

Lunch & Learn

@ Large Investment Advisory Organization

Exposing Uncomfortable Topics: Errors and Omissions with Scaling

About Gene

Gene Gendel is an organizational design specialist, agile/lean coach and trainer, consultant and adviser to senior leadership. Almost 15 of 20+ years of his professional experience – Gene dedicated to working with companies of various sizes and lines of business, trying to help them improve internal dynamics, organizational structure and and becoming a better place to work. Gene engages at all organizational levels: senior- and mid-level management, teams and individuals. In his work, Gene uses various methods, tools and techniques to amplify learning by others and to ensure that people gain autonomy after Gene “coaches himself out of the job”.

Gene is Certified Agile Coach (CAC), by Scrum Alliance – small elite community of Scrum Alliance Certified Enterprise Coaches (CEC). Today, he is the only CEC who resides in NY State. Gene is also one of the co-creators of Team Level Coaching Certifications (CTC) program for Scrum Alliance.

Gene is also one of very few Certified LeSS Trainers (CLT) that are based in the United States.

He is also well-recognized blogger and publisher. He is the author of Adaptive Ecosystems: Collection of Independent Essays About Agility and co-author of:

- Agile Coaching: Wisdom from Practitioners (free pdf).
- Best Agile Articles of 2017 and 2018



Gene significantly contributes to global and local agile communities, where he influences people by running workshops, webinars, seminars, panel discussions, lunch & learn, coaching retreats, professional gatherings and other events. While teaching and coaching Large Scale Scrum (LeSS) to organizations, Gene has built the biggest in world (today) LeSS meetup community that is used as a free educational hub about LeSS available to many people.

Community Building

Big Apple

BUSINESS **AGILITY**

(a.k.a. "BABA")

~ 500 people



More details on: [ScrumAlliance® Page:](https://www.scrumalliance.org/get-certified/scaling/large-scale-scrum)
<https://www.scrumalliance.org/get-certified/scaling/large-scale-scrum>

Principles | Frameworks | Guides | Experiments

> 4200 people

Join through:

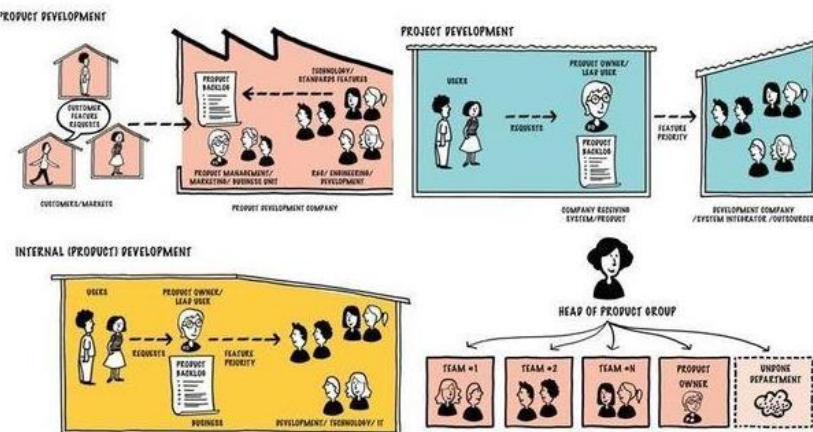
<https://www.keystepstosuccess.com/>
(right navigation)

STAY INFORMED & ENGAGED:

Join Local (NYC) Agile Communities:

- Large Scale Scrum -NYC (LinkedIn)
- Large Scale Scrum – NYC
- Product Management Group – NYC
- Big Apple Business Agility (BABA)
- Agile in Education

Agile News: [Click to Sign Up for Agile Flyer](#)



~ 900 people

Simplifying Overloaded Jargon

Top-3 Abused Words

- 1) Agile
- 2) Enterprise
- 3) Scaling



Avoid This:

When put together, in the same phrase, they become **“Enterprise Agile Scaling”** – what seems to be the most popular and frequently used phrase, nowadays.

Instead of using this phrase obsessively (titles, job descriptions, etc.), please consider the following:

- ❖ *If “Agile” != “Adaptive”, the word is probably misused*
- ❖ *“Enterprise” – does not always mean “huge”. A 50-person company (HR, Finance, Legal, Vendor Management, Business, Operations, IT) – is an enterprise. A 500-person IT Department alone – is NOT an enterprise.*
- ❖ *In order to SCALE goodness, you may need to DE-SCALE (remove) badness. Therefore, by obsessively focusing on scaling, you might be increasing (scaling up) levels of badness in the system (organization) ☹*



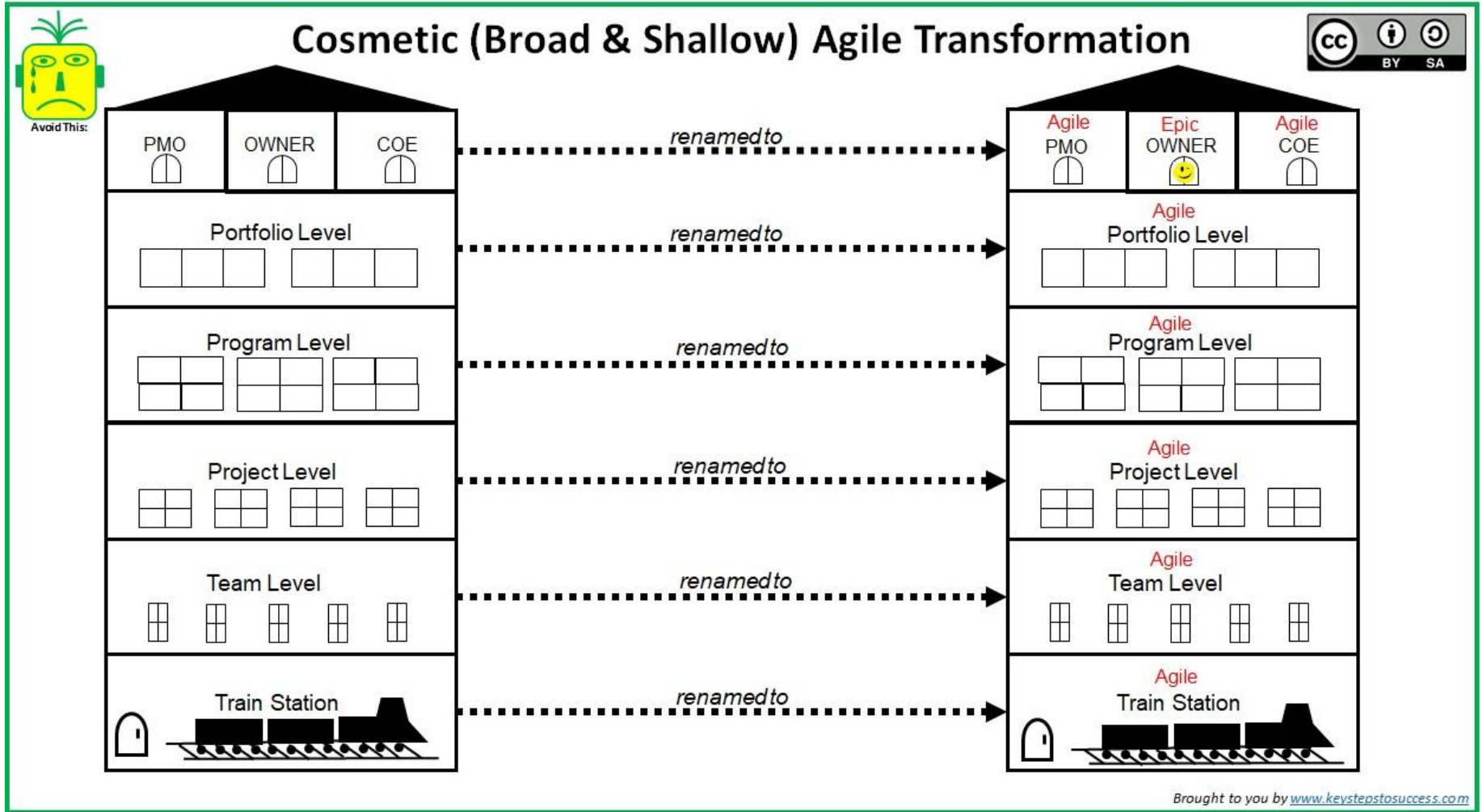
Brought to you by www.keystepstosuccess.com

To Scale Goodness, We Must Have Goodness

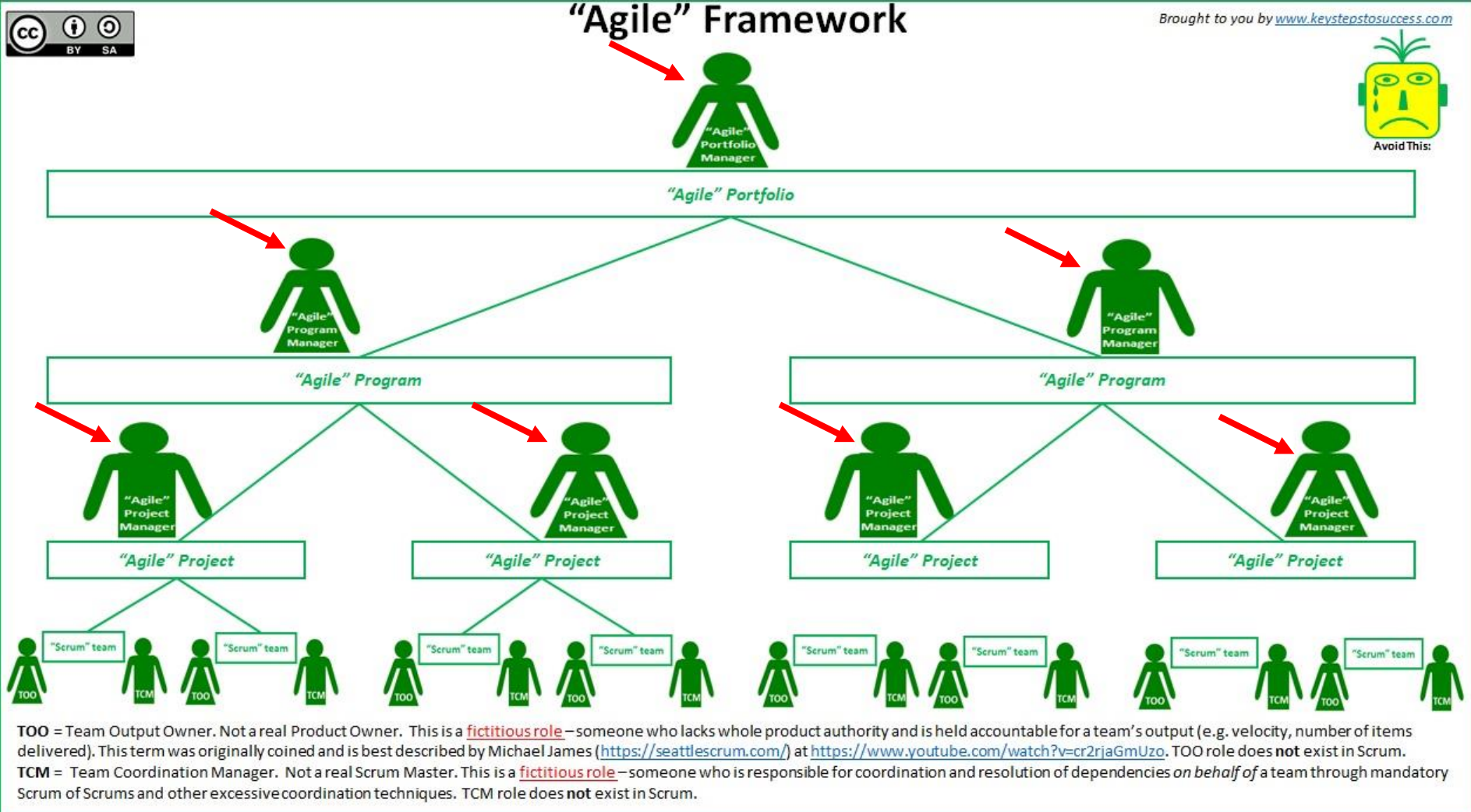
Tip:

Try keeping things simple. Avoid unnecessary complexity. Don't attempt to scale, unless it is absolutely necessary and you are confident that you are scaling goodness.

