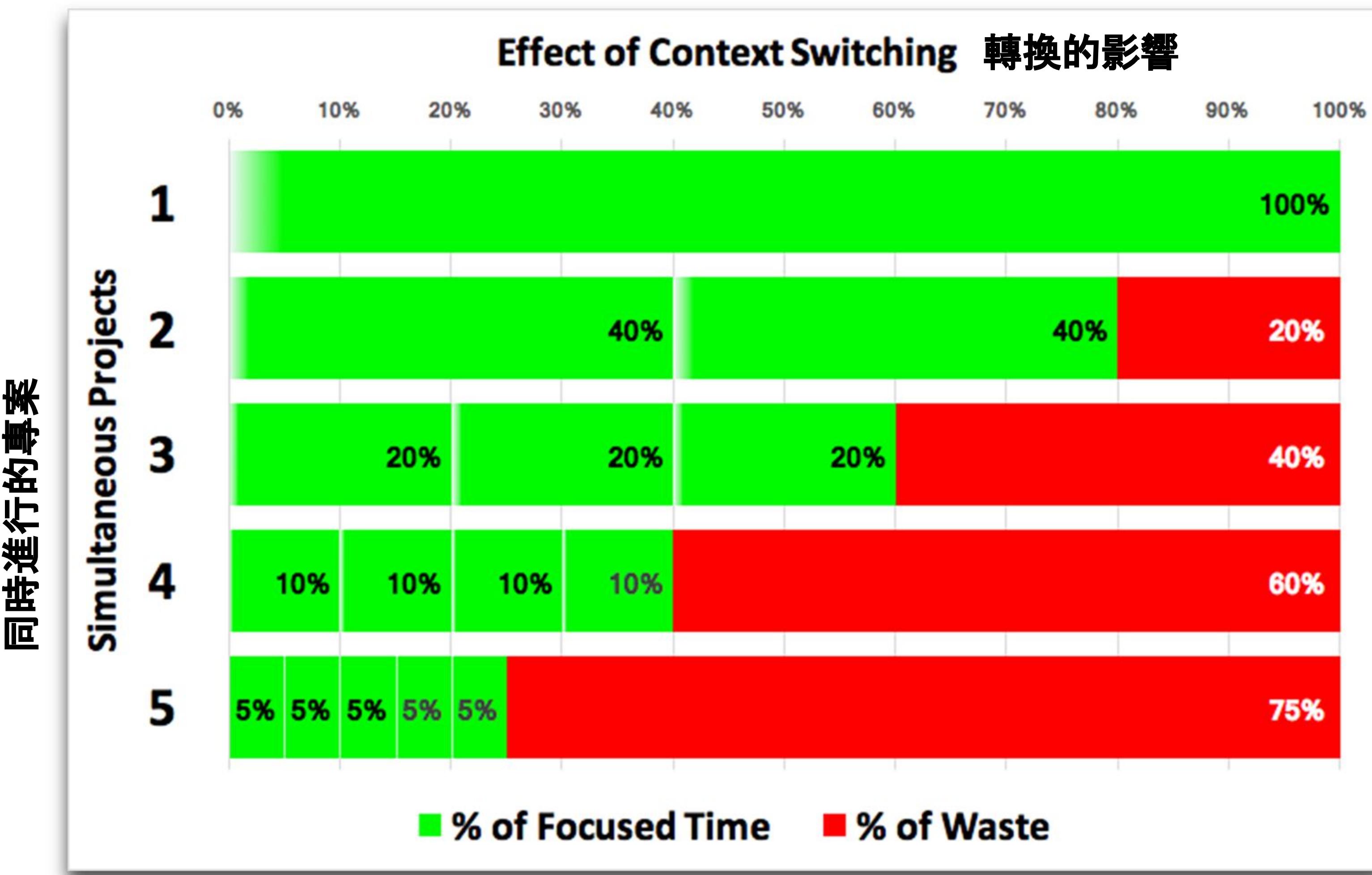


Weinberg's Table of Context Switching Waste

轉換浪費



Weinberg, Gerald M. (1992) Quality Software Management: Systems Thinking. Dorset House, p. 284

Scrum Pattern: Ready Backlog 準備好的待辦事項

Don't start without it 還沒準備好就千萬別開始做

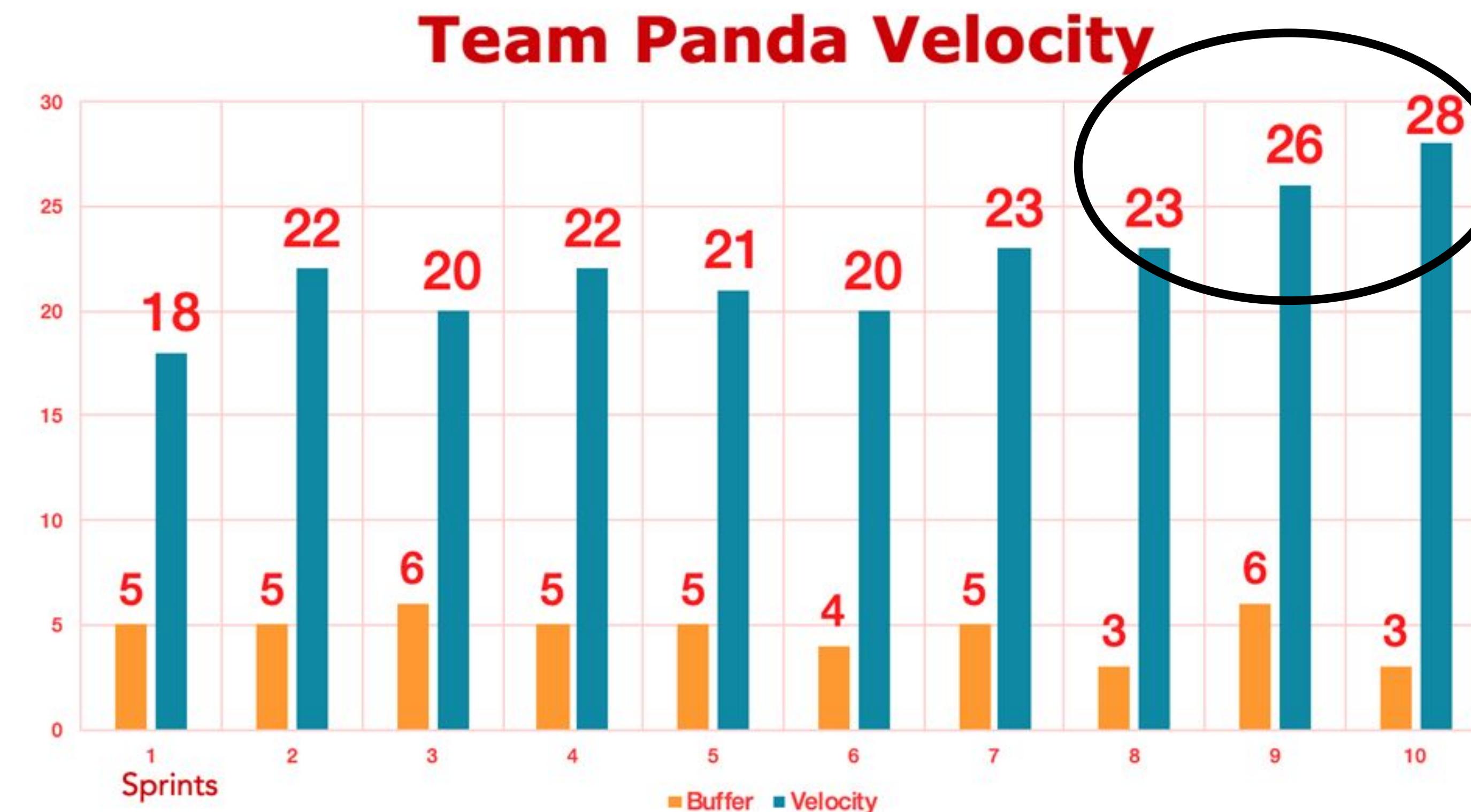
Scrum Pattern: Ready Backlog - Don't start
without it

Scrum Pattern: Yesterday's Weather 昨天的天氣

Use the last Sprint to predict the next Sprint

Scrum Pattern: Yesterday's Weather

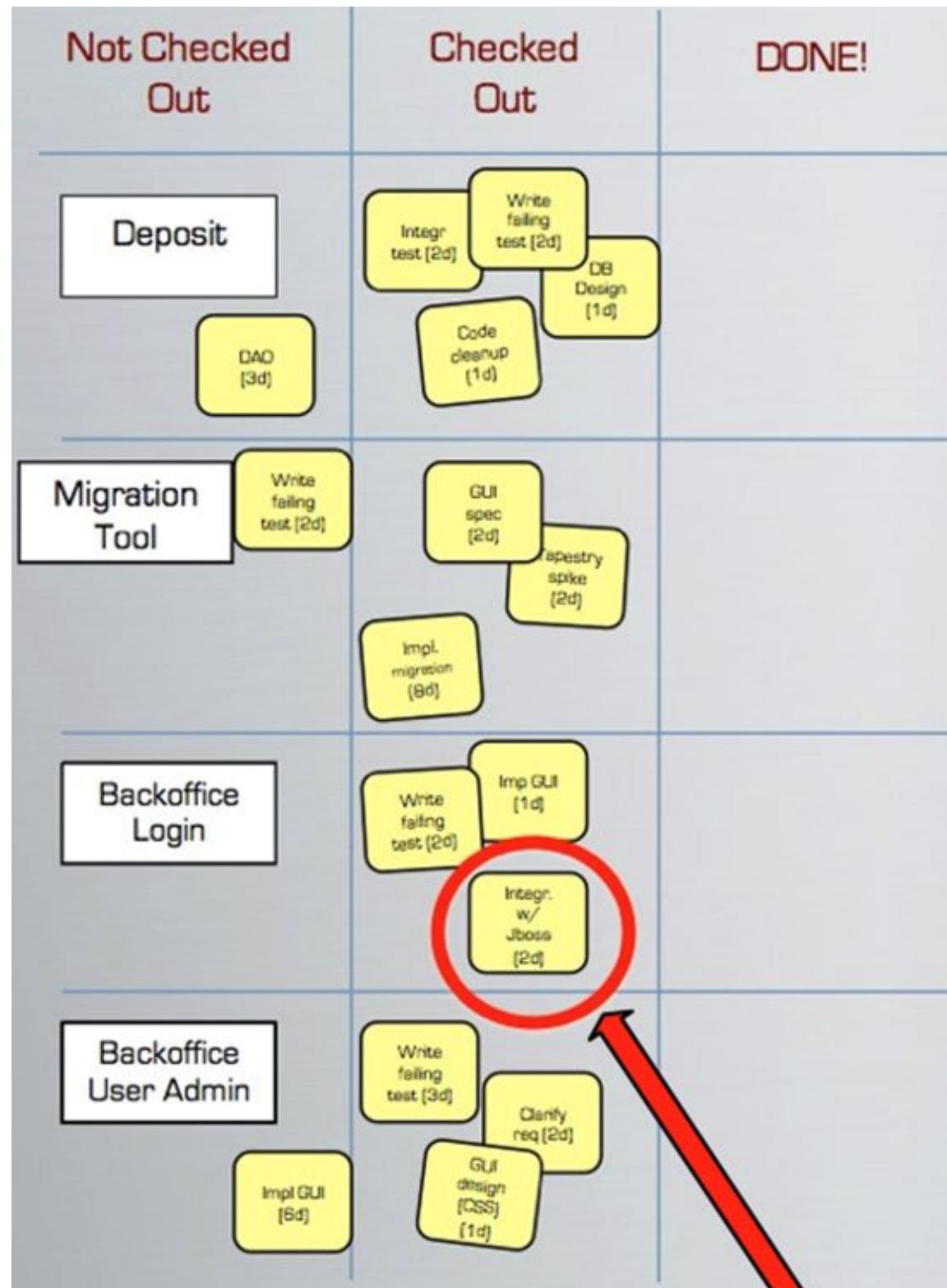
Use the last Sprint to predict the next Sprint



