

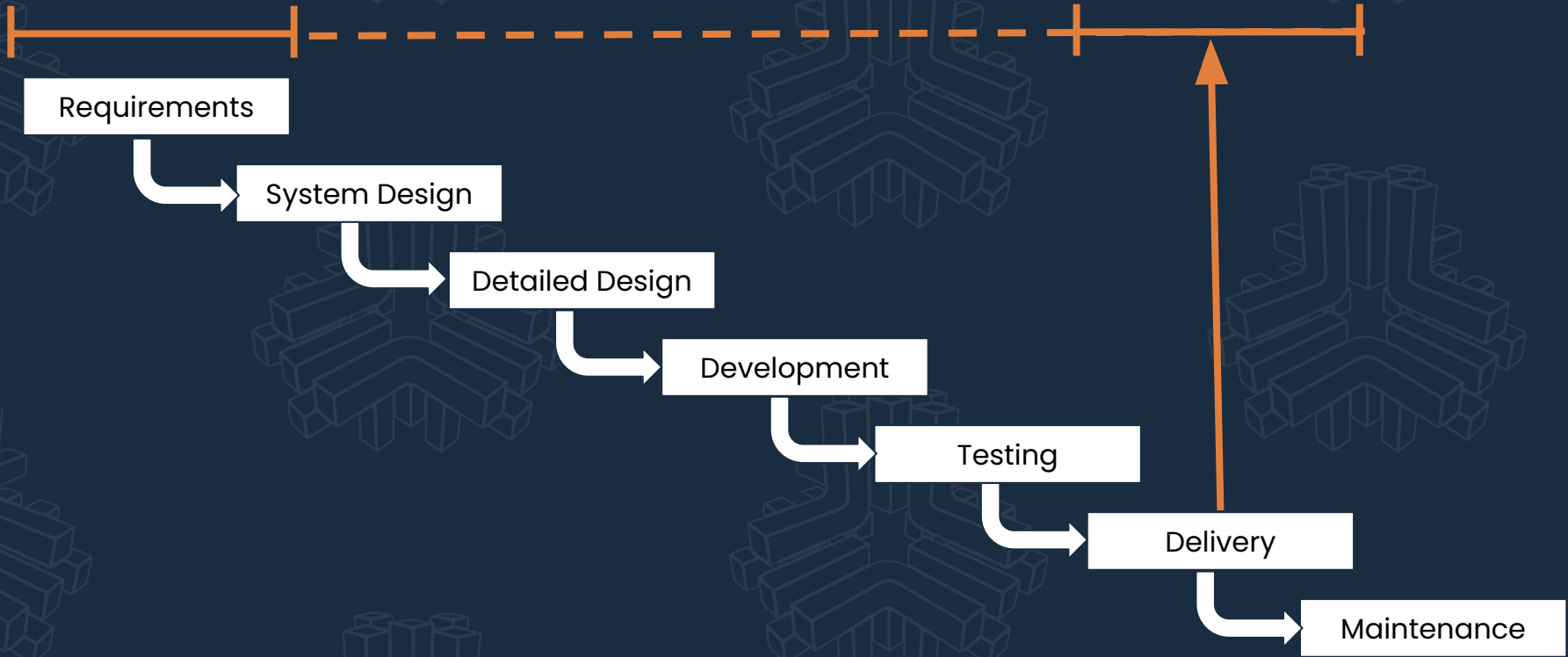


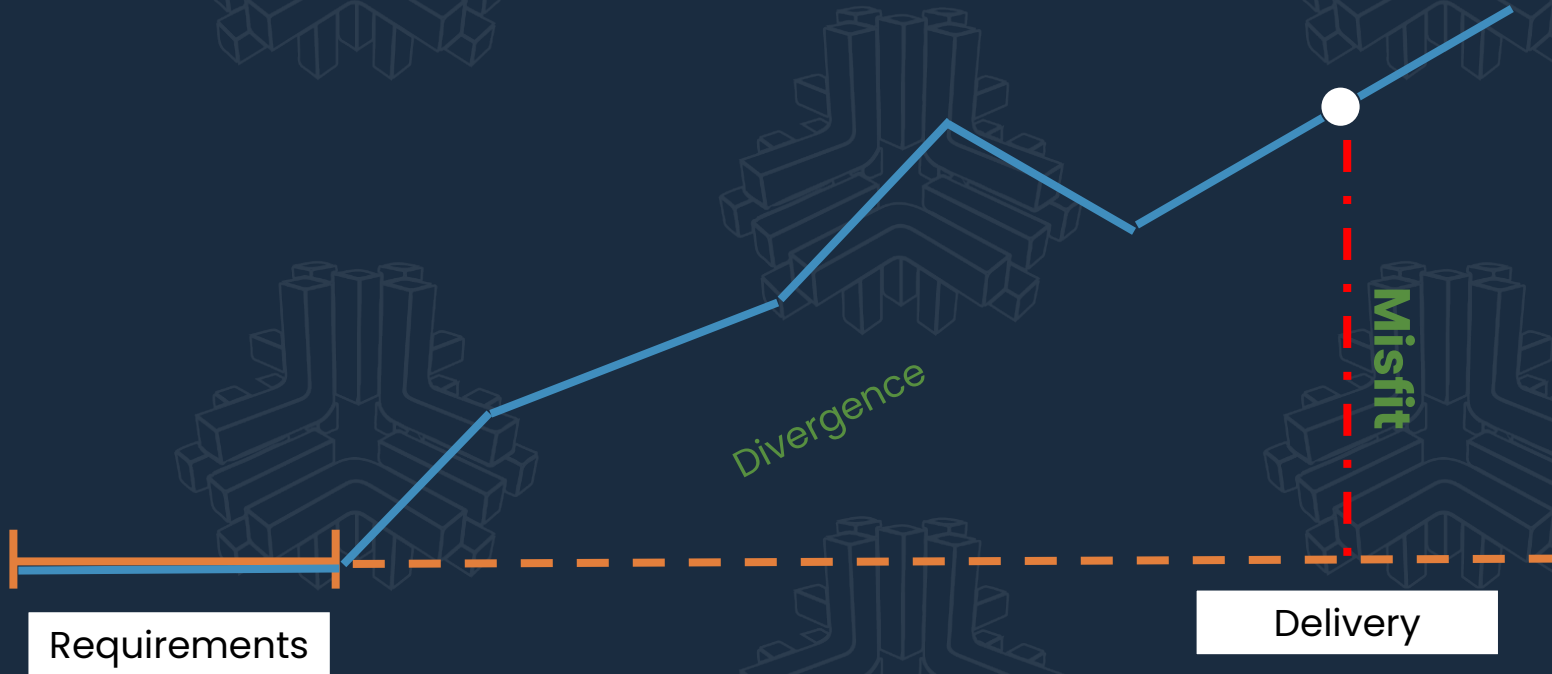