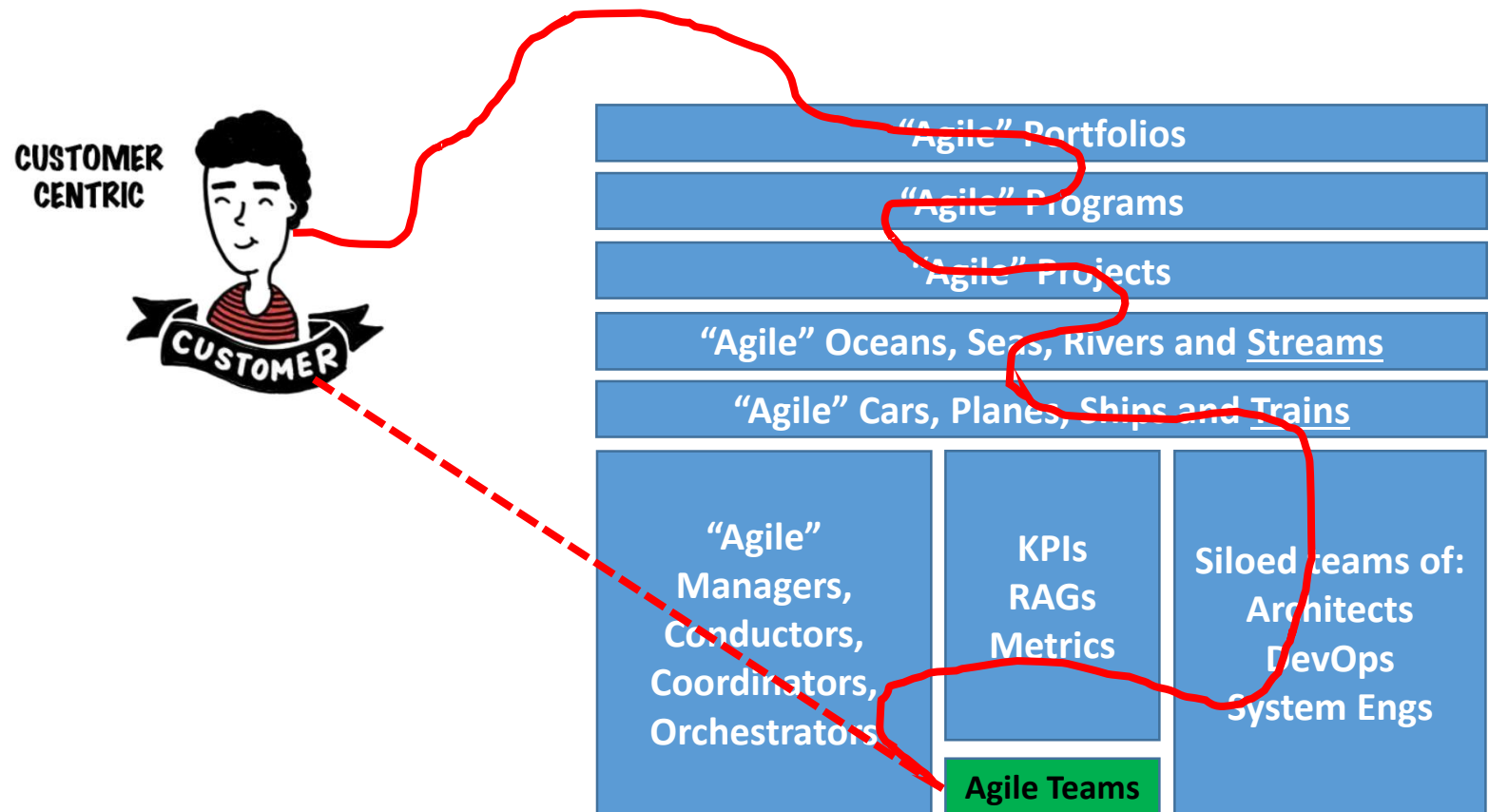
Merciless Relabeling



Merciless Relabeling



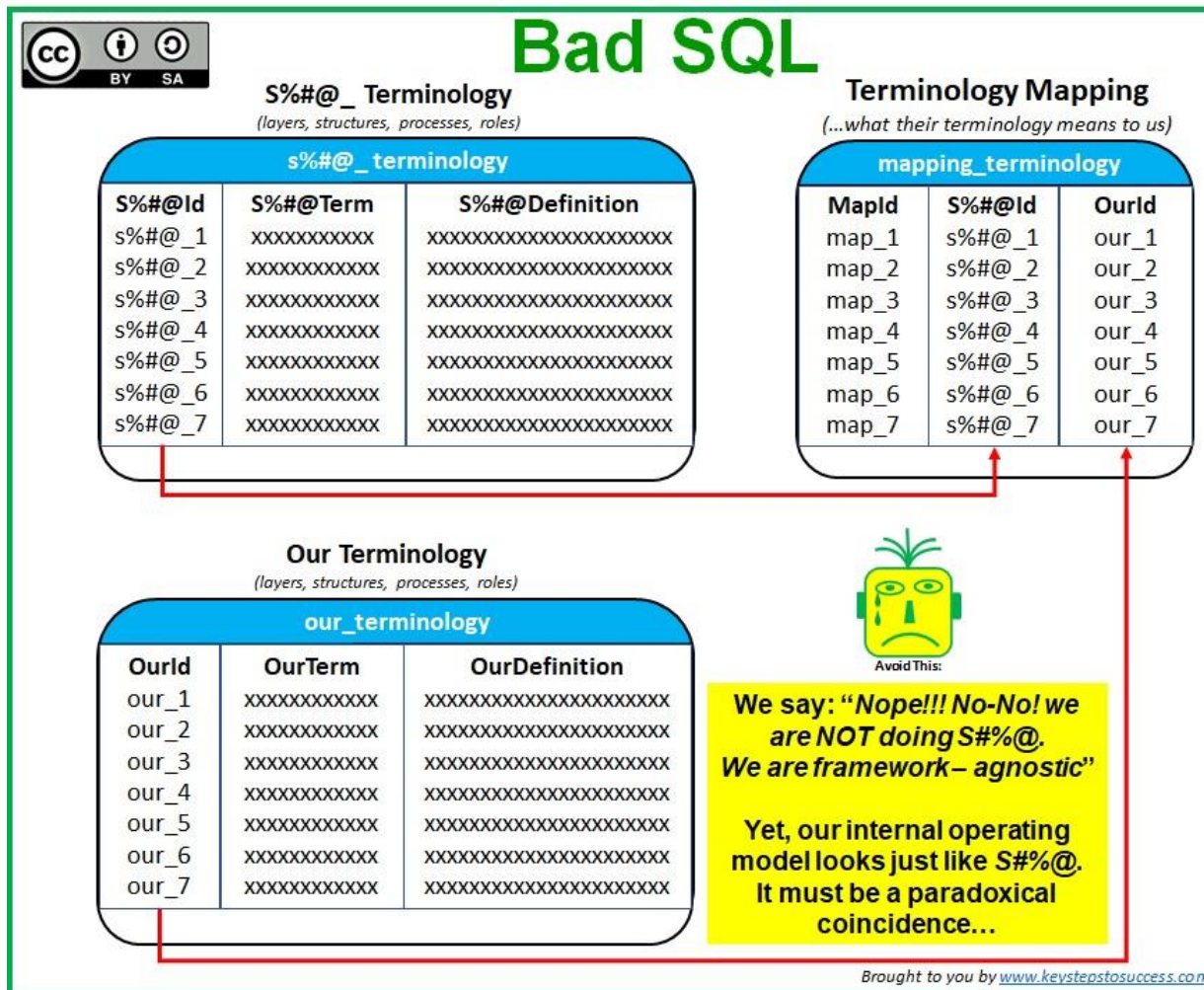
Putting Scrum Teams in The Basement



Version 348.54

- Where is customer – centricity?
- How **loooooong** is a feedback loop between customers and developers?
- How adaptive (agile, fast-reacting) can teams really be, to Customer's requests?
- What has changed from what you saw prior to "installation"?

Defining Our Own Scaling Model



Fake Scaling by Component Teams

Scrum Anti-Patterns

- separate, phase-specific backlogs or single backlog with phase-specific items
- local optimization by single-skill specialists (e.g., PM, BA, QA, Architect, Developer)
- hand-overs, toll-gates, "internal contracts"
- long periods of down-time by specialists, when it is not "their phase" to work
- "water-scrum" / "scrum-fall"
- Very weak Definition of Ready & Done
- PSPI – takes many sprints to produce



More obvious anti-pattern
Less obvious anti-pattern

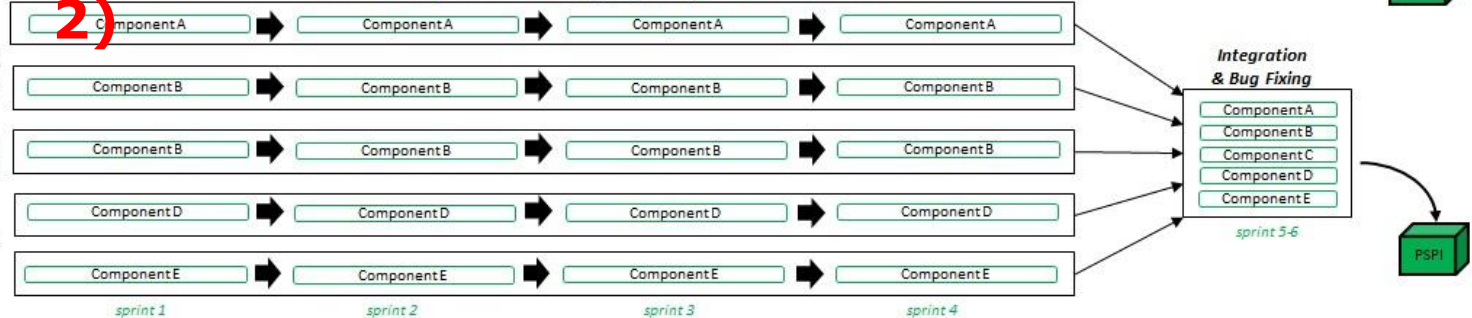
- separate, component-specific backlogs or single backlog with component-specific items
- local optimization by component specialists (e.g. UI/UX, middle-tier, back-end, web service, architecture)
- hand-overs, toll-gates, "internal contracts"
- multiple non-development sprints needed to integrate all components and fix bugs
- Weak Definition of Ready & Done
- PSPI – takes many sprints to produce

*In Scrum, in every Sprint, a team delivers Potentially Shippable Product Increment (PSPI). **This is fundamental for Scrum.** In order for this to happen, each team must possess all necessary attributes (skills, knowledge, domain expertise) required to get work fully DONE (potentially shippable). This is what makes Scrum - real Scrum. Many teams that lack the required Scrum attributes still attempt to sprint, however, effectiveness of such "sprint-like activities" is significantly reduced. **Not all anti-patterns of Scrum are equally obvious.***

When Waterfall / Sequential Project Phase Team(s) "sprint", it looks like this:



When Component Team(s) "sprint", it looks like this:



Typical Associated Problems

- Product definition is weak. Applications and components that don't have strong customer alignment are treated as products
- "Doing Scrum" efforts are often a result of trying to meet goals of agile transformation (some annual % goals must be met), set at enterprise level
- "Territorial" code ownership
- Top-down, "command & control" governance, with little autonomy and self-management at team level
- Importance of Scrum dynamics and its roles are viewed as secondary to existing organizational structure blueprints
- Too many single-specialty experts and very few T-shaped workers
- No meaningful HR changes to support Scrum team design

Spending Time, Money and Energy On...