Scrum Pattern: Swarming 蜂擁行動

Individual Contribution vs Group Contribution

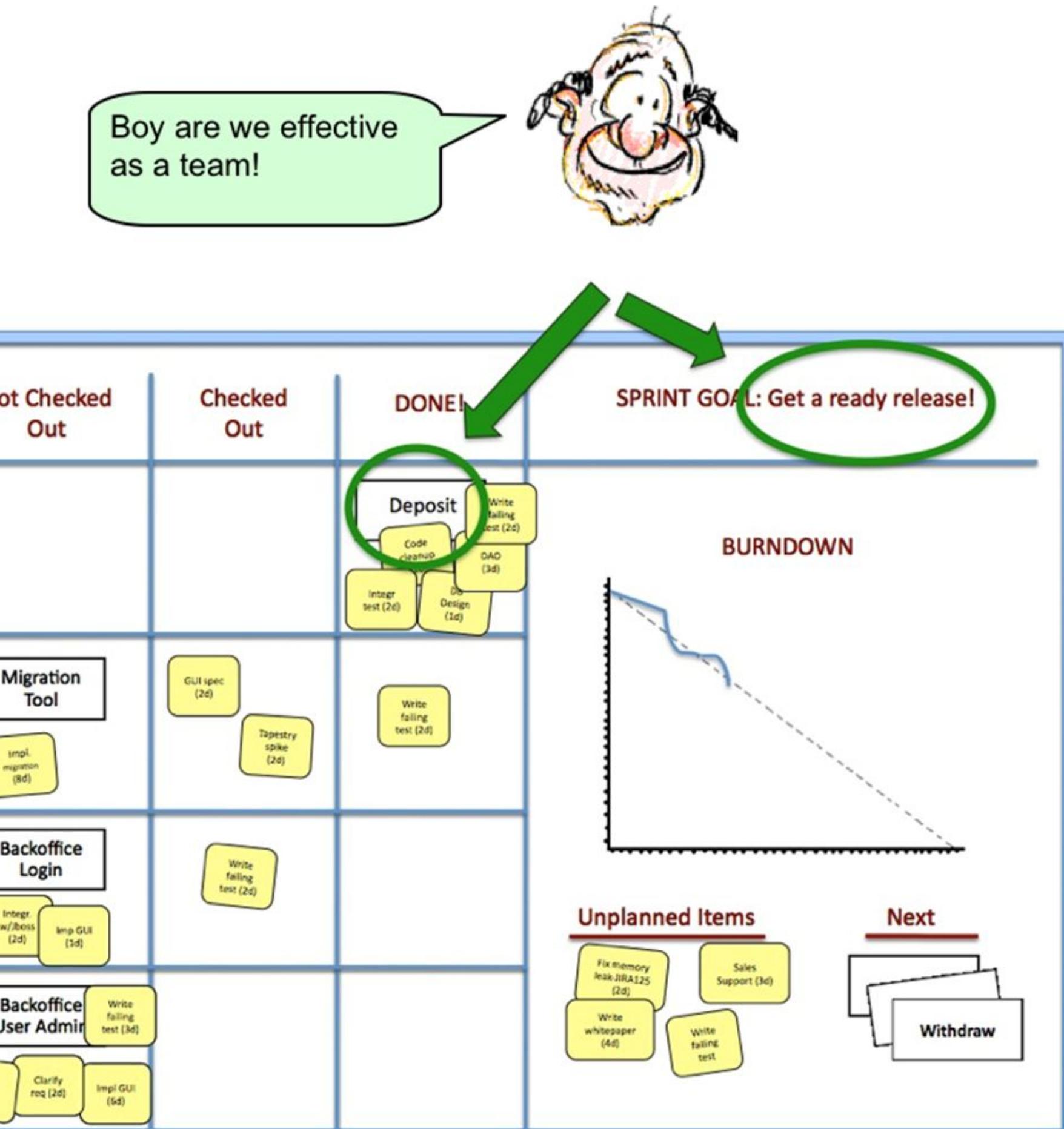
個人貢獻 vs 小組貢獻



I'm more efficient if I just do my tasks

Adapted from Henrik Kniberg

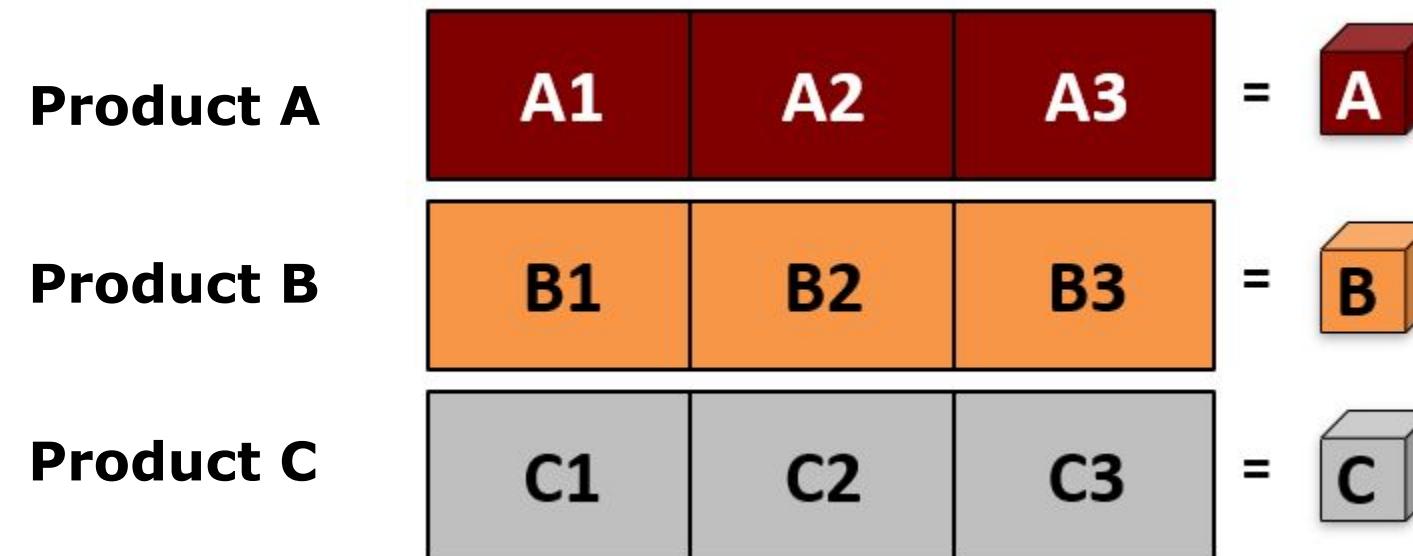
494



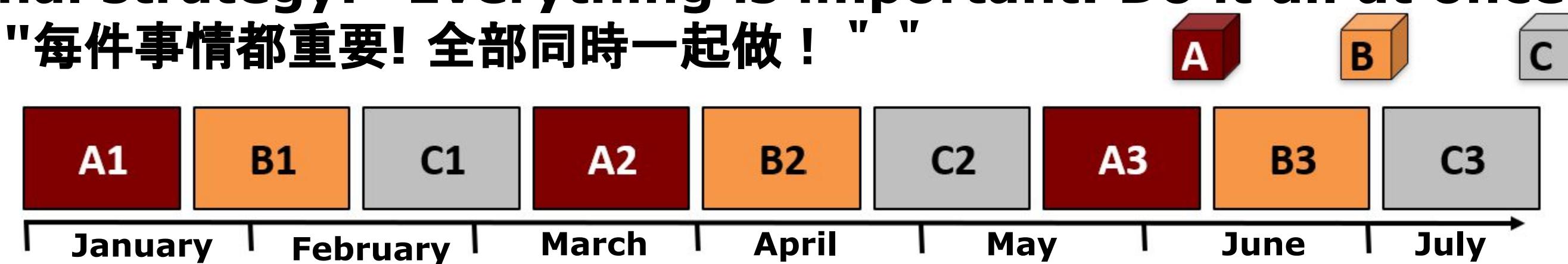
scruminc.