Why did we start doing agile to begin with?

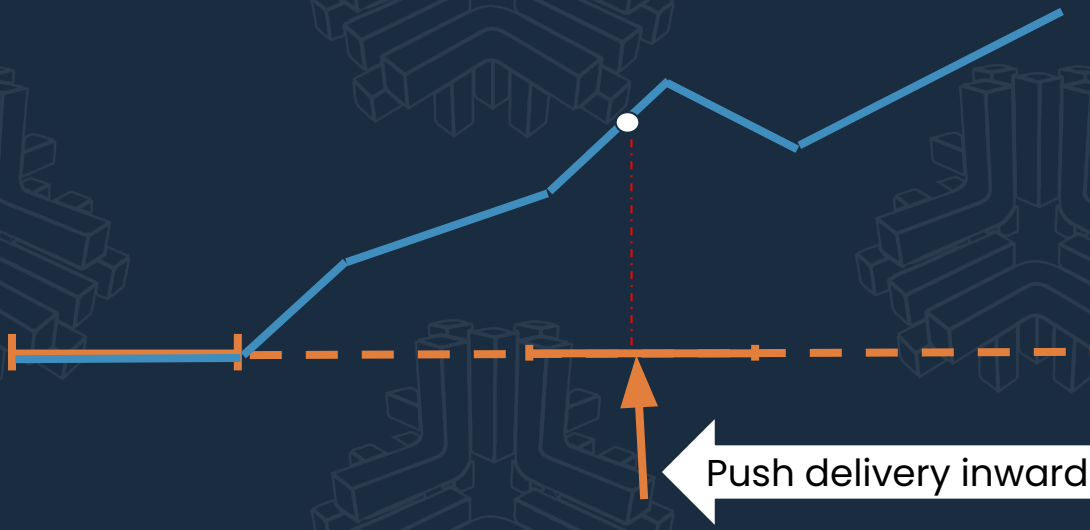
Why does it matter?