Brought to you by www.keystepstosuccess.com



Avoid
(Tool – Driven Transformations)

TDT

Success of agile transformations is inversely proportional to reliance on “agile” tooling solutions (in partnership with heavy “scaling” frameworks) and directly proportional willingness to de-scale and simplify;)

Brought to you by www.keystepstosuccess.com

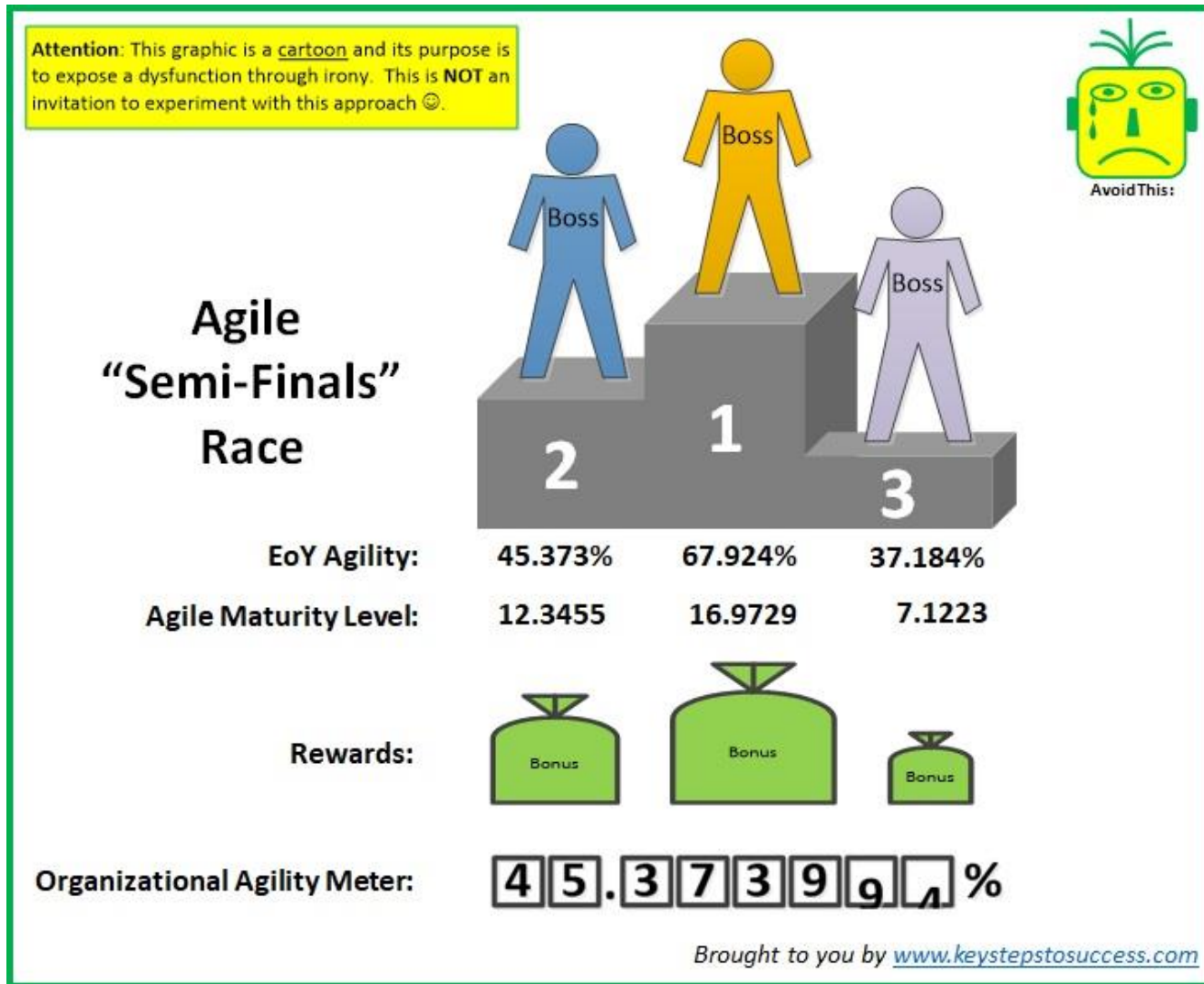


Avoid
(Deck – Driven Transformations)

DDT

Knowledge retention and quality of learning are inversely proportional to thickness of a power point 'agile transformation' deck and directly proportional to font size used in a deck ;)

“My Agile Is More Agile Than Your Agile”



This problem grows exponentially with bad “scaling”

Economic Impact of Bad Decisions

You Pay. They Get Paid.

“Triple Taxation” on Client Companies

Big Consultancies
(supported by
recruiting agencies),
advising of “Tool +
Framework” Big-Bang
Agile Transformations



“Agile” Tools,
supportive of
Big Bang “Agile
Frameworks”

Large,
Commercially
Successful **“Big-
Bang Agile”
Frameworks”**



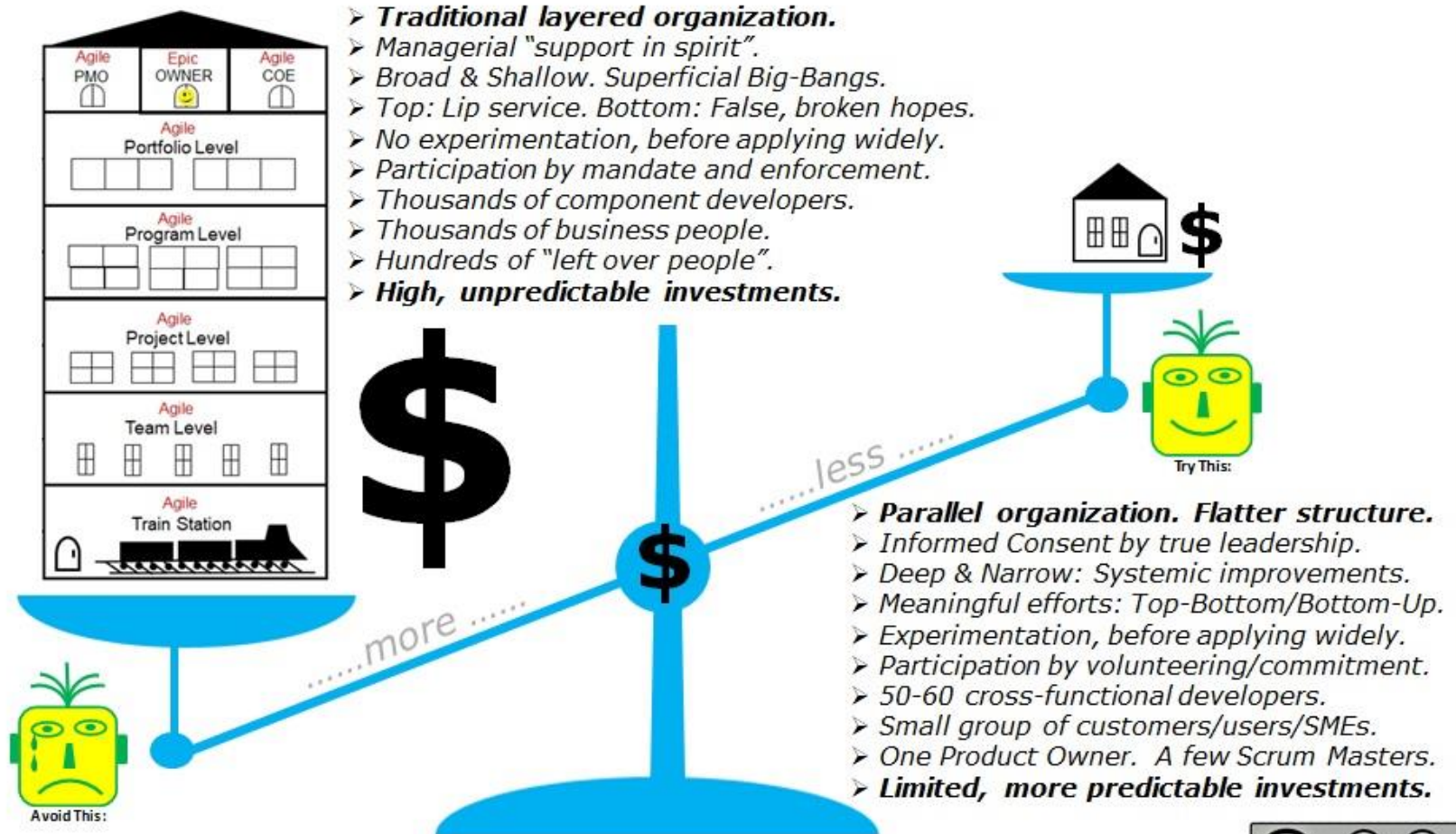
Avoid This:



Brought to you by www.keystepstosuccess.com

The Big Picture of Economics

Economic\$ Of Agile Transformation



Brought to you by www.keystepstosuccess.com



Industrial Model: Making A Profit From Agile

05/05 – LESS TALKS: DAVE SNOWDEN: ANSWERING TOUGH QUESTIONS (Q&A)

🕒 MAY 5, 2020 👤 GENE 💬 LEAVE A COMMENT ✎ EDIT

A great talk today (this is round 2), with **Dave Snowden** (round 1 was on [04/20](#)), who took on some provocative and pretty powerful questions. All points that Dave made were strong. Here is one that resonated really strong (the quote in blue below is semi-transcribed/paraphrased, starting from about 4 min 20 sec in the video recording below):

“...SAFe is perfect for big consultancy firms...

With big consultancies, when the ratio between a principal and a doer (partner and consultant) is up to about from 1:5 to 1:10 – apprentice model.

With ratio of above 1:15 – it becomes an industrial model (you have to “feed” a lot of people), when you get more structured processes and recipes.

This is why big consultancies want high utilization and long-term projects, [using] Six Sigma, BPR, SAP...etc.

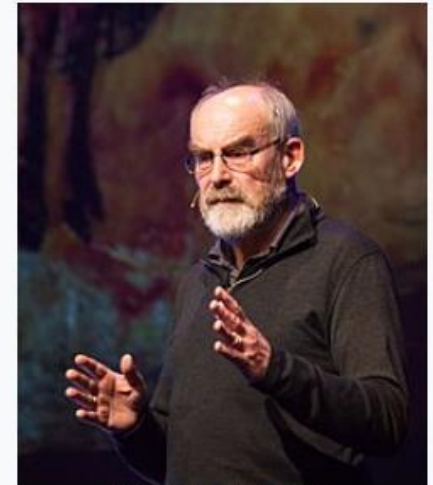
What they like is a massive roll out, with lots of people, over a long period of time.

What they DONT like, are small improvements in the present.

...So you [if you are a client company] are better off working with small consultancies, not big consultancies....”.