Scrum Pattern: Swarming 蜂擁行動

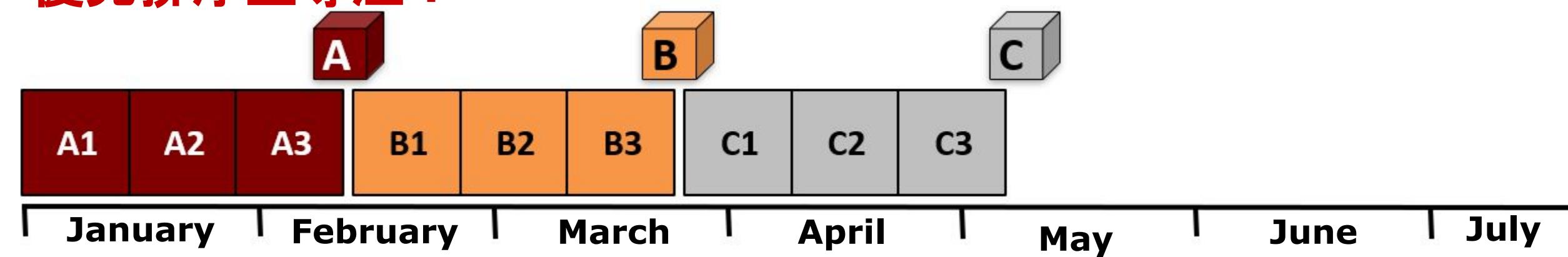
Prioritizing Between Projects 在專案之間排序



Traditional strategy: "Everything is important! Do it all at once!"
傳統策略："每件事情都重要！全部同時一起做！"



Agile strategy: "Prioritize & focus!"
敏捷策略："優先排序並專注！"



Adapted from Henrik Kniberg

Focus + Teamwork = Swarming

Warning: this could be loud



Warning: this could be loud

Scrum Pattern: Interrupt Buffer

插件緩衝

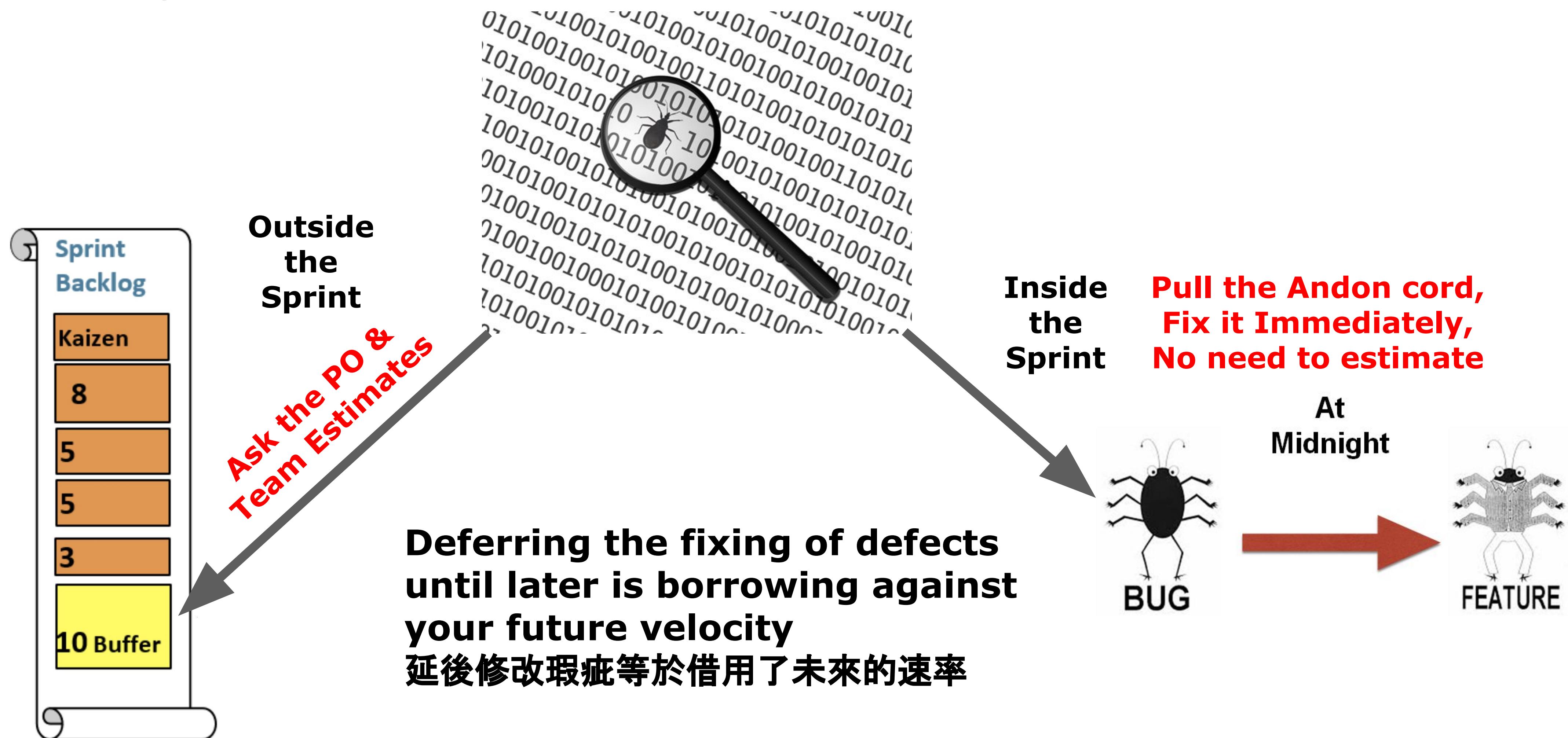
[Scrum Pattern: The Interrupt Buffer](#)

[Dealing with the Unexpected - Implement a Buffer](#)

Scrum Pattern: Good Housekeeping

良好的即時清理管理

How to get defect-free product? 如何打造無瑕疵的產品?



Scrum Pattern: Happiness Metric 幸福指標

How do you get teams to have fun?

如何讓團隊有樂趣？

Scrum Pattern: Happiness Metric

Happiness causes better performance

Scrum Pattern: Quantum Entanglement

量子糾纏

A Rapidly Advancing Field

- Interesting Engineering 20 Jun 2020 - researchers bypassed the classic method of data transfer where you send an information-bearing photon between two chips. Instead, they [teleported](#) information from one silicon chip to another utilizing a quantum-mechanically entangled photon pair
- Photons connected in an entangled quantum state know each other's characteristics at any time. **A change in the state of one immediately results in a similar change in the other**
一方的狀態改變時會立即導致另一方產生類似的改變
- Quantum internet allows instantaneous connection at any distance
- In the time of COVID-19 this presents an interesting opportunity (silver lining)



Physicists take first-ever photo of quantum entanglement.
(Image: © University of Glasgow/CC by 4.0) Life Science 18 Jul 2019

Quantum Entanglement 量子糾纏

Distributed Teams are unable to work at the same level of productivity as a collocated team. They produce lower quality work and have reduced morale. Even a team that starts off well will see degradation over time.

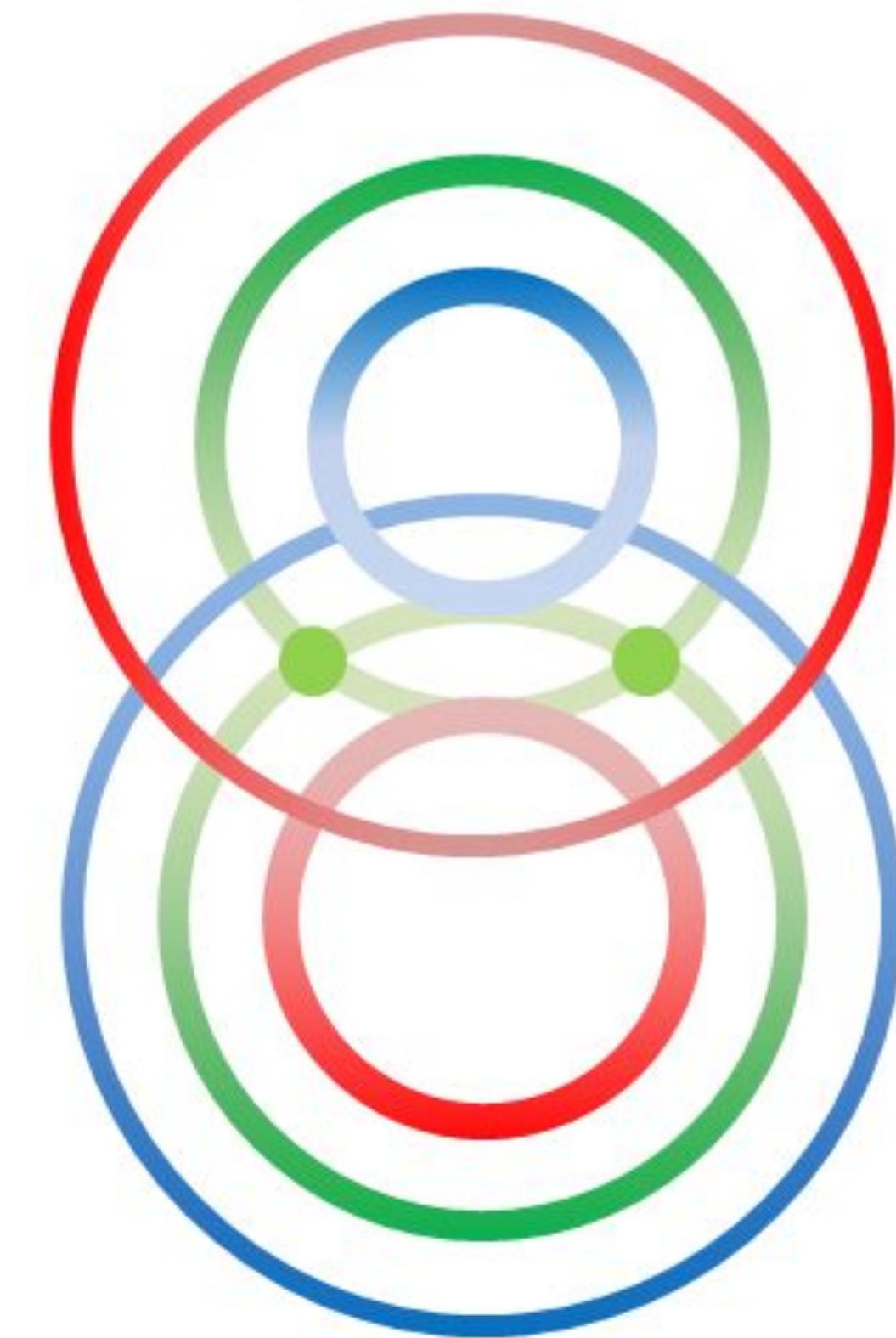
分散式團隊無法像同地點工作的團隊一樣做到在同一個產能水平工作，他們工作產出的品質較低，並且工作室氣較低，即使團隊有好的開始，也會隨著時間變差

Therefore: Establish social, cultural and technical connections between the locations of the team at the time the team is formed, establish measures of normal accomplishment, and monitor these measurements. If degradation is detected, re-establish the connections.