Divergence





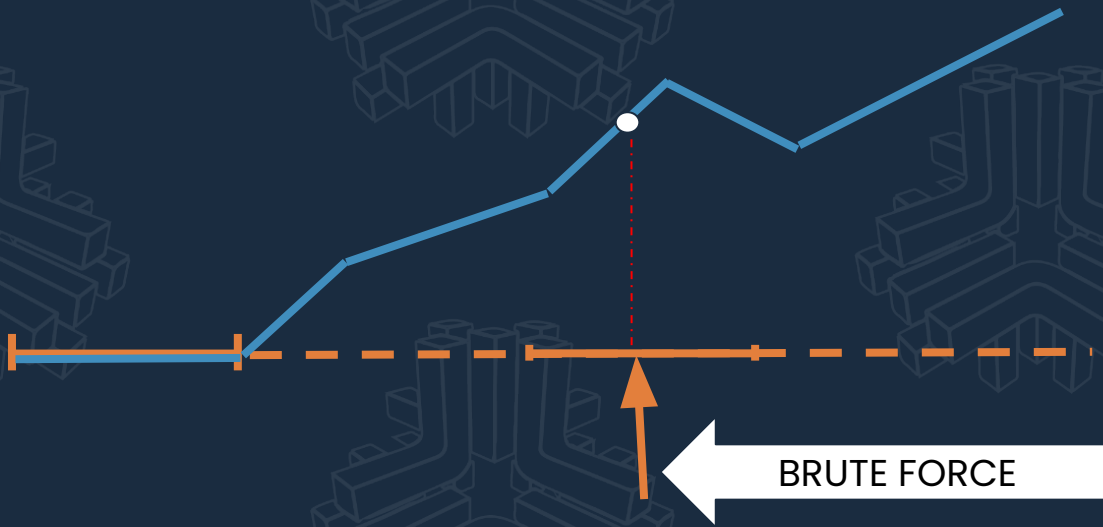




Deliver the software while they still need it!

Push delivery inward!

Divergence!



Surely the answer is to get it *all* done so fast that the divergence won't get out of hand!

Divergence!

Bigger Teams

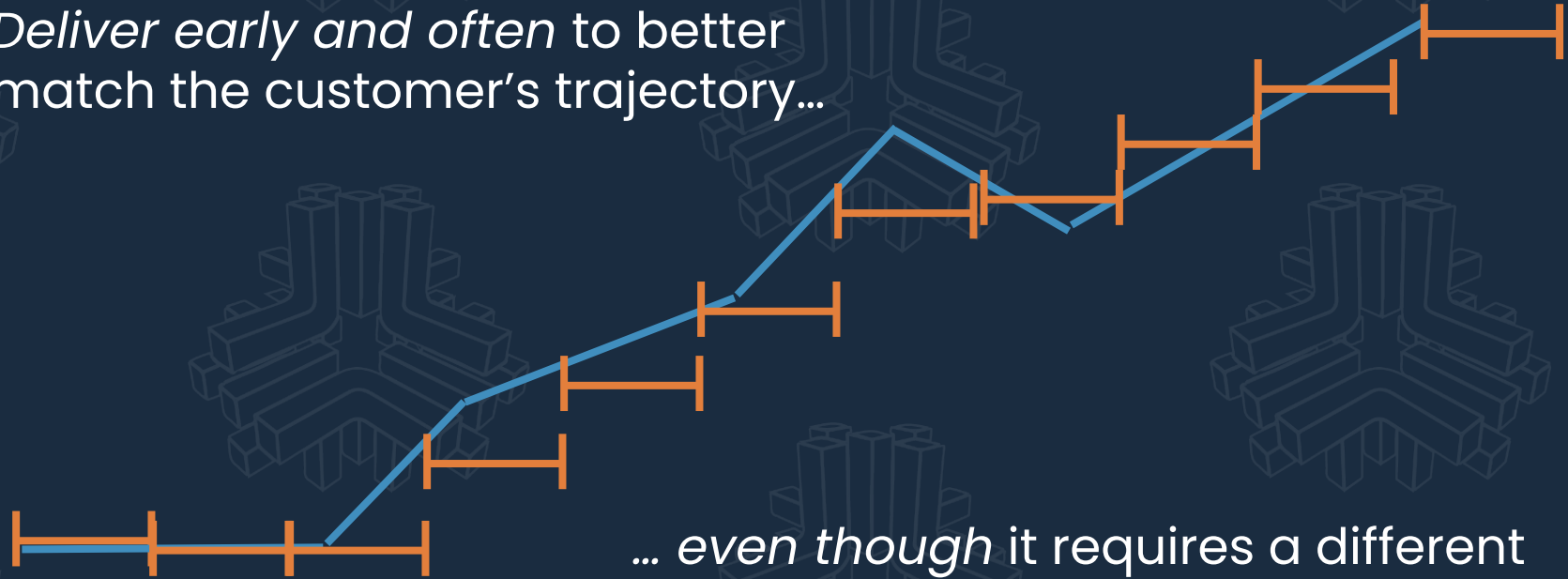
4GL

CASE

Crunch Mode

OR

Deliver early and often to better match the customer's trajectory...



... even though it requires a different set of skills and structures.