David J. Snowden

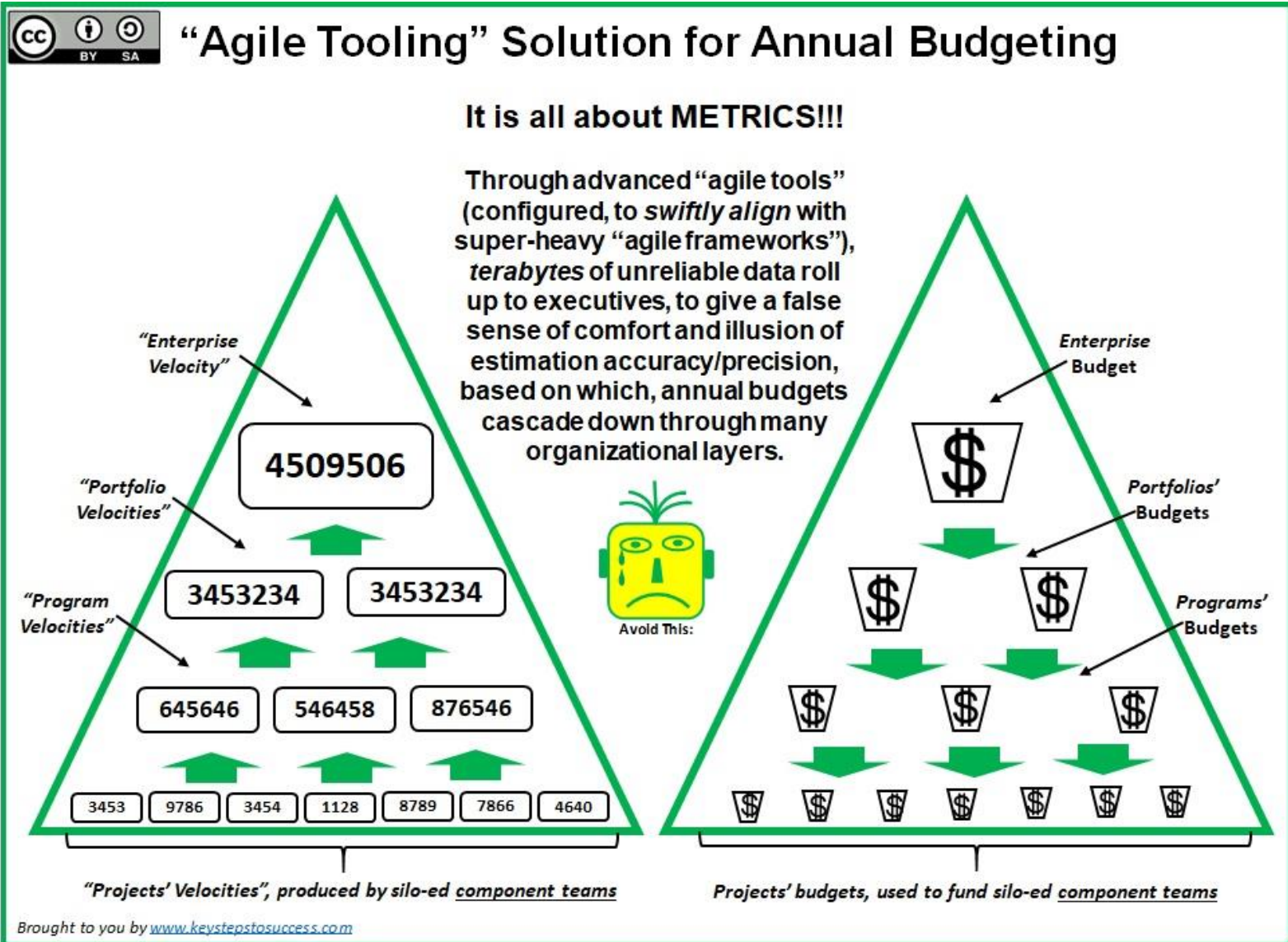


Dave Snowden, 2016

Born	David John Snowden 1954
Nationality	Welsh
Education	BA (philosophy), University of Lancaster, 1975 MBA, Middlesex Polytechnic, 1985
Occupation	Management consultant
Employer	Cognitive Edge Pte Ltd, Singapore
Known for	Cynefin framework
Website	www.cognitive-edge.com

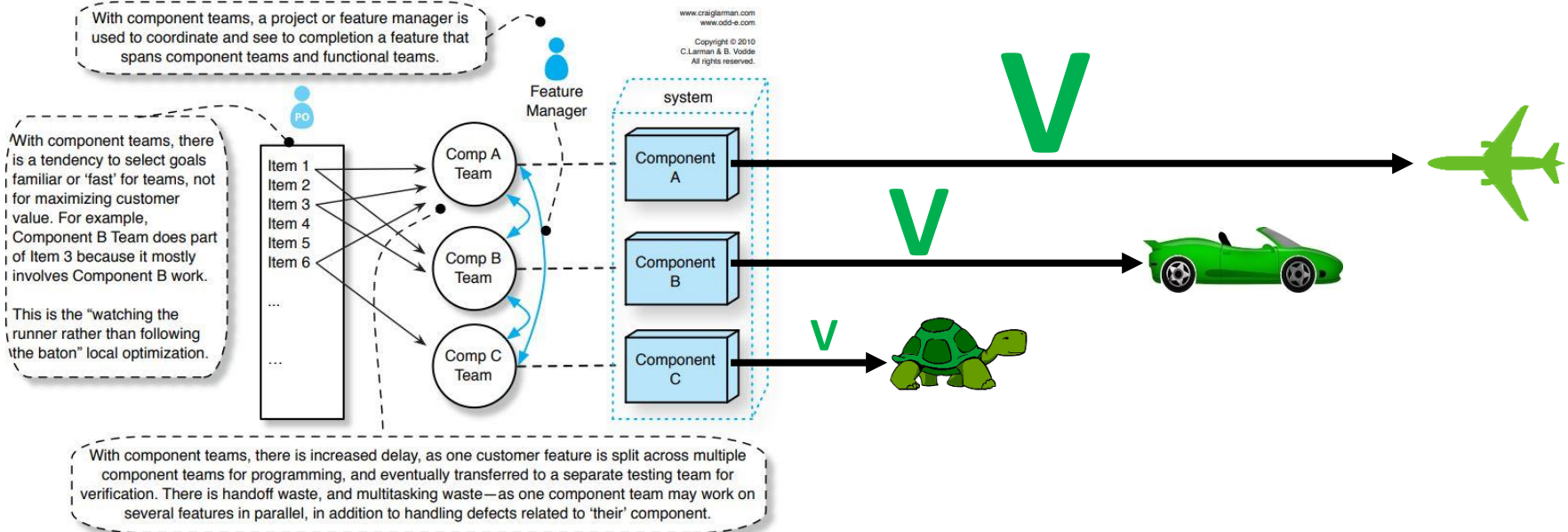
<http://www.keystepstosuccess.com/2020/05/05-05-less-talks-dave-snowden-answering-tough-questions-qa/>

Relation Between Estimation and Funding



Velocities Saga

Component Teams



Sourced from:

<https://less.works/resources/graphics/index.html>

Big Mistake # 1

$$V > V > v = \text{Thinking Face}$$

Big Mistake # 2

$$V + V + v = \text{Thinking Face}$$

Can these velocities be used to reliably estimate volume and complexity of delivered features?

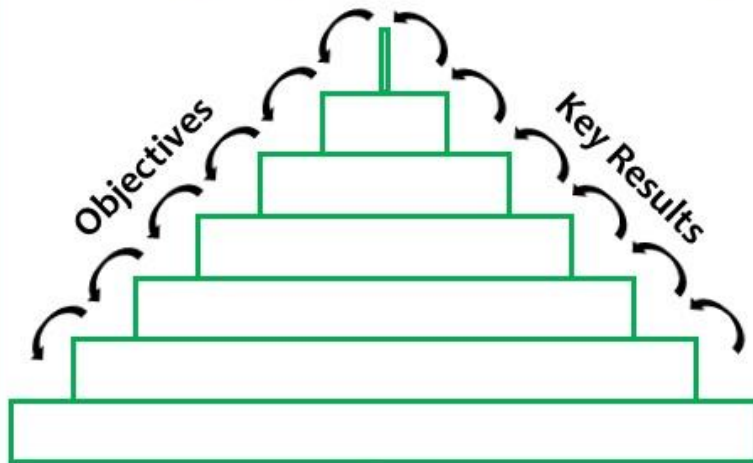
Organizational Layering And OKRs

OKRs: Narrowing The Gap Between “O(s)” and “KR(s)”

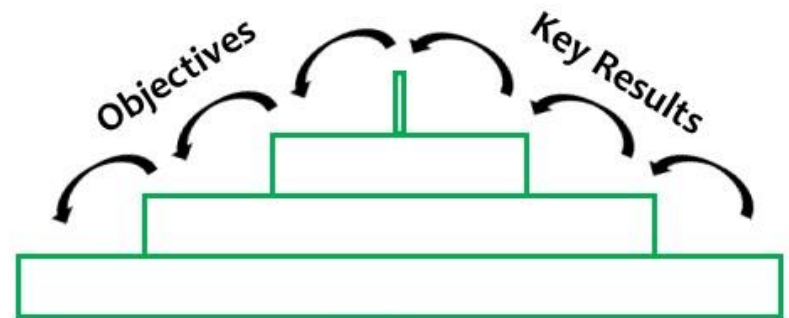
As strategic **objectives** ("O") cascade from-top-to-bottom, through an organizational structure, they get interpreted, at each level. This interpretation is subjected to cognitive bias, variance, errors & omissions... and even intentional numbers gaming. The same is true about **key results** ("KR") as they roll up, from-bottom-to-top.

More hierarchical organizations, with excessive reporting layers and complex WBS (e.g. projects, programs, portfolios) run a higher risk of OKRs, becoming inaccurate and unreliable measures.

Simplicity of organizational design will ensure that OKRs remain a useful framework.



VS



Avoid This:

Brought to you by www.keysteptosuccess.com



Try This:

Their Opinions Should Matter

- [Ken Schwaber: unS**e at any speed](#)
- [Ron Jeffries: Issues with S**e](#)
- [Martin Fowler: "S**e = sh**y Agile for Enterprises"](#)
- [Mike Beedle: Truth about the S**e](#)
- [Mike Cohn: L.A.F.A.B.L.E \(Large Agile Framework Appropriate for Big, Lumbering Enterprises\)](#)

The Big Question:

- **Is NOT:** Is our own scaling Right or Wrong?
- **It IS:** Are our scaling approaches consistent with the fundamental principles of Agile & Scrum?



Ken Schwaber



Agile Manifesto
Co-Author

Scrum & Nexus
Co-Creator

Ron Jeffries



Agile Manifesto
Co-Author

XP Co-Founder

Martin Fowler



Agile Manifesto
Co-Author

Experienced Software
Developer

Mike Beedle



Agile Manifesto
Co-Author

Co-Founder of
Enterprise Agility

Mike Cohn



Early contributor to Scrum

Co-Founder of Scrum Alliance
Experienced Software Developer

References

- [05/05 – LESS TALKS: Dave Snowden: Answering Tough Questions](#)
- [Ken Schwaber: unSAFE at any speed](#)
- [Mike Cohn: L.A.F.A.B.L.E \(Large Agile Framework Appropriate for Big, Lumbering Enterprises\), by Mike Cohn](#)
- [S. Fe is not Agile. S. Fe is not even Scrum, by Mike Beedle](#)
- [Issues with SAFE, by Ron Jeffries](#)
- [“SAFE = shitty Agile for Enterprises”, by Martin Fowler](#)
- [SAFE: Market Share Increase. Rapid Growth. What Is The Recipe?, by Gene Gendel](#)
- [Dan Snowden: SAFE: the infantilism of management](#)
- [Does SAFE agree with the Agile Manifesto?, by Peter Merel](#)
- [SAFE ≠ AGILE, by Tom Mellor](#)
- [Dependencies, Scrum of Scrums, and SAFE, by Ron Jeffries](#)
- [https://ronjeffries.com/categories/safe/, by Ron Jeffries](#)
- [GOTO 2015: Agile is Dead, by Dave Thomas](#)
- [Why do so many companies seem to jump straight to SAFE when starting Agile?, by Mark Levison](#)
- [U.S. Air Force Questions about Agile /SAFE Memo? -highly discouraging from using rigid, prescriptive frameworks such as SAFE | \(Slide 12 screenshot\)](#)
- [CSO Memo on Agile – and SAFE, by Nicolas M. Chaillan \(US Air Force Chief Software Officer\)](#)
- [“It’s just a toolbox” – essentials and accidents in scaling agile, by Dr. Agilefant](#)
- [Scaling Agility or Bureaucracy, by Ari Tikka and Ran Nyman](#)
- [LeSS SAFE comparison, by Ari Tikka and Ran Nyman](#)
- [The Horror Of The Scaled Agile Framework, by Neil Killick](#)
- [You Don’t Need a Complicated Story Hierarchy, by Mike Cohn](#)
- [Let’s Acknowledge SAFE for What It Is....And Move On, by Mike Cottmeyer](#)
- [Revenge of the PMO, by Marty Cagan](#)
- [Why SAFE Is Not The Scaled Agile Approach You Need, by Renee Thoughton](#)
- [Remove References To Scrum From SAFE!, by Den Sunny](#)
- [Beware SAFE \(the Scaled Agile Framework for Enterprise\), an Unholy Incarnation of Darkness, by Sean Dexter](#)
- [10 Common Mistakes when Implementing SAFE, by Michael Küsters](#)
- [Steal “Agile”: Despicable Mission, Jacques Morali, Victor Willis, Henri Belolo](#)
- [How is SAFE different from Scrum/Agile project management?, by Peter Stevens](#)
- [Failed #SquadGoals, by Jeremiah Lee](#)
- [Case Study : When emulating Scaling Agile at Spotify went awry at Refinery29, by Andy Park](#)
- [The Spotify “Model”: Don’t Simply Copy-Paste, by Evan Campbell](#)
- [Spotify didn’t implement the Spotify model by copying Spotify...by Kent Beck](#)
- [“Spotify Model” – 10 lessons in transplantology, by Kate Hobler \(Terlecka\)](#)
- [Scaling Agile @ Spotify, with Tribes, Squads, Chapters & Guilds, by Henrik Kniberg & Anders Ivarsson](#)
- [Don’t Copy the Spotify Model, by Ben Linders](#)
- [Over-engineered “Agile Operating Model” by Gene Gendel](#)
- [Accelerated career path by “!@#%^ Coach”, by Gene Gendel](#)

Please, Rethink Your Intentions

Tired of *Scaling*?