因此：在建立遠端分佈的團隊時，應該建立團隊之間的社交、文化和技術的關係連結，建立一般績效的衡量標準並進行監測，如果發現關係連結出現下降的現象，就重新建立連結

Three-step process:

- **Initialization (establish a bond)**: 準備好開始(建立關係) requires getting the team to work together locally (Face to Face Before Working Remotely) 先一起面對面工作
- **Inspect (monitor for consistent)** 檢視(監測以求一致): involves collection of key metrics on team performance and watching for any degradation 收集有關團隊績效的關鍵指數並注意下降出現
- **Adapt (re-bond if needed)** 適應(必要時重新連結關係): is repairing the team to reestablish shared culture with local performance. This is difficult and often needs another face-to-face experience 修復團隊以重建



Biphoton symbolic representation by Neolexx [wikimedia](#)

Scrum Pattern: Emergency Procedure

緊急程序

- How do you deal with surprises?

Scrum Pattern: Emergency Procedure
How do you deal with surprises?

Scrum Pattern: Scrumming the Scrum

用Scrum來改善Scrum

How do you ensure you continuously improve?

[Scrum Pattern: Scrumming the Scrum](#)
[Use Scrum to Kaizen Scrum](#)

Build a list/steps for starting up your Scrum Team

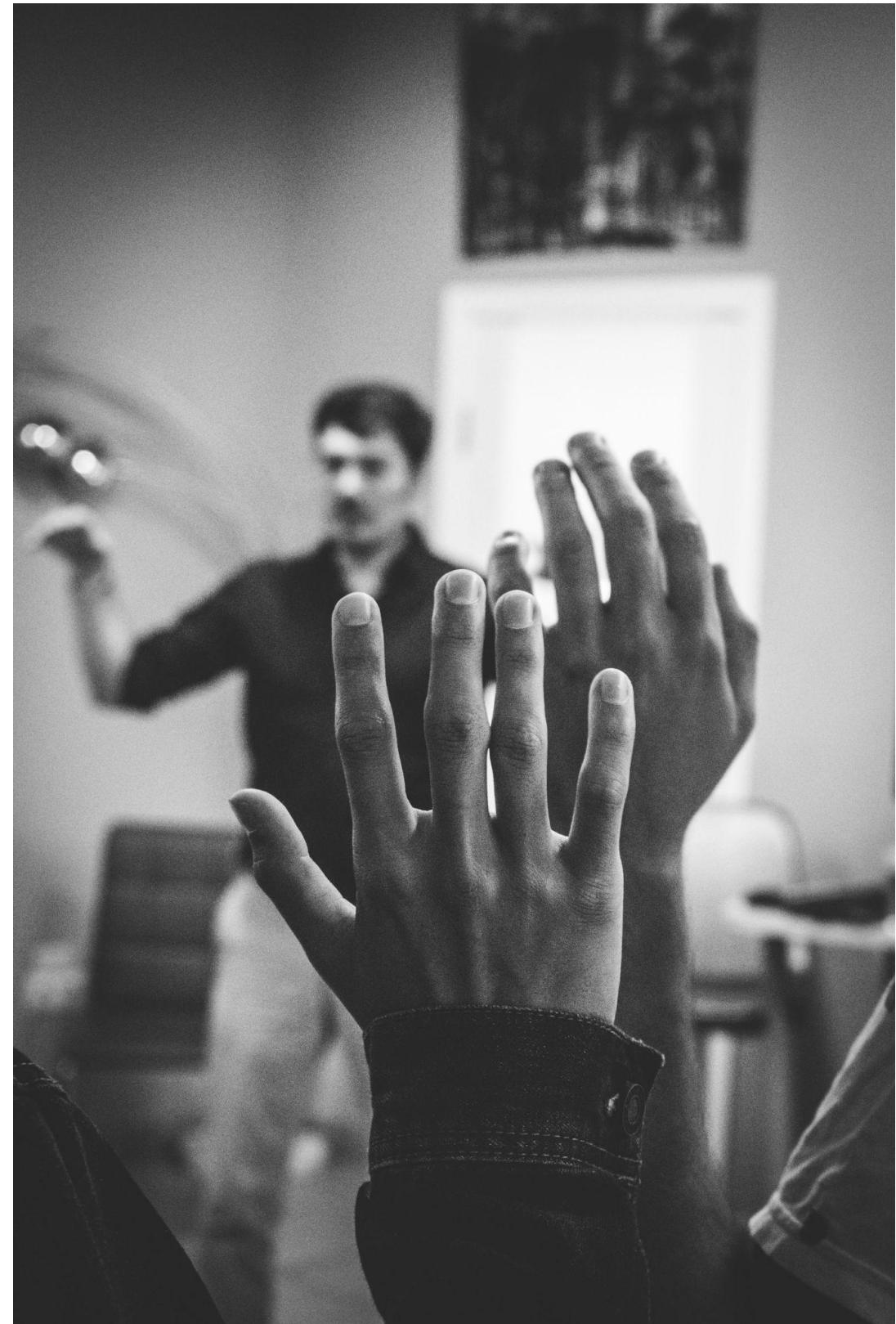
課後作業：列出創建Scrum團隊的清單

- You have a team
- You need to build a Product or Service for something
- Can you come up the list/steps to get started?

你有一個團隊，你需要做出一個產品或一種服務，你可以列出清單/步驟來開始做嗎？



Questions?



The person asks the question is the person who learns the most

Lean Tools and Principles

精實管理工具和法則

An Effective Scrum Master Has a Lean Team
Lean means remove waste. The key metric for waste
removal is Process Efficiency.

強有力的Scrum Master擁有精實團隊
精實管理表示消除浪費，其關鍵指標是流程效率
By definition, Lean means **Process Efficiency > 25%**
精實管理表示 **流程效率 > 25%**

Lean Means Not Building Useless Stuff

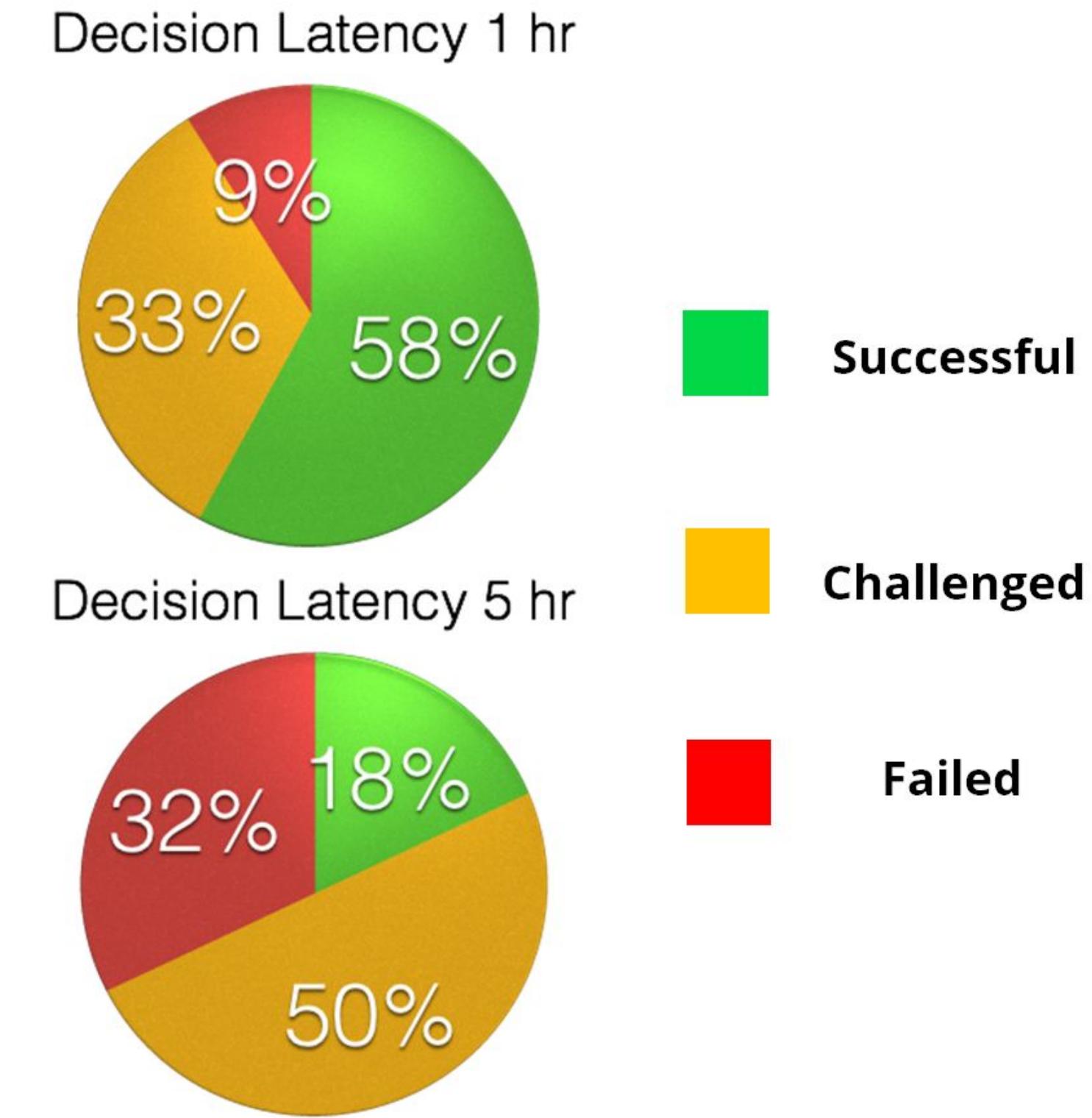
精實管理表示不要打造無用的東西

Customer
Will Use

Junk
Stories
Not
Needed

Dark
Work

- **25% of staff delivering stories **customers will use****
25%的員工交付消費者要的功能
-> **Scrum Master optimizes process efficiency**
Scrum Master優化流程效率
- **45% of staff delivering stories the customer **will never or rarely use****
高達45%的員工交付消費者永遠不會或非常少會用到的功能
-> **Product Owner Team eliminates junk stories**
Product Owner排除垃圾故事(工作)
- **30% of staff working on **zero value stories****
高達30%的員工在做零價值的工作
 - Lack of direction causes staff to make up work
缺乏方向會導致員工裝忙
 - Unwillingness to prioritize increase corporate waste
不願意做優先排序會增加企業浪費
 - Executive Metascrum eliminates useless projects
EMS排除無用的專案