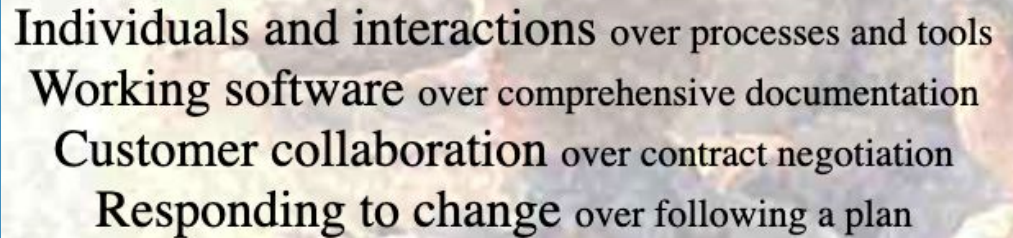
New skills

Close customer contact

Evolutionary Design

Frequent Delivery

Technical Excellence



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

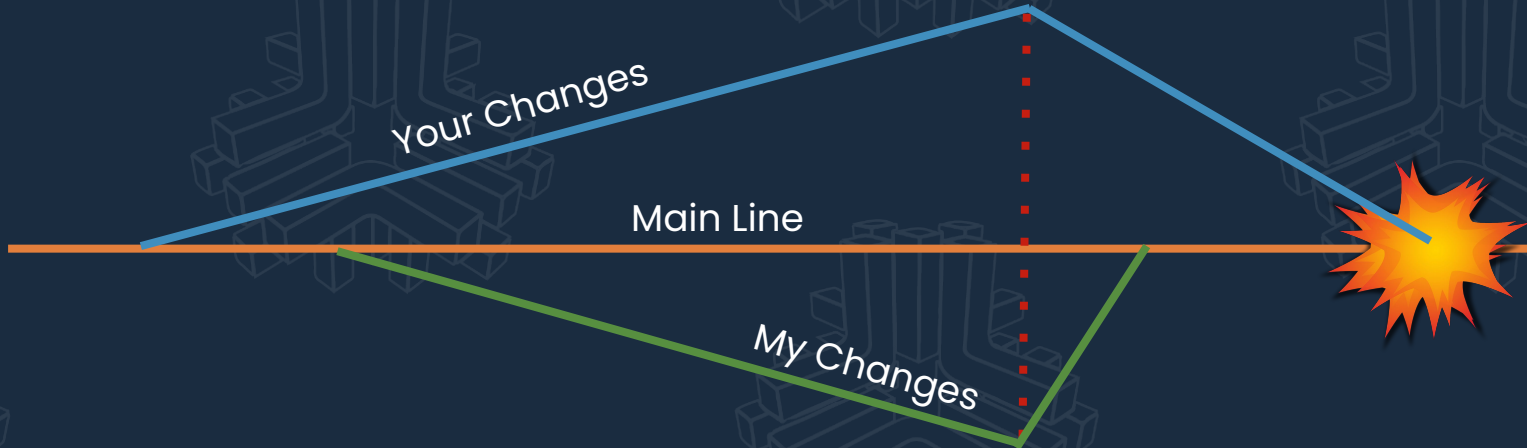
It should have been this way all along.



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

Why is CI so important?

Divergence!



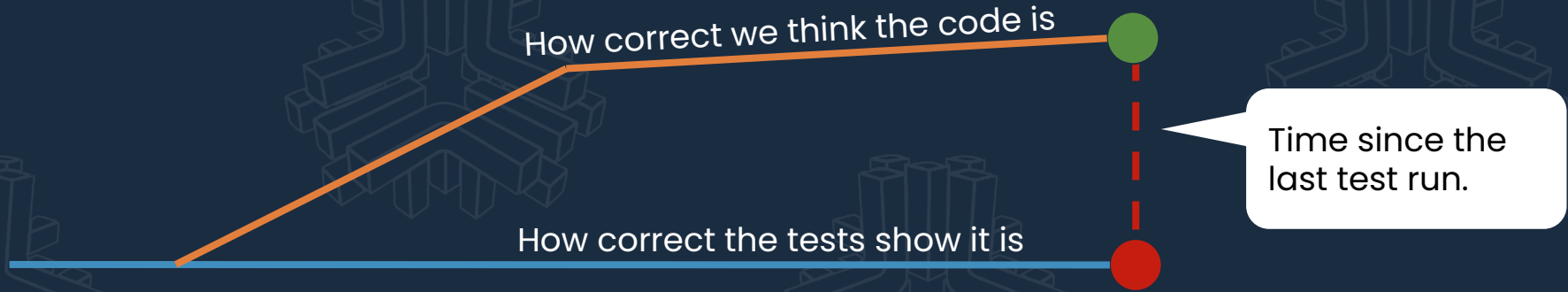
Why Refactor?

Divergence!



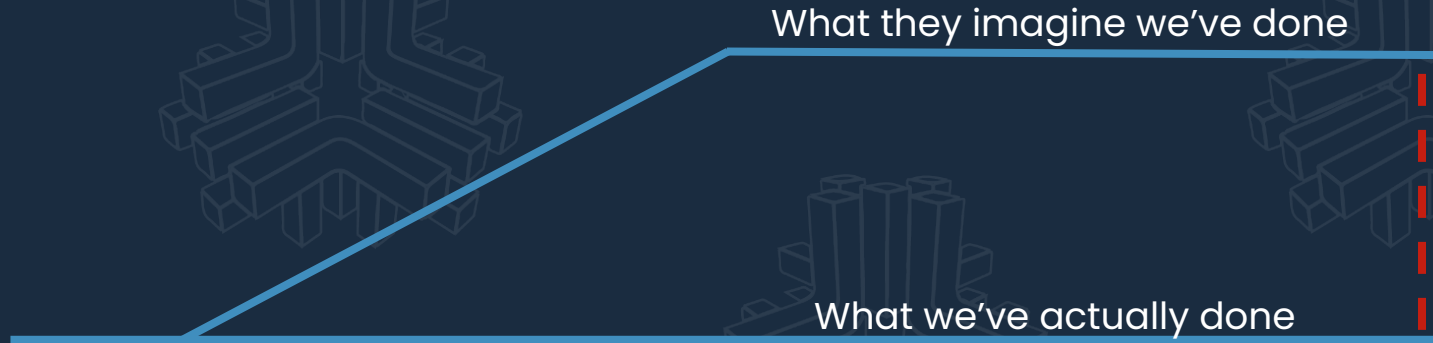
Why test continuously?