(...you are so not alone...)

**Maybe, it is time to
DE-Scale?**



ALTERNATIVES???

Please, UNDERSTAND...

Organizational STRUCTURE –

is the 1st Order Factor (Variable) that has impact on everything else inside ECOSYSTEM.

Everything else just follows: culture, mindset, behaviors, norms, values, processes, policies

e.g. metrics, velocities, OKRs, KPIs, RAGs, maturity levels – much less relevant

Please, UNDERSTAND...

It is vital to appreciate that organizational agility cannot be achieved by a development team in isolation -- it is a system challenge for organizational redesign. Especially when you are interested in LeSS within an R&D department of thousands, where each product group may have 200 or 700 people distributed in two or five sites around the world. If an engineering team has the technical capacity to adapt or change quickly, but requirements management, legal practices, product management, HR policies, site strategies, and deployment processes all emphasize rigidity, conformance to original plans, conformance to the status quo, and slow practices, then how can there be real agility?

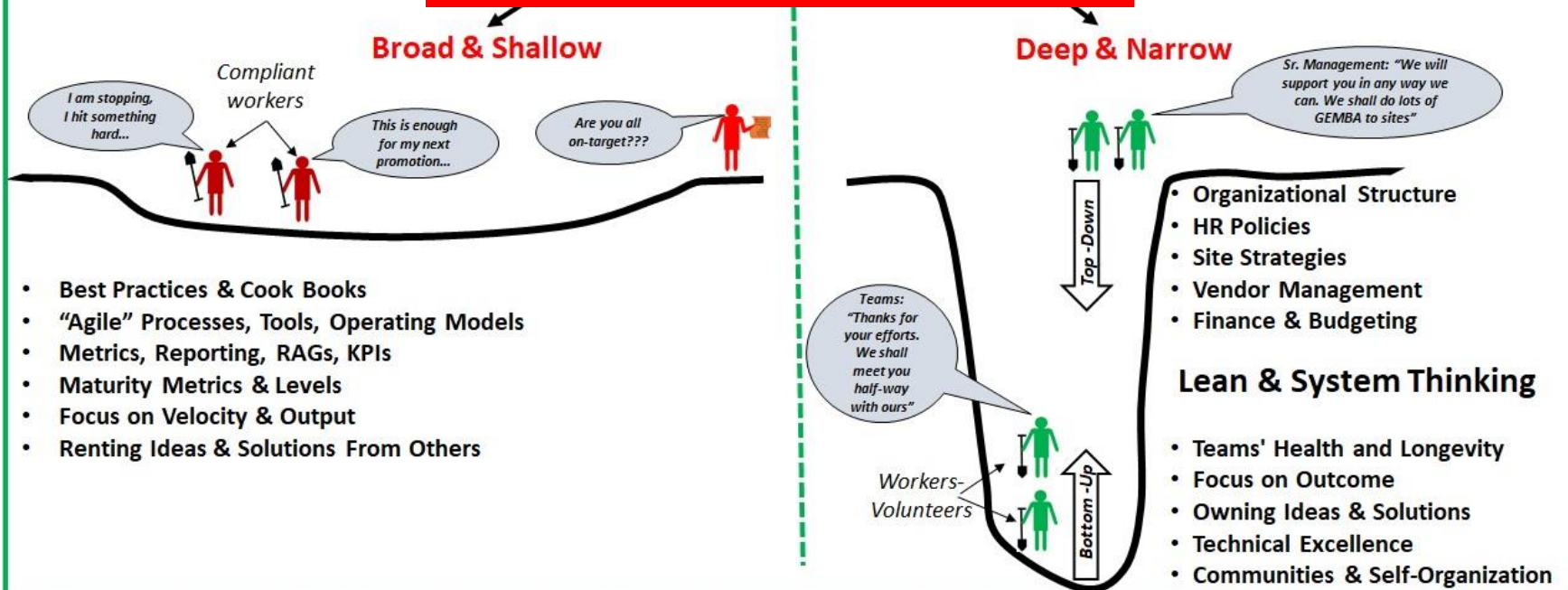
Source: <https://www.scrumalliance.org/community/spotlight/craig-larman/june-2015/less-agile-or-less-agile>



<http://www.keystepstosuccess.com/2017/08/agile-organization-as-a-sushi-roll/>

Scaling: Deep & Narrow vs. Broad & Shallow

Make Your Choice

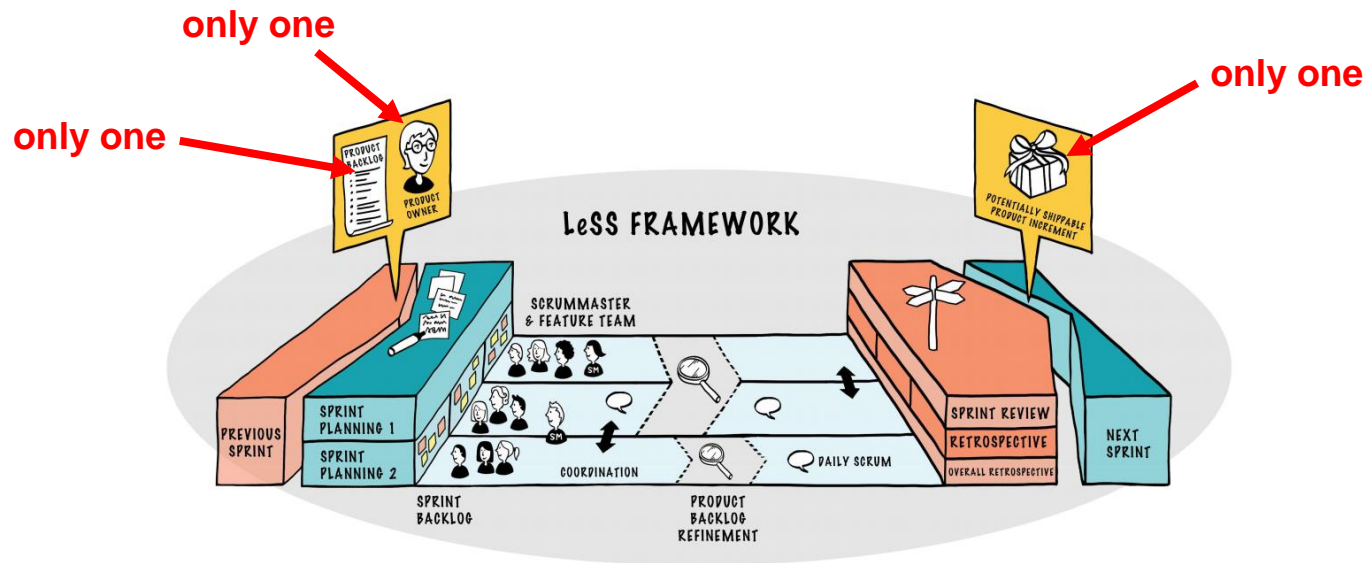


When it comes to LeSS adoptions (right side: deep & narrow), what companies are focused on – matters a lot. Although focus may change over time, in LeSS adoptions, efforts are top-down and bottom-up – and they are complimentary.

By Gene Gendel. 2020

Try: “Deep & Narrow”. Parallel Organization.

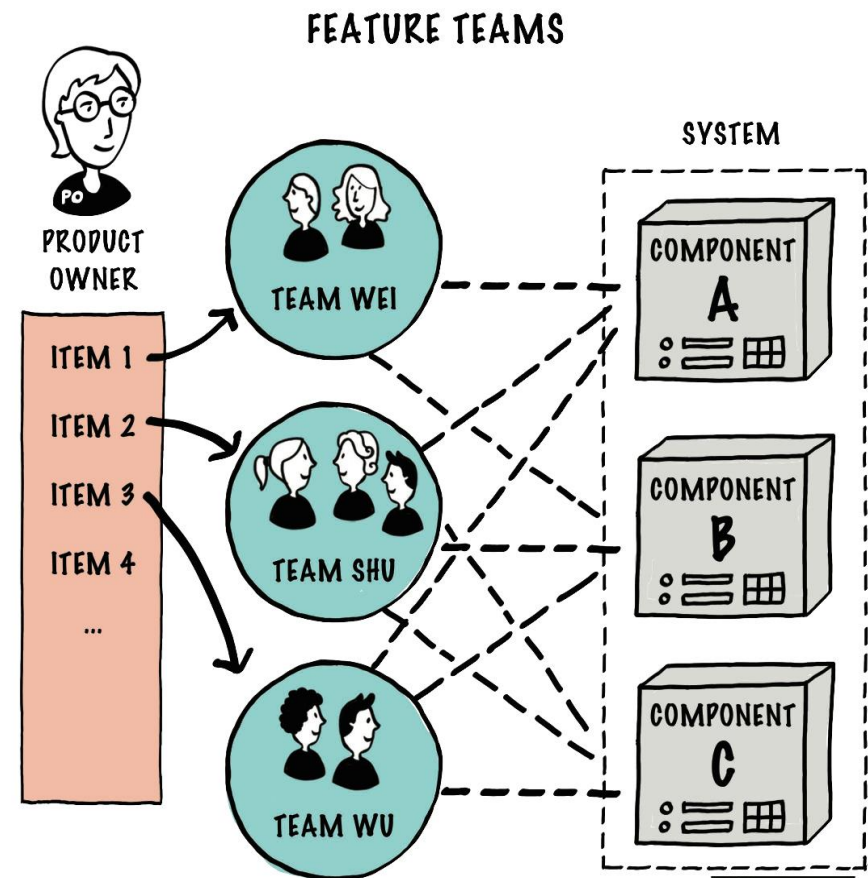
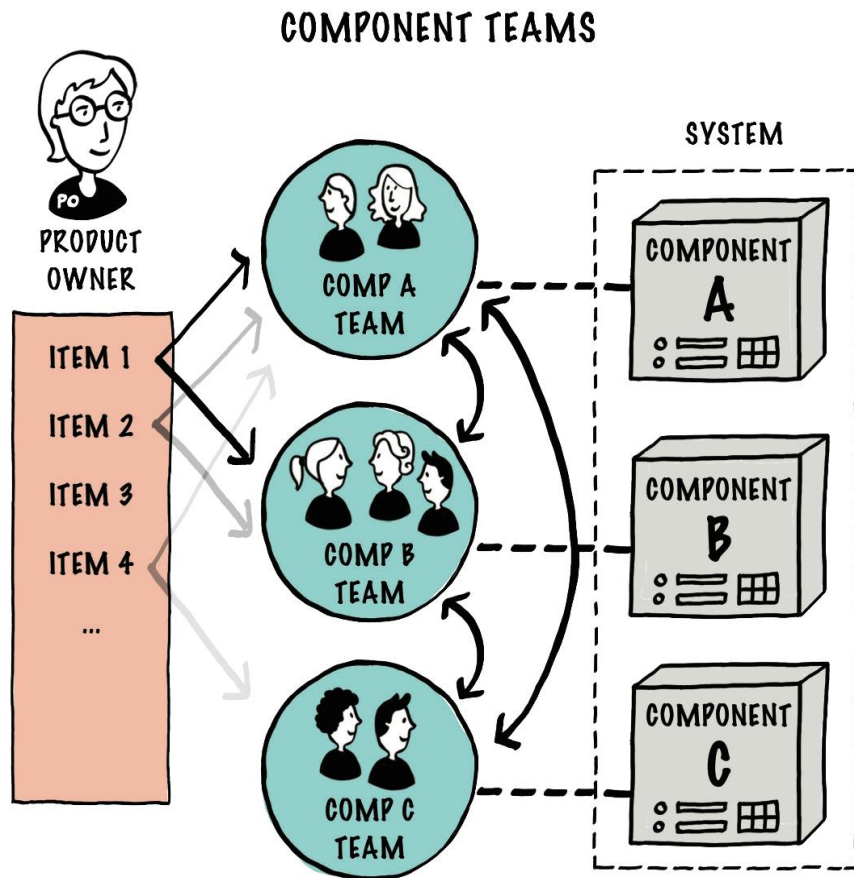
- No more than 50-60ppl: don't try to bite more than you can swallow
- 2-8 Teams
- **One Product. One Product Owner. One Backlog**
- Months of preparation and then a 'FLIP'.