Standish Group 2013-2017

Source: Dr J Sutherland, Scrum.inc based on research by the Standish Group

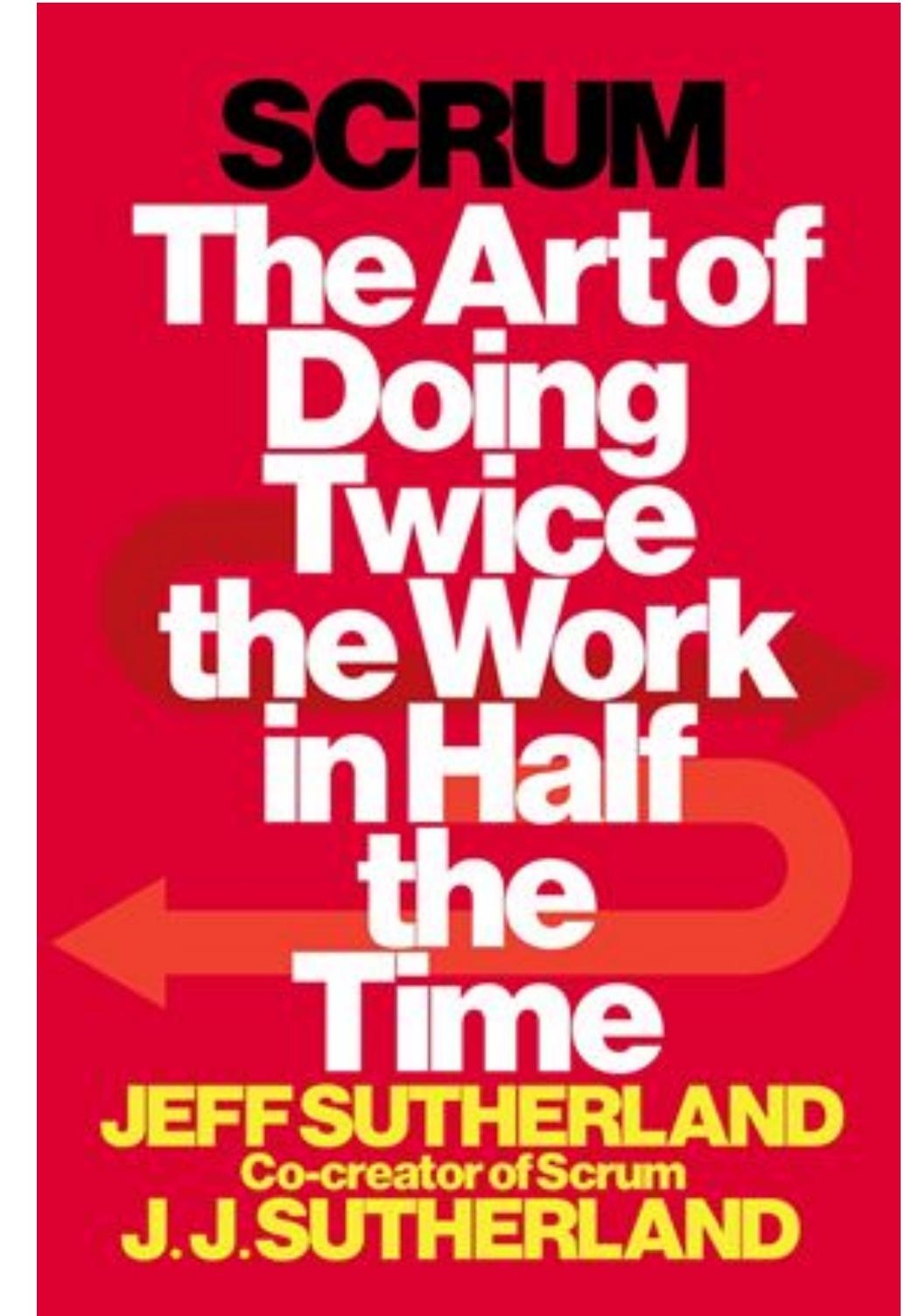
Delayed decision-making is the primary driver of project failure and budget overrun
決策延遲是專案失敗和預算超支的主要原因

Lean Means Process Efficiency > 25%

精實管理表示流程效率 > 25%

Customer
Will Use

- 25% of staff delivering useful stories
25%的員工交付有用的功能
- Process Efficiency = Value Added Work / Clock Time
流程效率 = 增值的工作/時間
- Goal: Improve Process Efficiency > 25%
目標：改善流程效率 > 25%
- “Twice the value at half the cost”
"以一半的成本交付兩倍的價值"

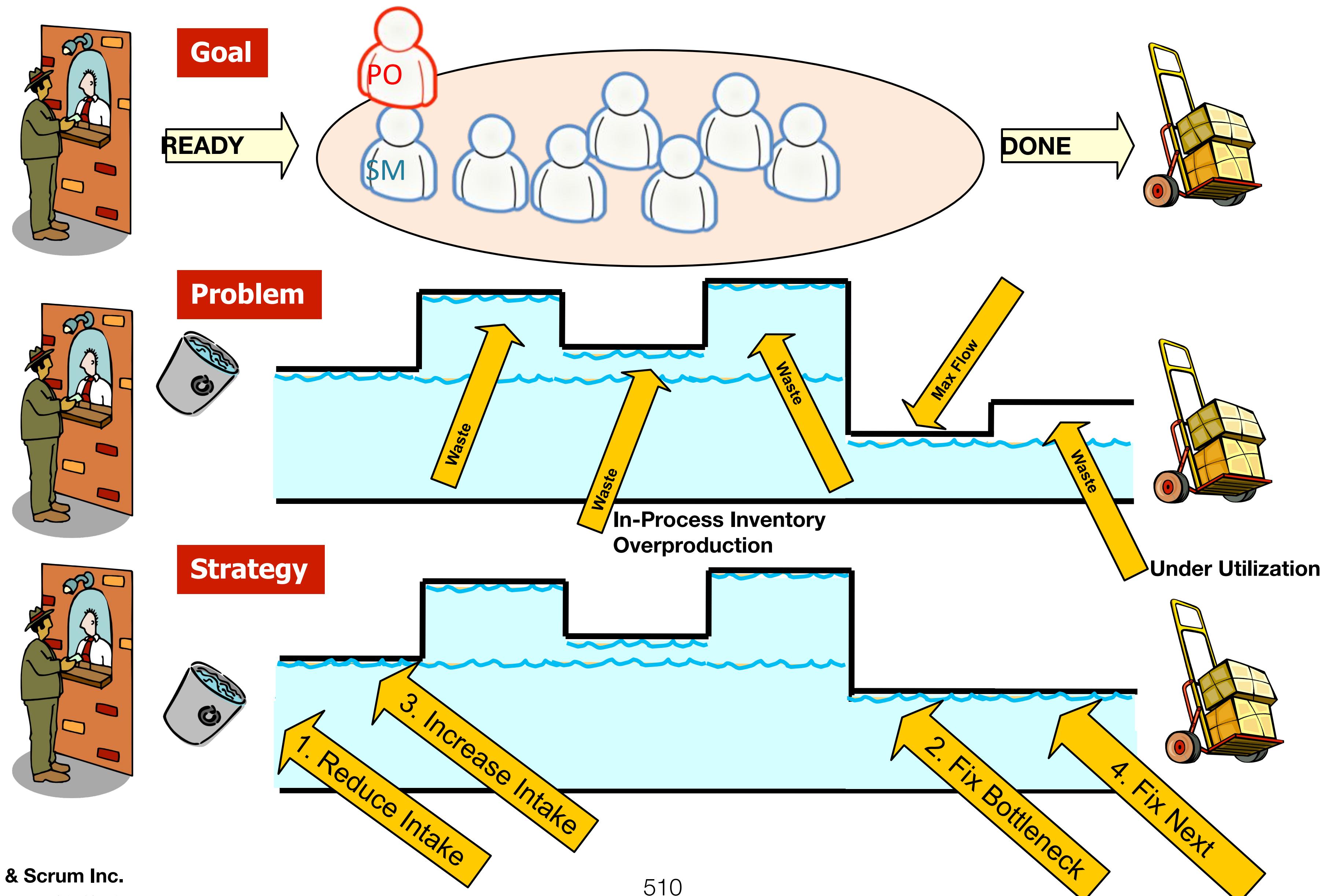


The Theory of Constraints and Value Stream Mapping

限制理論和價值流圖

Two tools that should be in every Scrum Master's toolkit
應該在每位Scrum Master工具箱裏的兩個工具