Divergence!



Why frequent deployments?

Divergence!



Why daily checkpoints?

Why group planning?

Why retrospectives?

Why cross-functional teams?

Why keep backlogs short?

Why specify by example?

Why use automated tests?





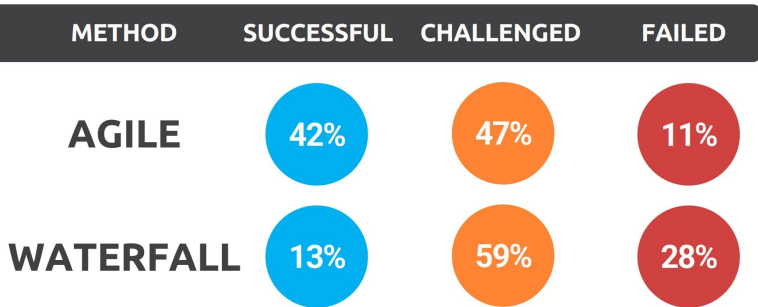
Why individuals and interactions over processes and tools?

Why working software over comprehensive documentation?

Why customer collaboration over contract negotiation?

Why responding to change over following a plan?

PROJECT SUCCESS RATES AGILE VS WATERFALL

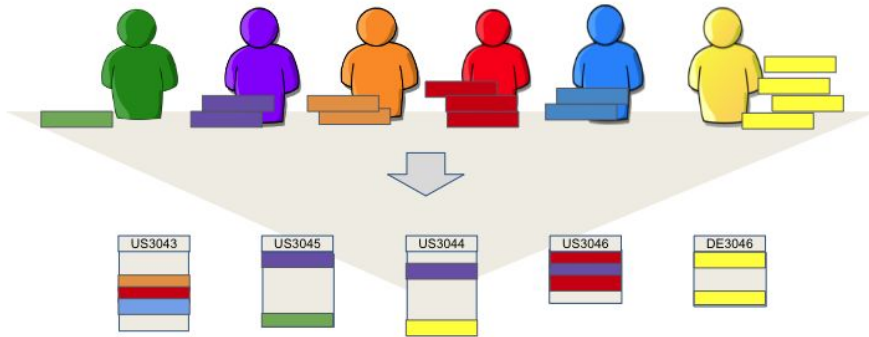


WWW.VITALITYCHICAGO.COM

Source: Standish Group Report 2020

This might help explain
*29% more "success" and
12% less "challenged"*

Scatter-gather describes a way of operating that has diverged from agile intentions and practices.



**How can our agile practice(s)
evolve to be more helpful?**

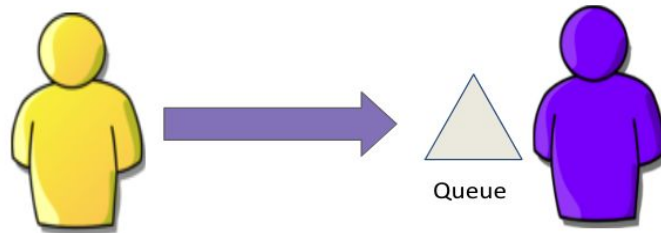


Advances are made by fighting **divergence** with ever more incremental and iterative development.

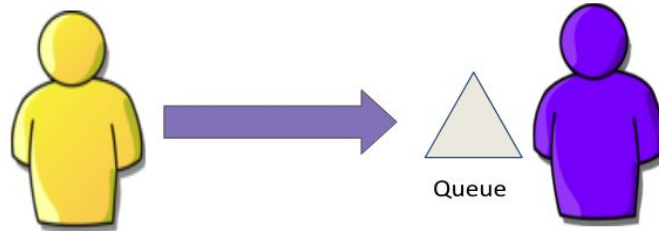


Queuing Systems and Flow

Wherever there is a handoff, there is a queue.
Queue time often dwarfs developer time,
delaying delivery and feedback.