<http://less.works> BY-ND

Scale beyond this ONLY, if it is ABSOLUTELY necessary and when you are READY.

From Component Teams to Feature Teams

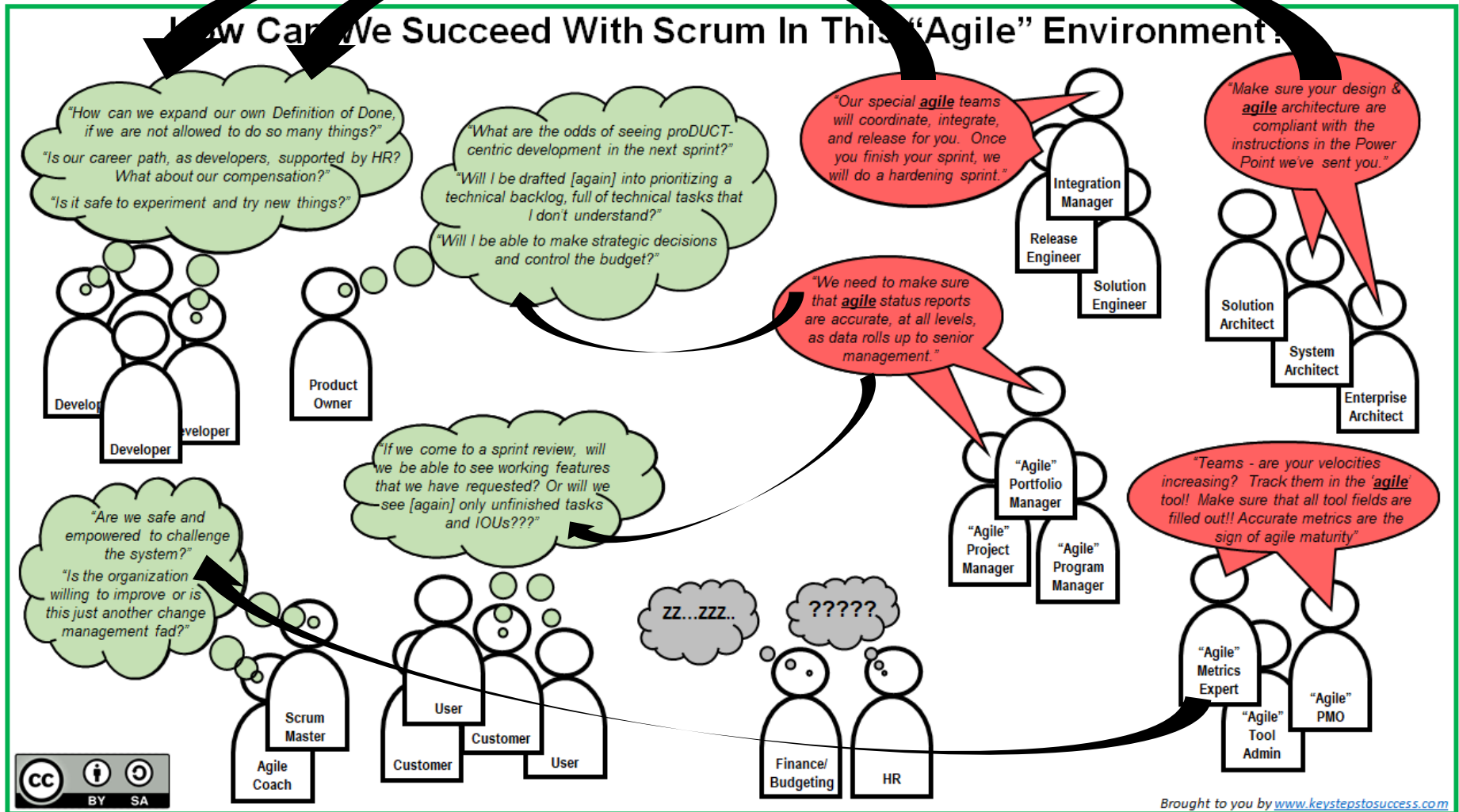


<http://less.works> CC BY-ND

Q & A

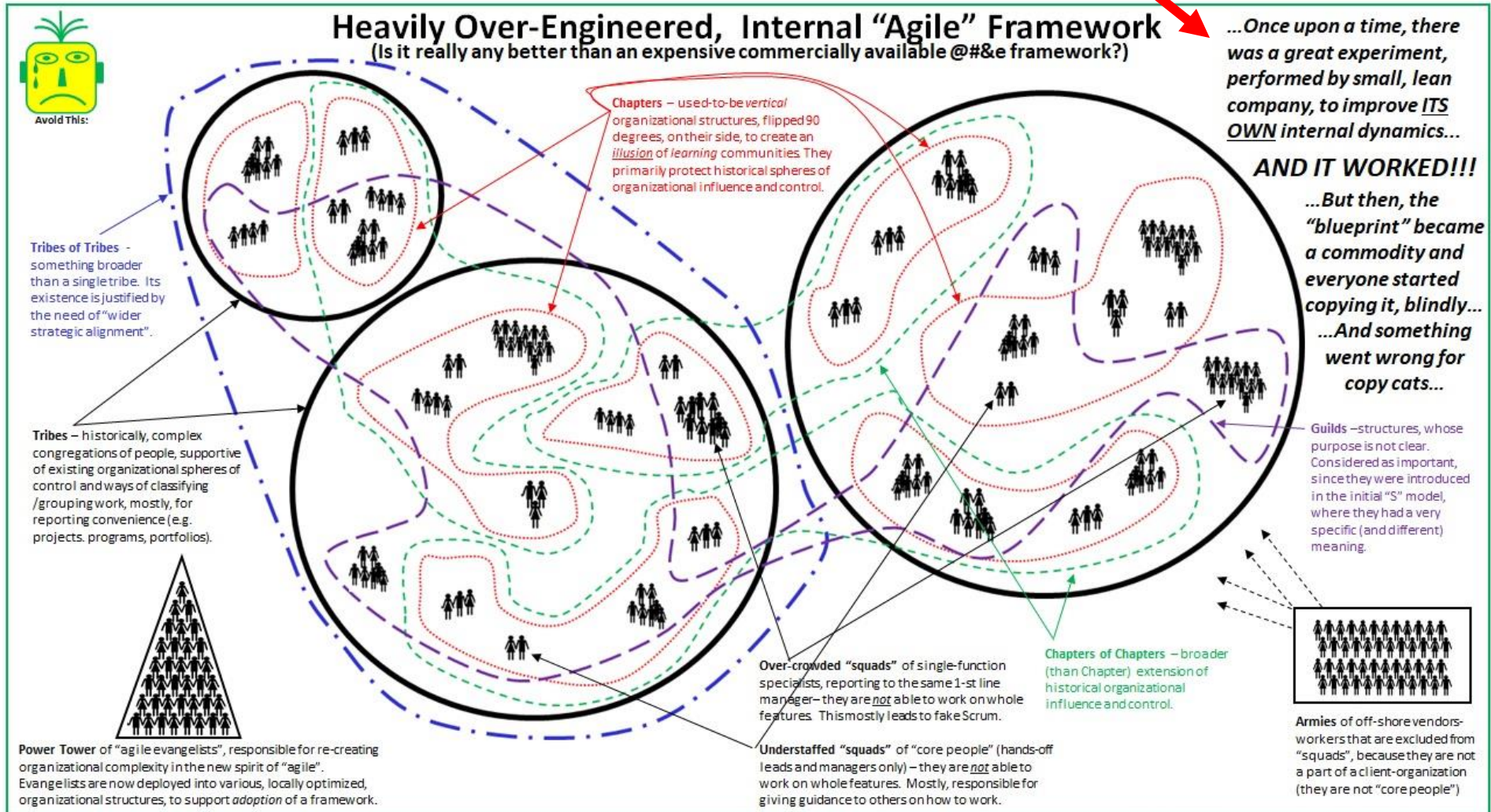
APPENDIX

Leaving “Scraps” for Scrum



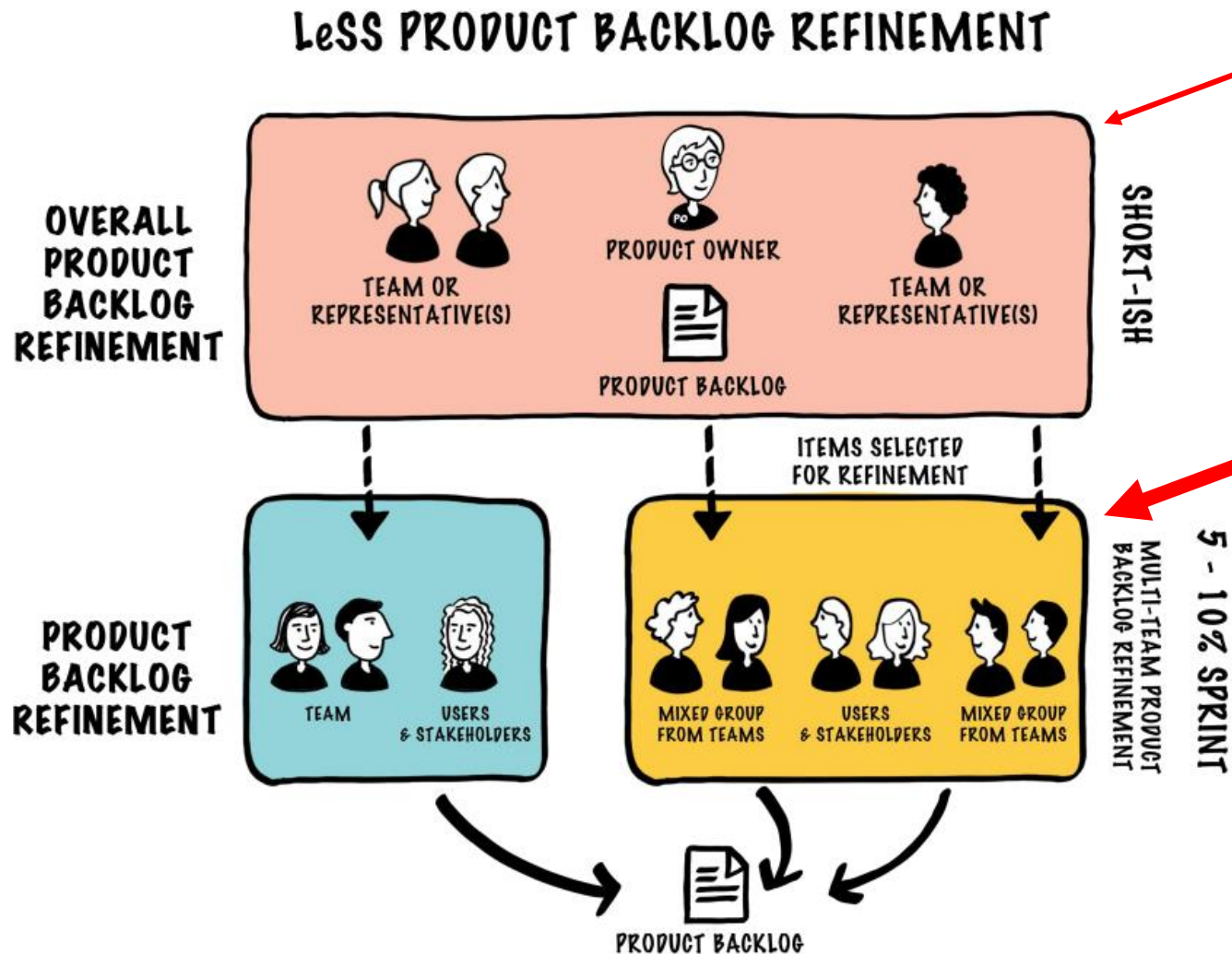
This problem grows exponentially with bad “scaling”