Theory of Constraints 限制理論 vs Continuous Flow 連續流

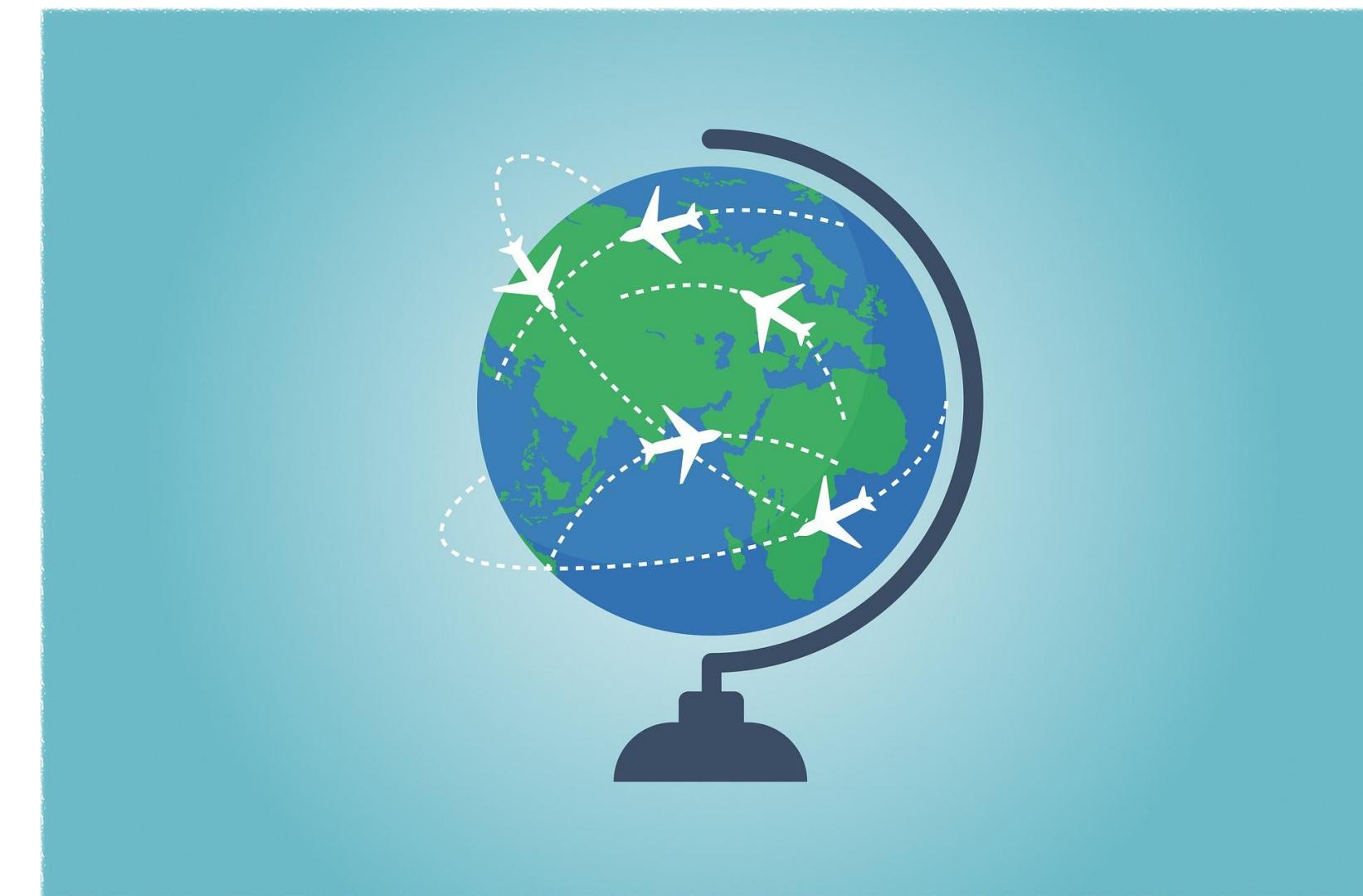


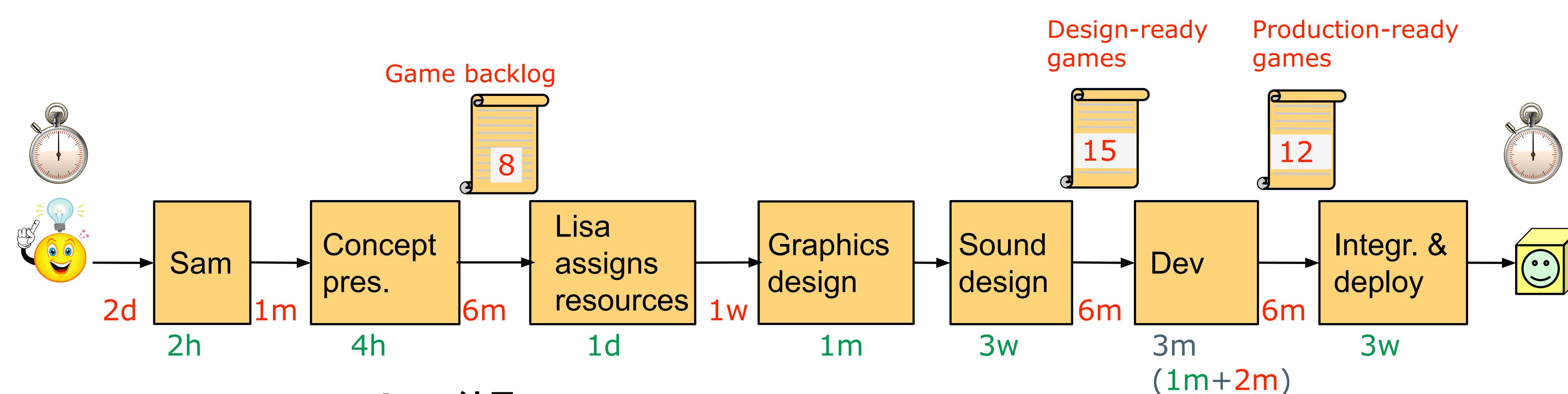
Case Study: Swedish Gaming Company

Developing Products >5x Faster

A real-life example of applying Value Stream Mapping and Scrum to dramatically speed up product development

採用價值流圖及Scrum去加速產品開發的實例



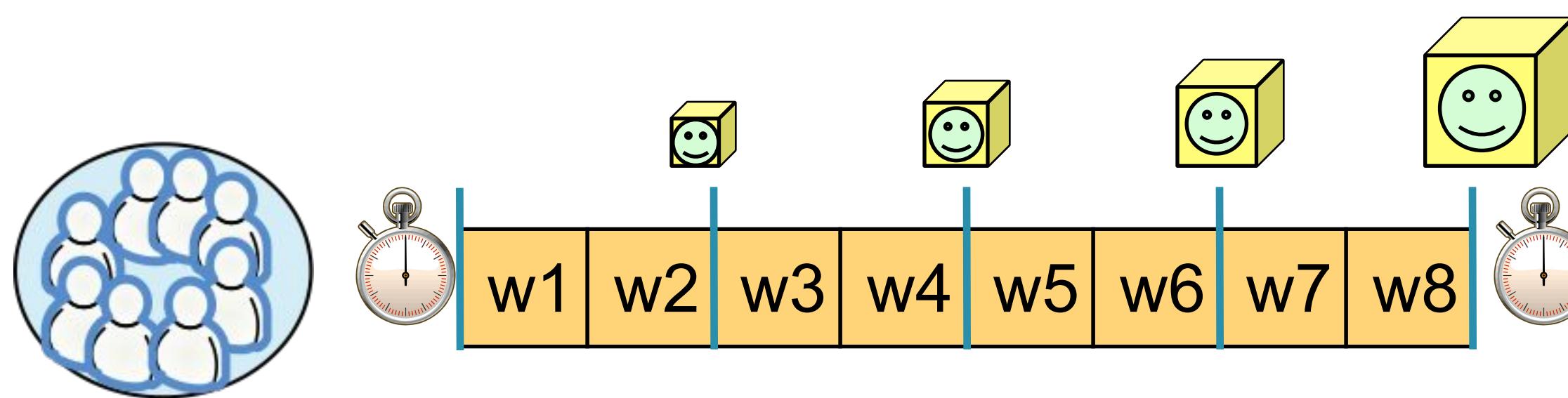


Results? 結果?

Games out of date 過時的遊戲

- ⇒ **Missed market windows 錯過市場機會**
- ⇒ **Demotivated teams 失去動力的團隊**
- ⇒ **Overhead costs 管理費用**

$$\frac{3.5 \text{ m value added time}}{25 \text{ m cycle time}} = 14\% \text{ Process cycle efficiency}$$



Ideal Stage

$$2 \text{ m cycle time} = 12x \text{ faster}$$

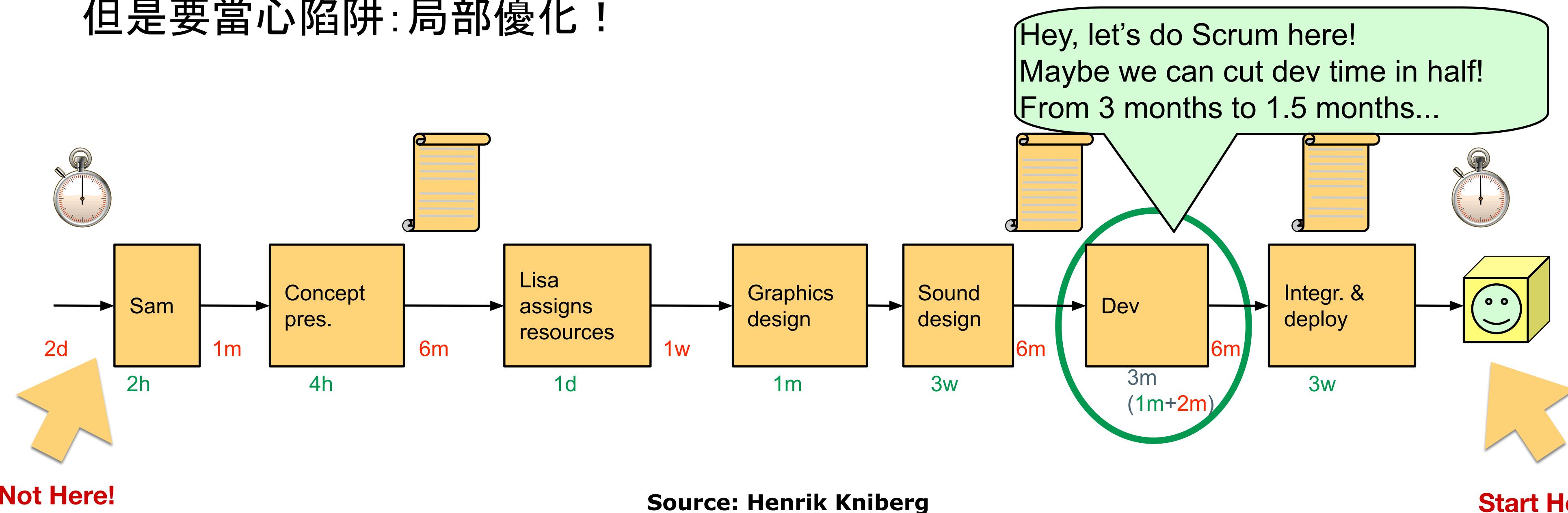
Quick improvement

$$3-4 \text{ m cycle time} = 6-8x \text{ faster}$$

Source: Henrik Kniberg

Case Study: Take-away Points

- Speeding up product development is often a matter of **improving the process rather than adding people**
加快產品開發通常是改善流程而不是增加人員
- **Value stream mapping** is a great tool for **spotting bottlenecks**
價值流圖是發現瓶頸的好工具
- **Scrum is a great tool for removing bottlenecks**
Scrum是消除瓶頸的好工具
 - But **beware of a trap: Local Optimization!**
但是要當心陷阱：局部優化！