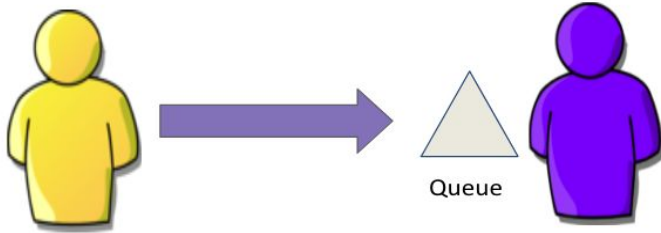


Do people on both sides of the handoff have a common understanding and expectations for the work products?

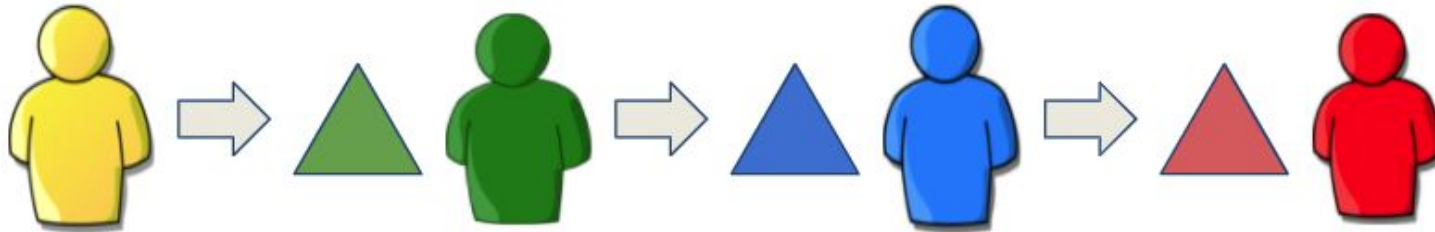


Queuing Systems and Flow

Divergence?

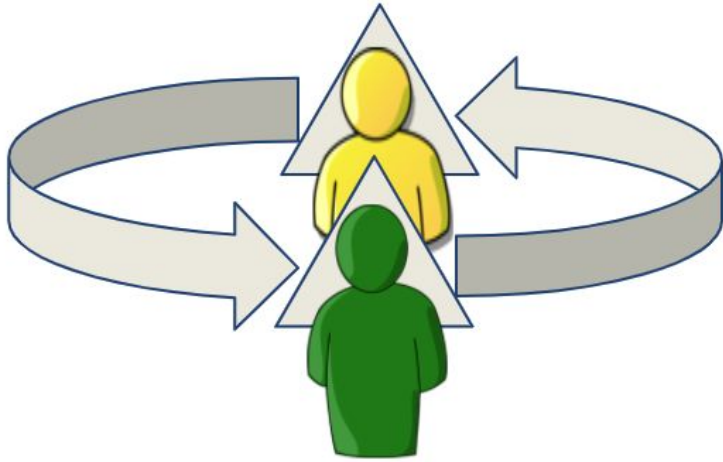


Big companies have pipelined processes with many handoffs.



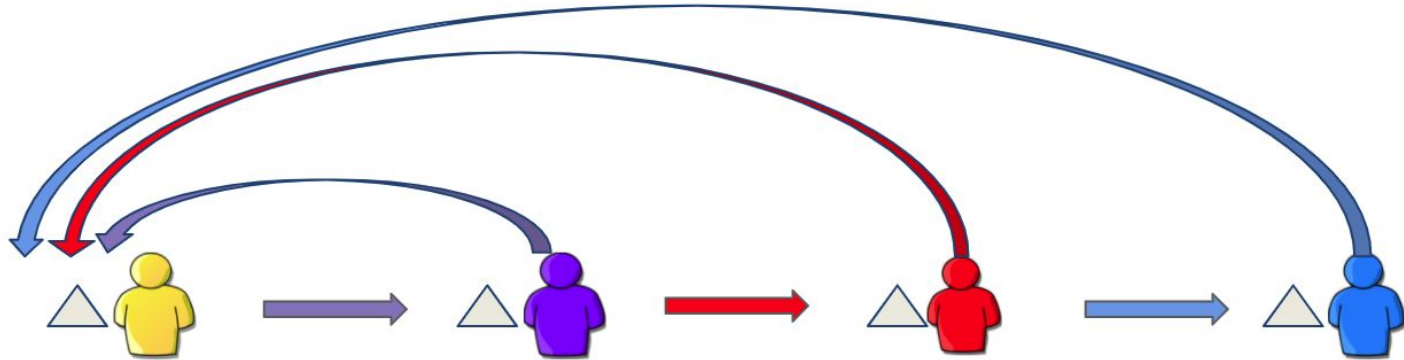
Divergence!

Approval handoffs ensure that things sometimes loop back.



Loops Are Nested

Divergence!



Management As A Service

It is incredibly hard to keep everyone on the same page, working toward the same goals, up-to-date on the same priorities!



It is too much to expect of any person.

Management As A Service

Managers have influence over the 5 Ts.

- **Time:** schedules
- **Talent:** who gets to work on teams
- **Target:** what we are working toward
- **Treasury:** what we can spend
- **Trust:** how to move politically in the organization.