Blindly Copying Someone Else's Experiment



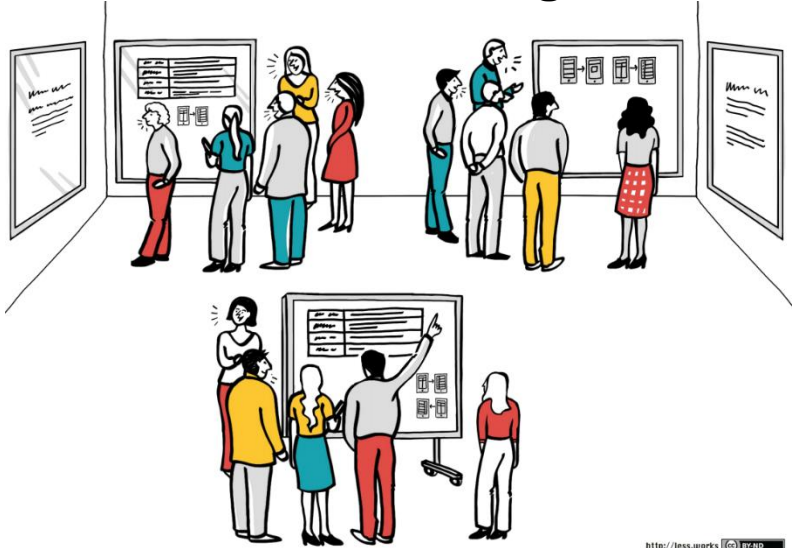
Brought to you by www.keystepstosuccess.com

Opportunities For Coordination In LeSS



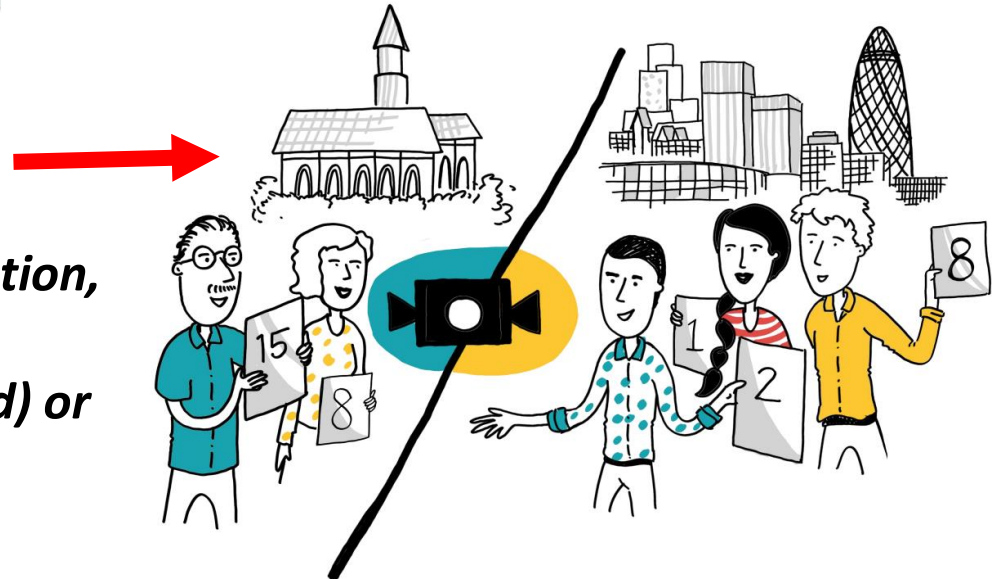
Opportunities For Coordination In LeSS

Multi-team Product Backlog Refinement



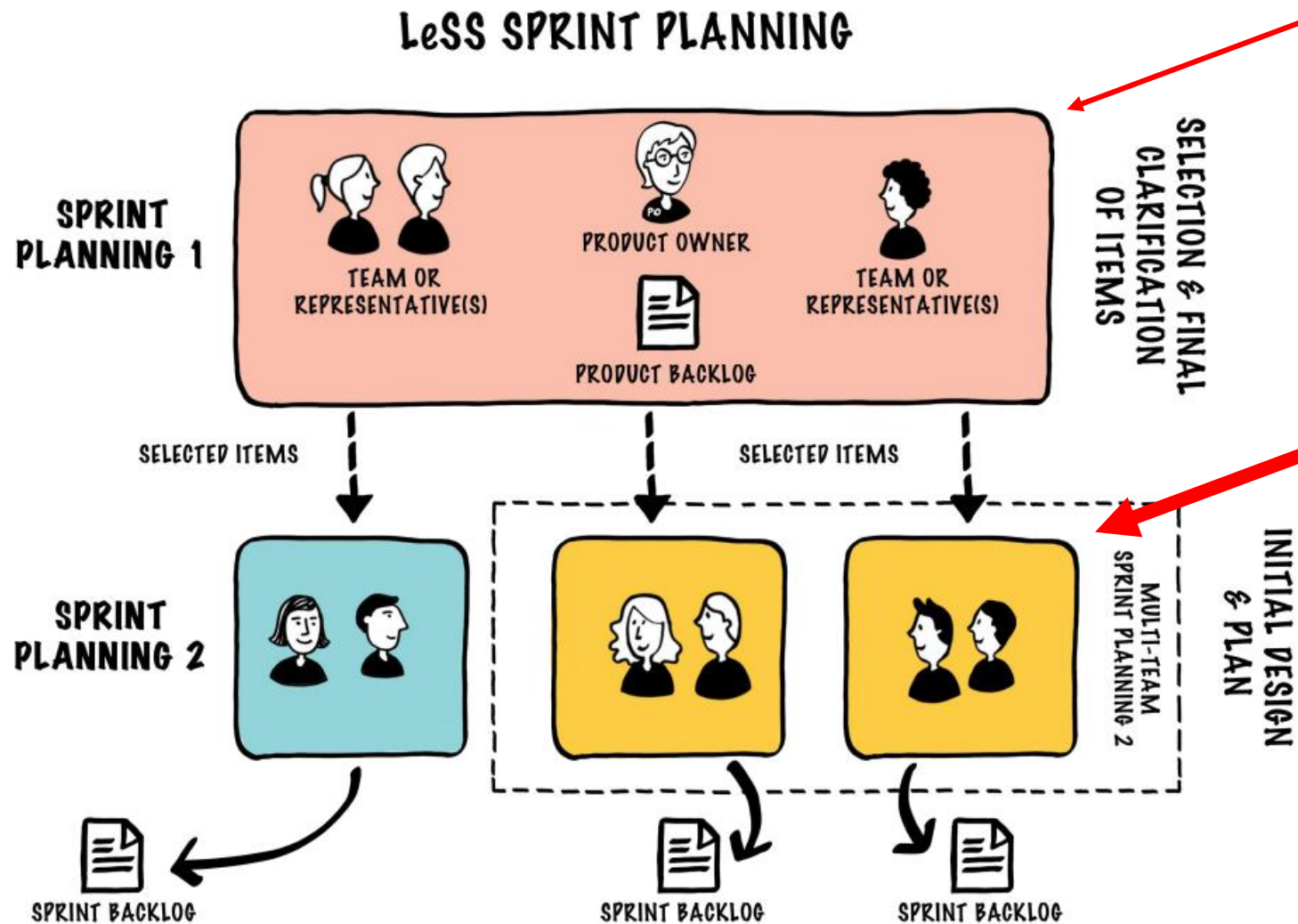
- *Open space (physical or virtual)*
- *“Promiscuous” collaboration*
- *Directly, with users and customers*

Multi-site estimation with Planning Poker

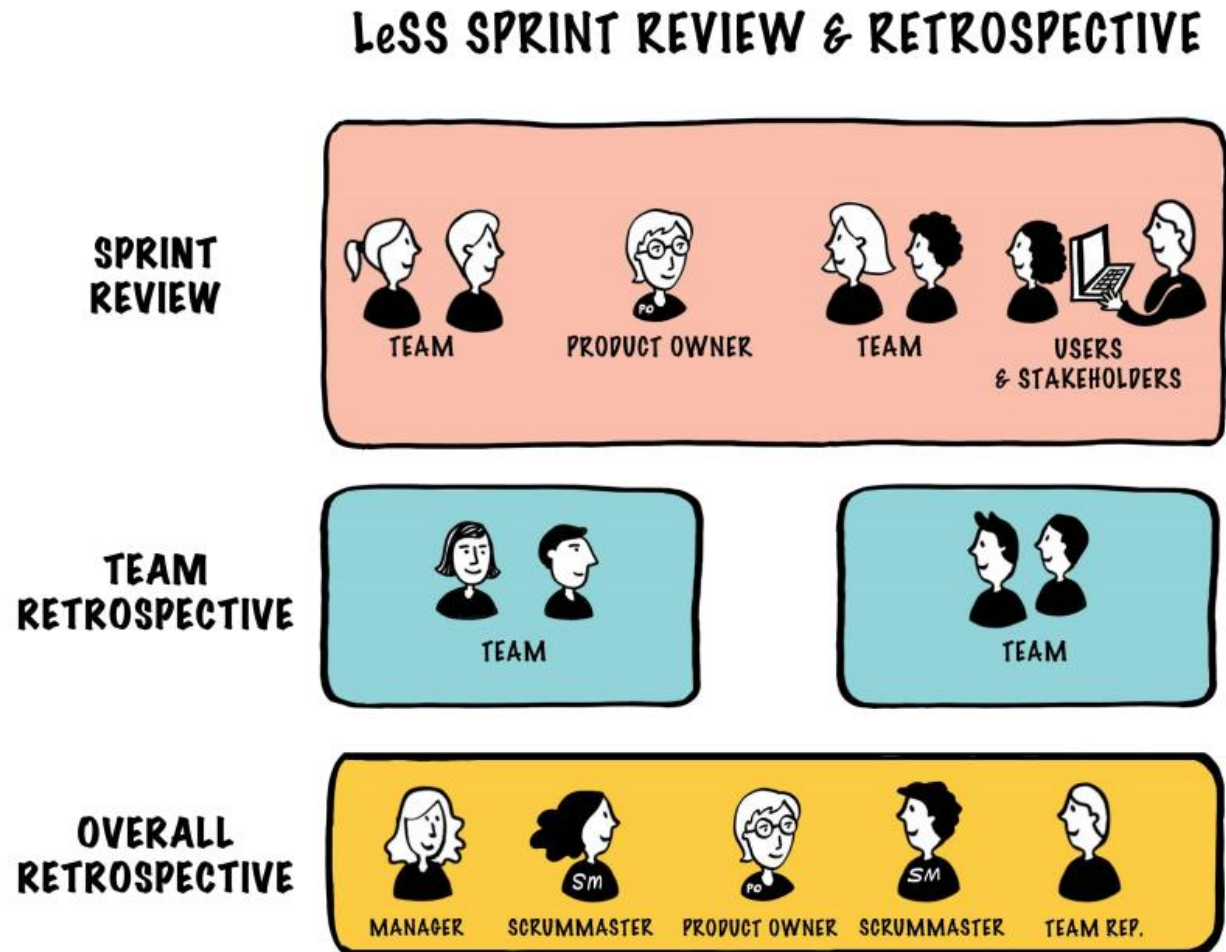


- *By “doers” (team members)*
- *Everyone is involved*
- *The main goal: CCC (Card, Conversation, Confirmation)*
- *No need to be too conservative (pad) or aggressive, while estimating*