Management As A Service

- **Time:** schedules
- **Talent:** team composition
- **Target:** what we are working toward
- **Treasury:** what we can spend
- **Trust:** how to move politically in the organization.



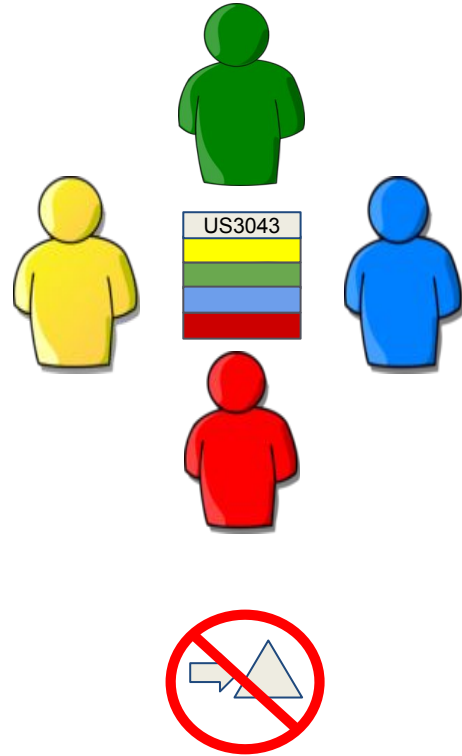
Can we intentionally apply these to push back against **divergence**?

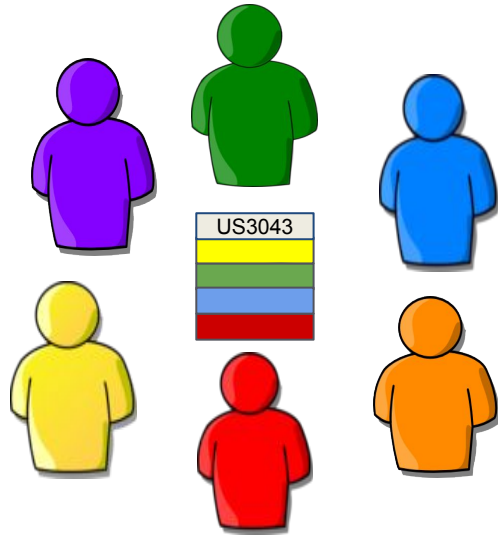


Inspection catches mistakes after they have been made.

What if, instead, we can spot mistakes as we are making them?

Gather people to avoid
queues and handoffs
between them.







Include those who might otherwise reject/loop work back.

Start together
Work together
Finish together





Average Cubicle Cost = \$8000  

80 inch 4K Monitor = \$3799

2:00

No Cubicles!

**Dual
Keyboards**

80 inch 4K Monitors



Organizational Design

Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when, and how.

The best architectures, requirements, and designs emerge from self-organizing teams.

XP teams are *self-organizing* and *cross-functional*. This has two important consequences: first, they're responsible for their own success. This means teams define success (by interviewing stakeholders and sponsors), create plans to achieve success, and execute on those plans without explicit management direction.



Simplest Version

One Funnel
One Team
One Product



Limited opportunity for **divergence** if:

- Funnel is fast-flowing
- Package goes to users
- Dev time is short-ish

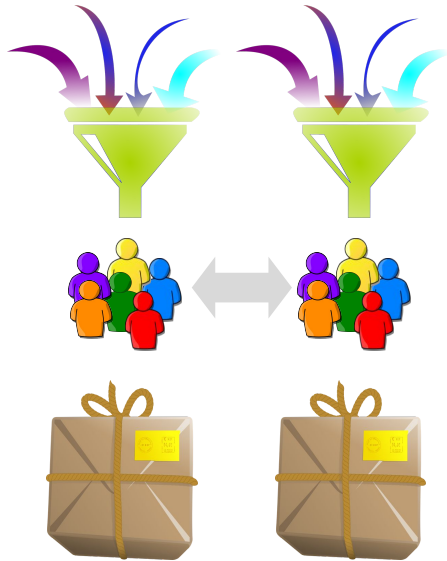


Scale Out

Funnel:Team:Product
Ratio is 1:1:1

Products are isolated and
independent?

Beware dependency!



Permeable Boundaries

People propagate knowledge as they flow freely between co-creating teams.

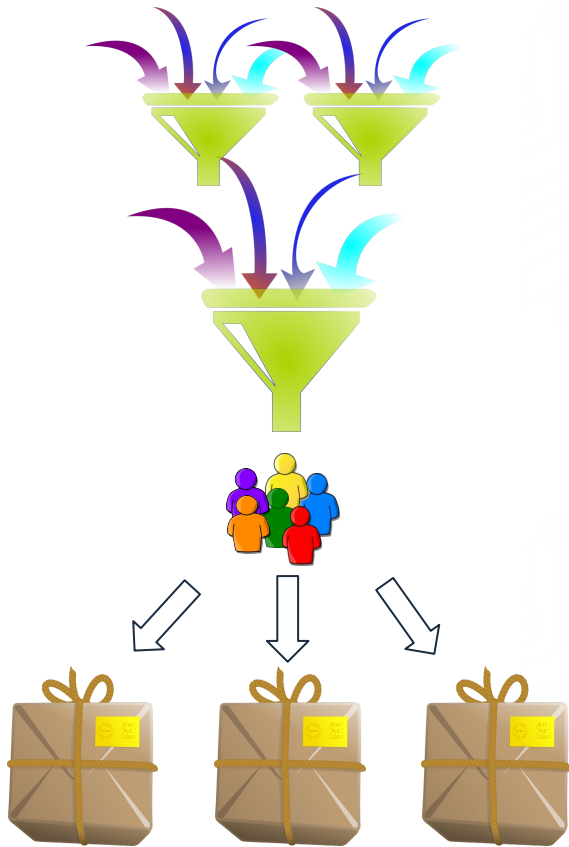


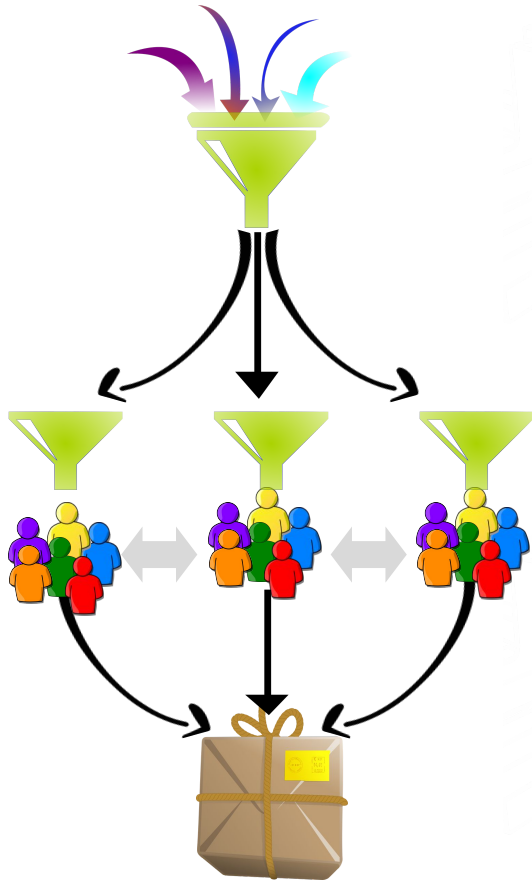
Shared Team

Funnel:Team:Product
Ratio is N:1:N

Scales poorly.
Requires hard prioritization.

Gives product 1/Nth team



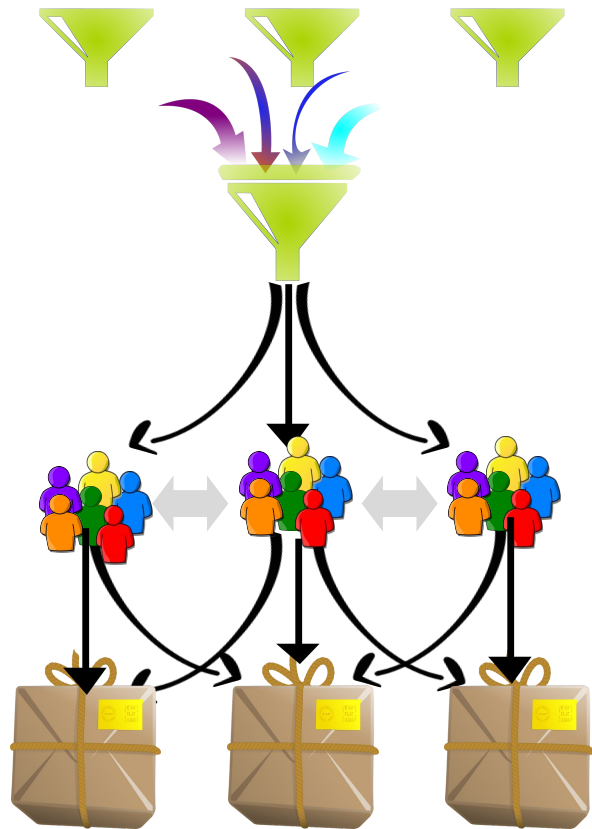


Multiteam Product

Funnel:Team:Product
Ratio is 1:N:1

Somewhat scalable

Increased risk of dependencies,
feature interactions, and
integration failure.



Multiteam MultiProduct

Funnel:Team:Product
Ratio is N:N:N

Multiple organizational
goals satisfied "at once"

Worst code curation, hard
to manage, risky.



Jason Gorman @jasongorman@mastodon.cl... @jasongorm...

Replying to @tottinge

AFAIC, CD *is* Agile. The rest is just Agility Theatre.



3



4



Close, but broader than that.



You can start where you are.

Recognize and stop divergence
whatever your process happens to be.

A little at a time is progress.

You can do it.

Tim "Agile Otter" Ottinger
tottinge@industriallogic.com