Opportunities For Coordination In LeSS



Opportunities For Coordination In LeSS



<http://less.works> BY-ND

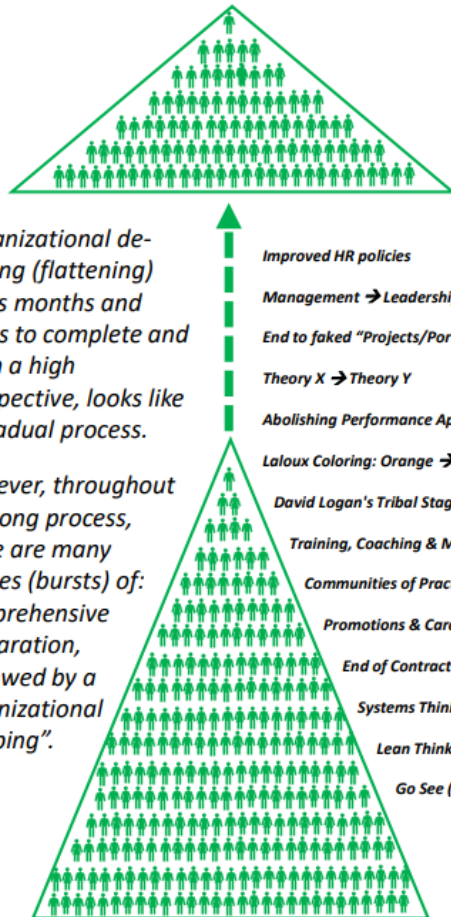
Less is More

Scaling Organizational Adaptiveness (a.k.a. “Agility”) with Large Scale Scrum (LeSS)

Organizational De-Scaling / Flattening

Organizational de-scaling (flattening) takes months and years to complete and from a high perspective, looks like a gradual process.

However, throughout this long process, there are many phases (bursts) of: comprehensive preparation, followed by a organizational “flipping”.



Improved HR policies
Management → Leadership
End to faked “Projects/Portfolios”
Theory X → Theory Y
Abolishing Performance Appraisals
Laloux Coloring: Orange → Green
David Logan's Tribal Stage: 3 → 4
Training, Coaching & Mentoring
Communities of Practice
Promotions & Career Paths
End of Contract Game
Systems Thinking
Lean Thinking
Go See (Gemba)

Diagram created by Gene Gendel

Source: http://www.keystepstosuccess.com/wp-content/uploads/2018/02/scrum_scaling_org_descaling-1.pdf

LeSS

Simplified (flatter) and **inverted** (senior leadership is in supportive function) **organizational design**. **System Optimization**. Reduction of: silos, hand-overs. Scrum is implemented by coordinated, feature-centric teams (2-8), building the same, widely defined Product/serving the same Product Owner. Scrum is the main building block of IT org. structure. Teams are collocated. Teams strive to ‘own’ their decisions, not ‘rent’ from others. Adoption of LeSS is “**deep & narrow**”, not “**broad and shallow**”; no Big Bang transformation attempts.

Multi-site development - used for multiple locations. Reliance of technical **Mentoring** and **Communities**. No subsystem code ownership. Reduction of ‘undone’ work. Focus on **Customer value**. Support of Senior Leadership. **Involvement of HR**, supporting the concept of **job security, not role security**.

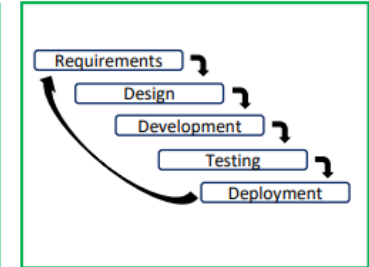
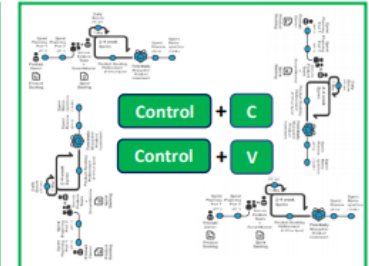
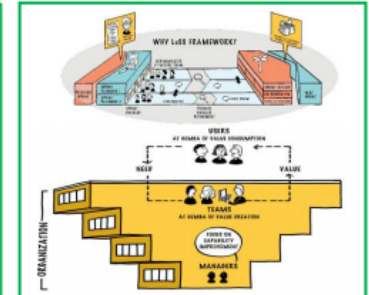
Transition from independent basic Scrum, performed by multiple teams to LeSS, is a desirable approach, when a product is widely defined and real Product Owner is identified

Scrum

Copy-paste scaling (no conscious scaling strategy) of Scrum throughout an organization: many teams doing their ‘own’ Scrum. Frequently manifested, when there is an internal competition between departments or company areas, on e.g. “*who is going to have more scrum teams by e.g. end-of-year*”. True product definition is weak. Cases of using **Scrum for component-centric development** are frequent (often, the result of trying to meet goals of agile transformation (% annually), set at enterprise level. This results in massive integration problems that need to be resolved by adding “*integration/stabilization*” sprints. Importance of **Scrum dynamics and roles** is viewed as **secondary**, to existing organizational structures and blueprints. Too many single-specialty experts and very few T-shaped workers. **No meaningful HR changes**.

Waterfall

Complex organizational design. Domains of single-function expertise, ownership and control. Numerous silos, hand-offs and translational layers between component teams, single-function specialty groups, and technology and business, overall. **Internal contracts and blame-shifting** strongly prevail. Long cycle “from concept to cash”. **Local optimization** by single-specialty workers/departments. **Theory X management** – is the primary way to manage/treat individuals. **HR** – putting a lot of emphasis on performance appraisals and individual ownership; manipulating workers with **subjective bonuses**; fostering an environment of **internal competition** and “*I am the best and you are not*” attitude. Weak definition of product value, from a stand-point of a paying customer. Manifestation of **Larman's Laws of Organizational Behavior**.



Graphics are courtesy of <https://less.works>

Scaling Scrum

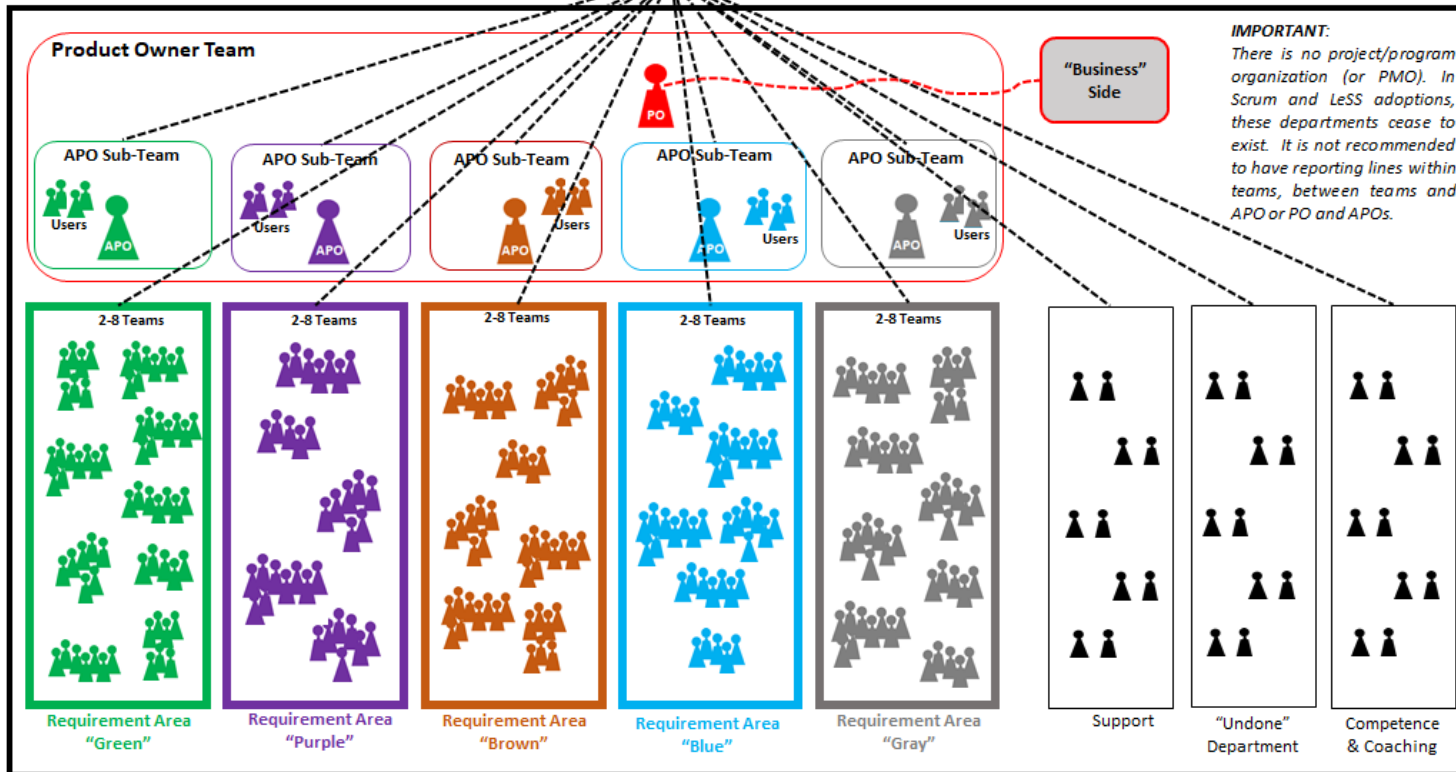
Waterfall

Relationships in LeSS

Guide: TRY.... Keeping LeSS Huge Organizational Reporting Structure As Flat and Simple As Possible

Product Owner Team— Product Owner plus all Area Product Owners. Conceptually, the same as in a LeSS structure. In huge LeSS Huge, there are also APO Sub-Teams that are, essentially, what is Product Owner team is in LeSS. Each APO Sub-Team consists of Area Product Owner and customers/users that can help APO with clarification and other daily work (**NOT** departments of analysts, specification writers, UI/UX designers, or architects).

Head of Product Group - most likely, a traditional “head of product group”. Whereas the whole group supports teams by Go See and help them remove obstacles and improve, Head of Product Group plays the *role of the line manager* of all the teams, whatever that is called in your organization. Product Owner Team members (including APO Sub-Teams) and all developers from teams in requirement areas, as well as people in other supportive organizational structures (e.g. support, ‘undone’ department, competence & coaching) -all report to Head of Product Group. **Note:** Since LeSS organizations don’t have matrix structures, there are no dotted-line managers.



IMPORTANT:

There is no project/program organization (or PMO). In Scrum and LeSS adoptions, these departments cease to exist. It is not recommended to have reporting lines within teams, between teams and APO or PO and APOs.

Undone department - e.g. test, QA, architecture, or business analysts - they should not exist in the smaller LeSS framework groups; rather, they should be integrated into the teams. Undone department is still frequently seen, in some LeSS adoptions, as they often cross organizational boundaries. A goal in every LeSS adoption should be to remove an undone department.

Support - In a LeSS Huge organization, it typically centralizes some support because of the massive volume of work. Still, this department should be as small as possible with the attitude “How can we help?” rather than “Take it this way!” This department is less likely to exist in smaller LeSS organizations, where teams support each other without the need for a separate, centralized group.

Competence and Coaching (technical, organizational) - group consists of skillful practitioner-experts who actively Go See and observe how people work. They pair up and work with people to discover their training and coaching needs.

By Gene Gendel. 2020