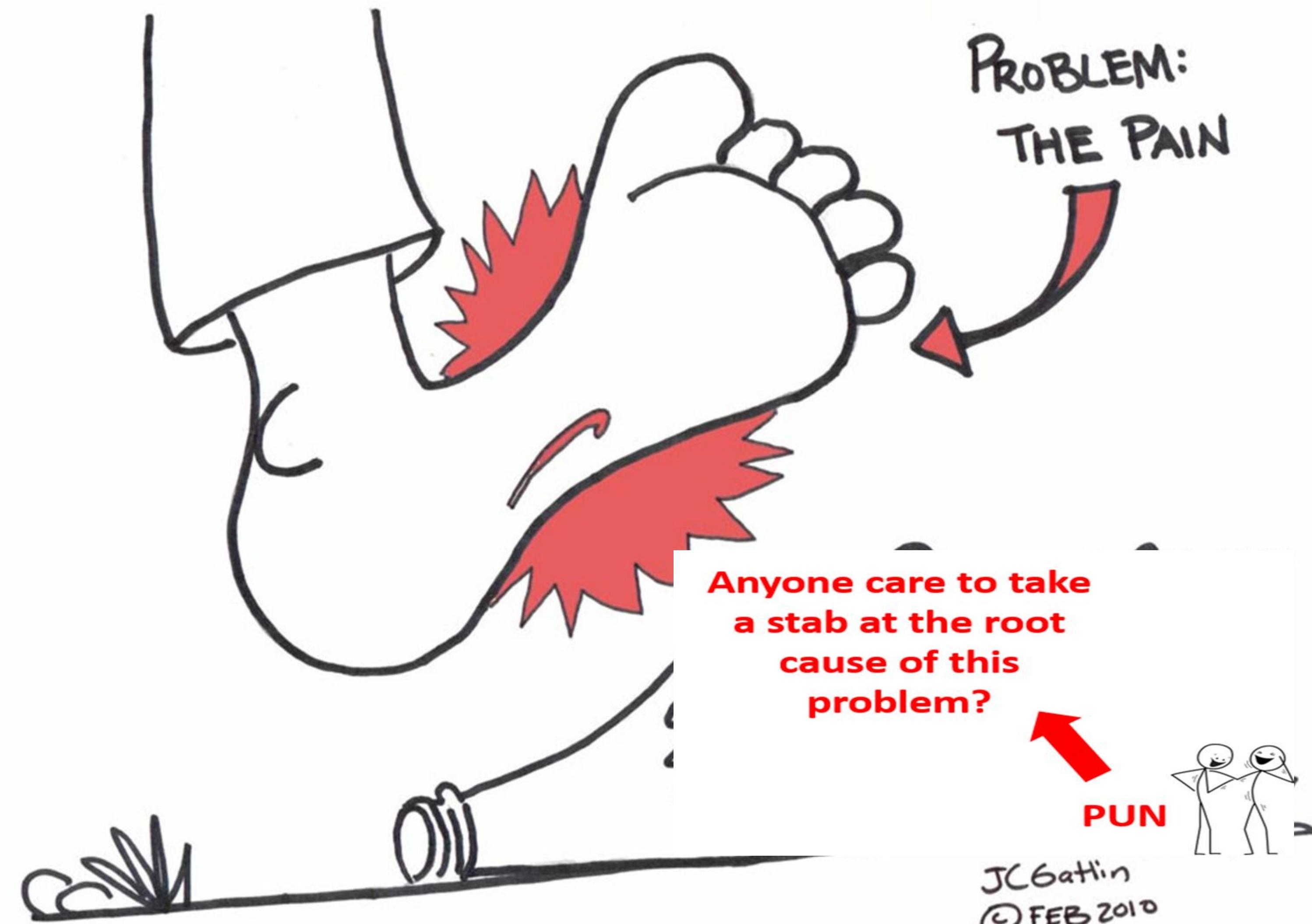


Root Cause Analysis 肇因分析

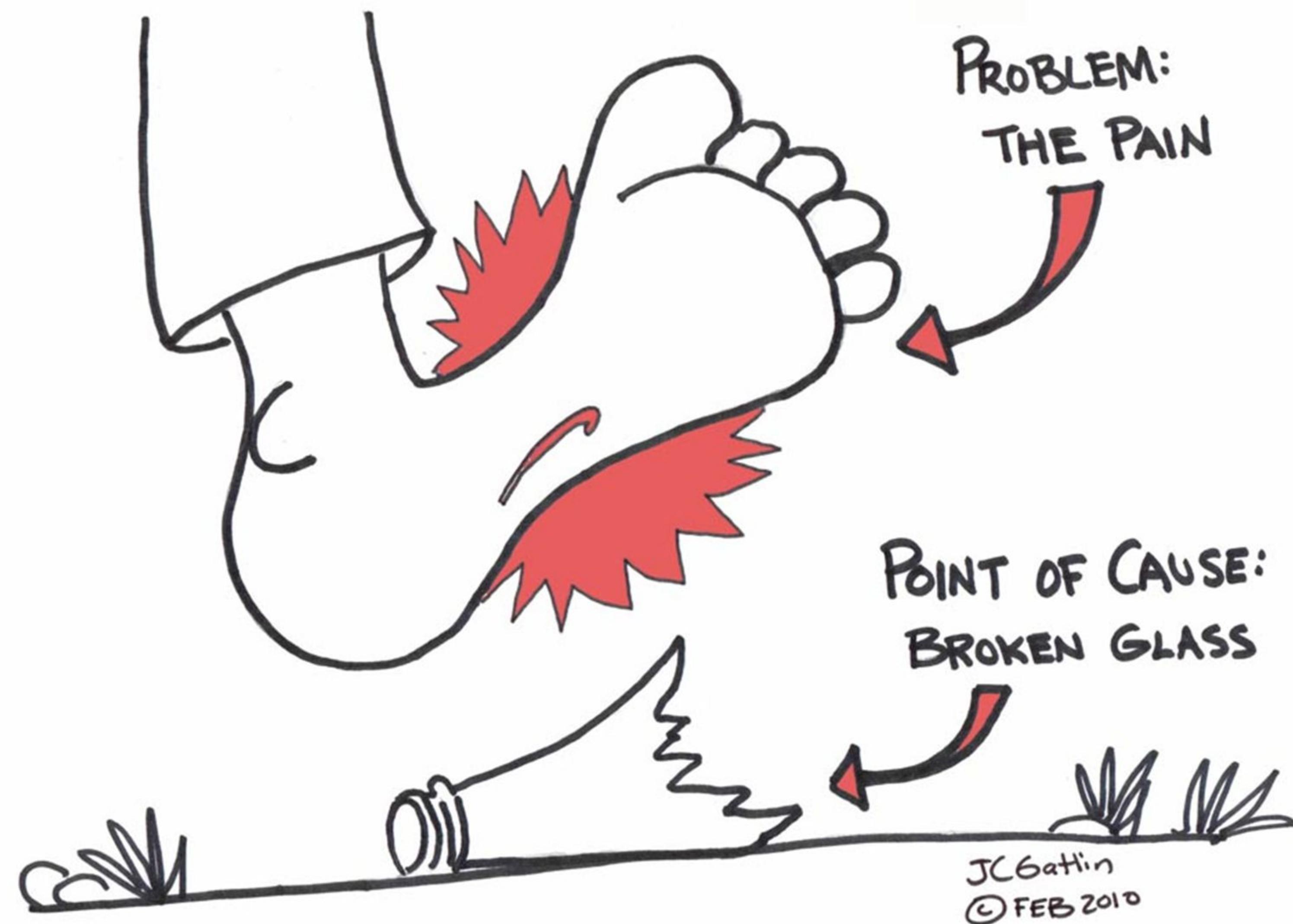
Perhaps the most important Lean tool

可能是最重要的精實工具

Understanding Root Cause 了解肇因



Point of Cause 肇因點 vs The Problem



Root Cause Analysis



Knowing when to stop asking “why”... 要懂得何時該停止問為什麼

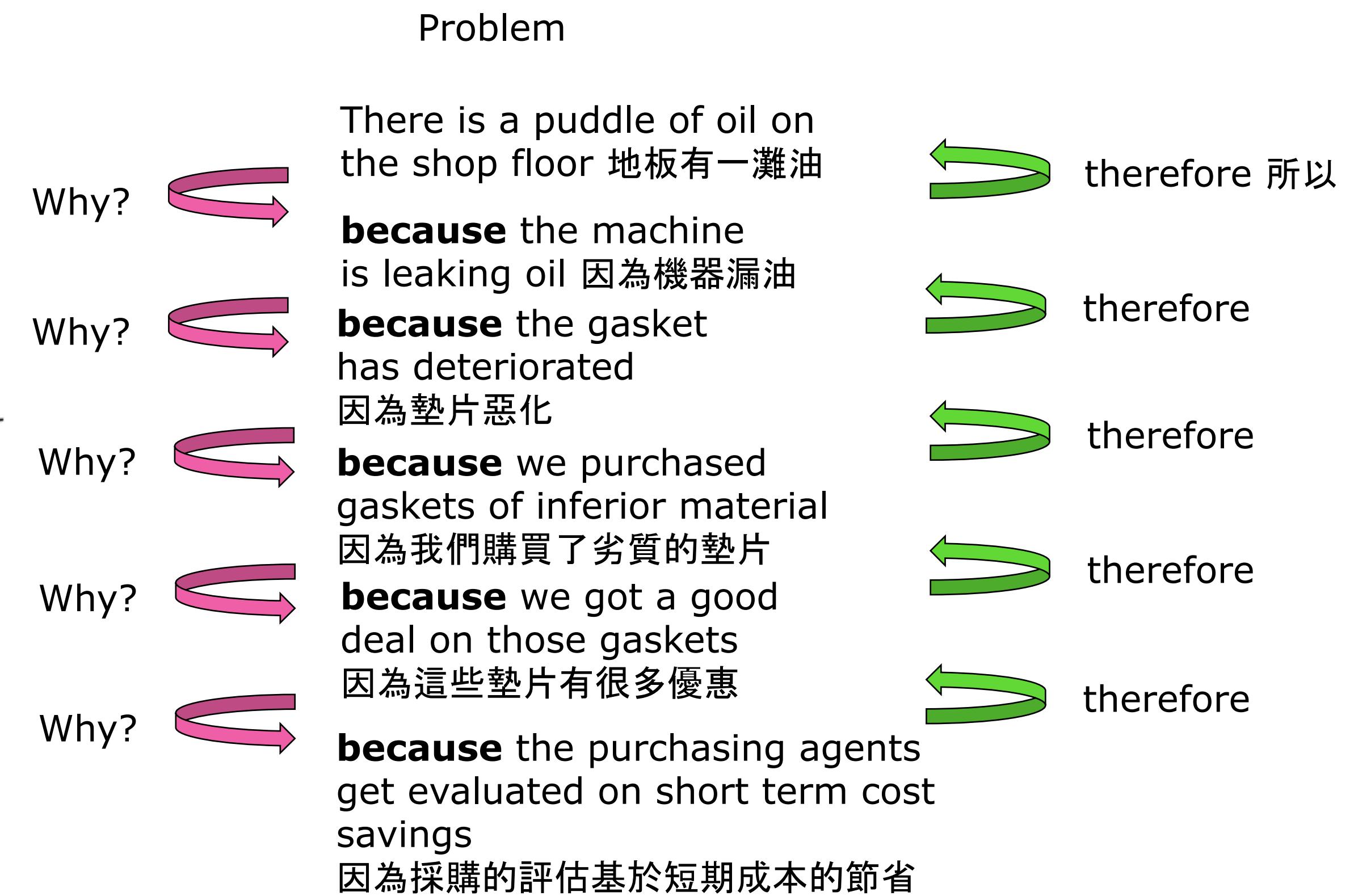
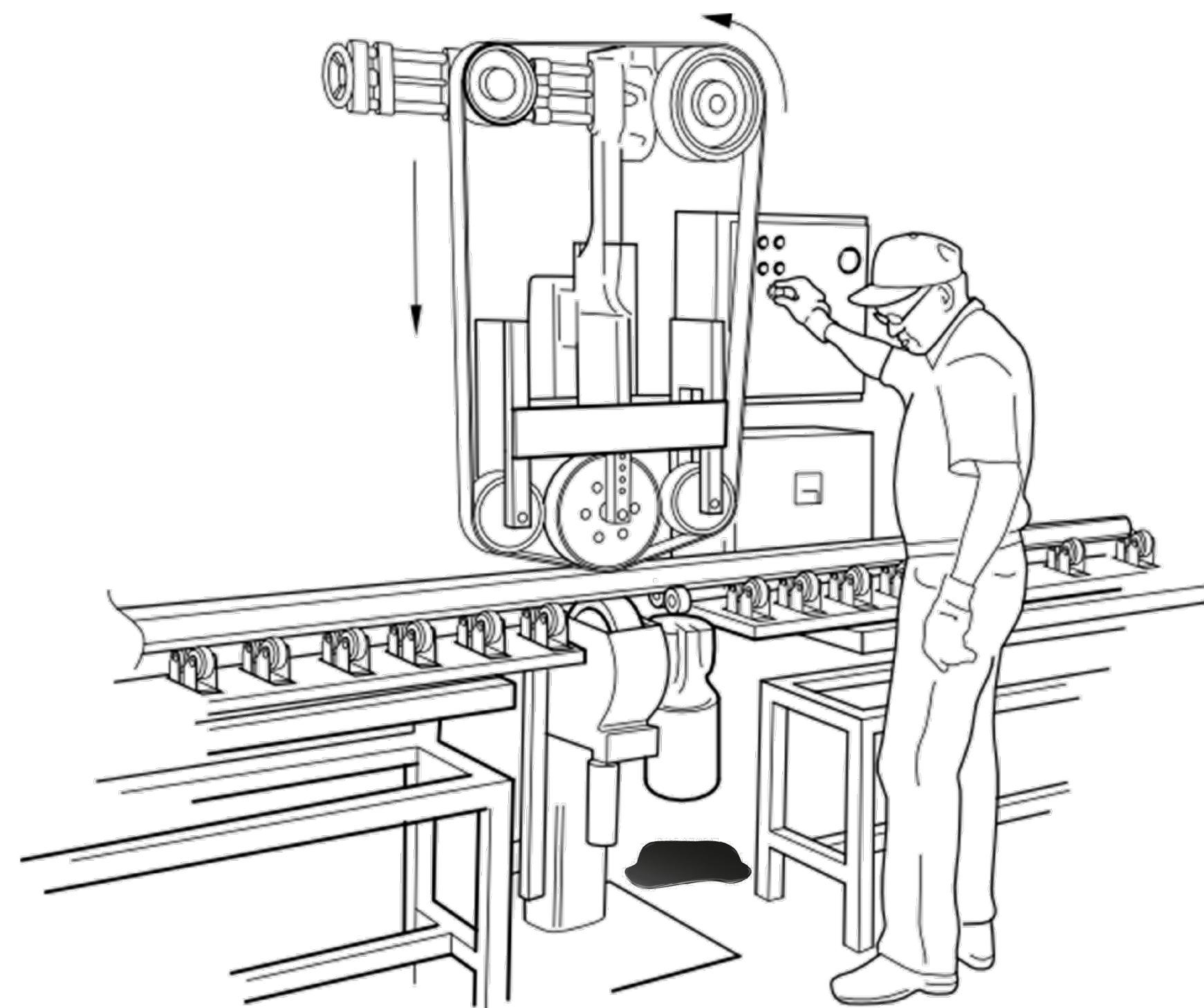
- **Know when you're beat**, even if you're right!
即使你是對的，但當你被打敗/講輸了時，你要知道你被打敗/講輸了
- **Know when you're at the limit of your capabilities**
當你已經發揮到極限時，你要知道你已經發揮極限
- We break down problems & **prioritize using the 80/20 Rule** so we address the right causes
我們拆分問題並且優先使用80/20法則，所以我們能找出正確肇因



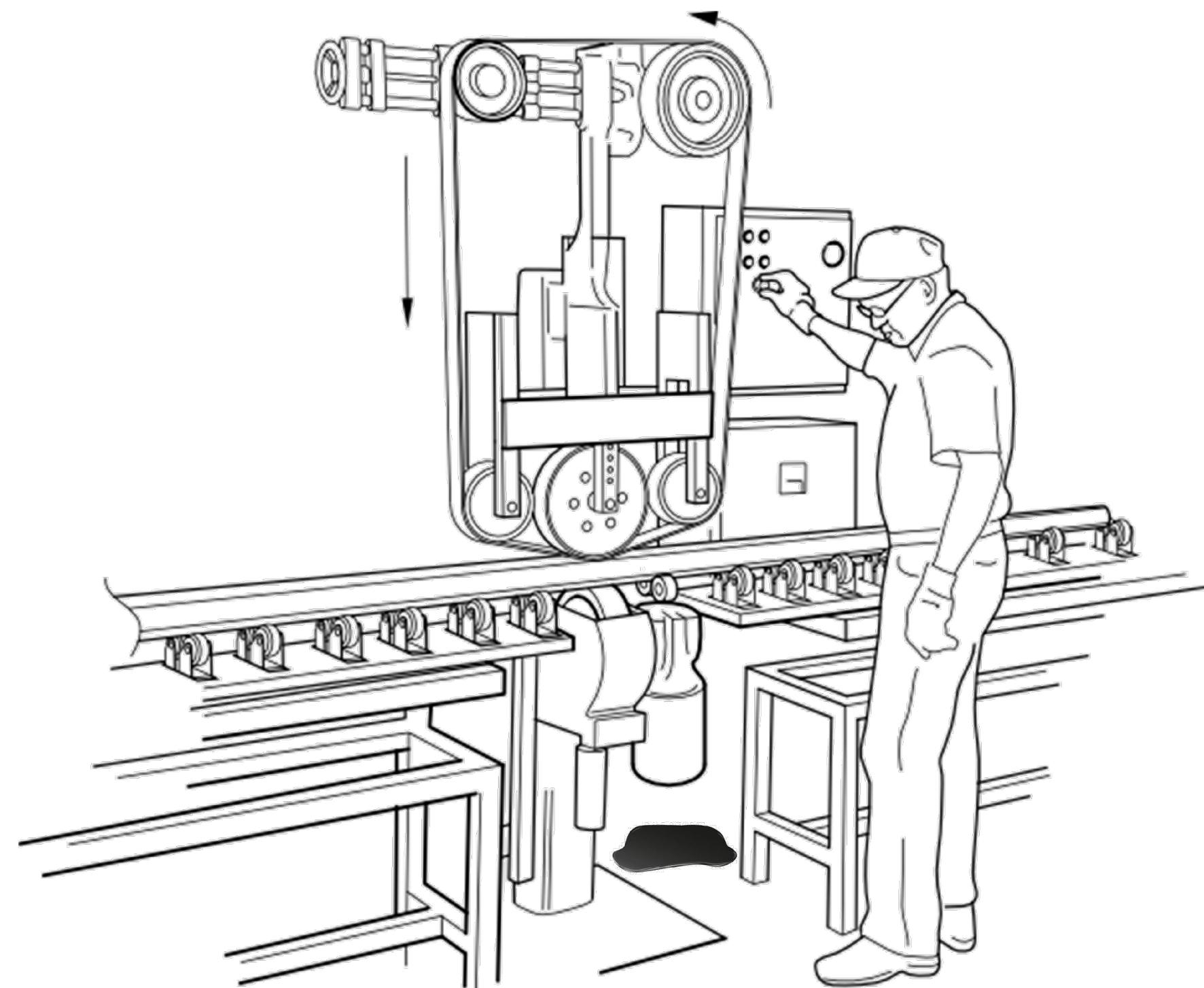
Spending millions of dollars to repair crumbling memorials
花數百萬元去修復崩塌的傑斐遜紀念堂

Why is the Jefferson Memorial crumbling?	Why?	Why?	Why?	Why?	Why?	Why?
Frequent Washings	Bird Droppings	Hundreds of Fat Spiders	Thousands of Midges	Emerge at dusk to mate	Attracted to the lights	Lights are on when they 1st emerge

Testing Why Analysis with Therefore 因為...所以



Proposing Countermeasures 提出對策



Problem	Countermeasure
There is a puddle of oil on the shop floor because the machine is leaking oil	Clean up the oil 把油清理掉 Fix the machine 修理機器
because the gasket has deteriorated	Replace the gasket 更換墊片
because we purchased gaskets of inferior material	Change gasket specifications 更改墊片規格
because we got a good deal on those gaskets	Change purchasing policies 變更採購政策
because the purchasing agents get evaluated on short term cost savings.	Change evaluation policy for purchasing agents 更改採購的評估政策

A3 Template for root cause analysis

Challenge	<p><u>Title:</u> The name of the problem/issue/challenge</p>	
	<p><u>Background:</u></p> <ol style="list-style-type: none">1. Why is this important?2. Challenge (the issue) description (What happened at When?)	
	<p><u>Current Condition (What, When, Where):</u></p> <ol style="list-style-type: none">1. How do things work today?2. What are the baseline metrics or current standard process (is this clear)?	
	<p><u>Goal / Target Condition:</u></p> <ol style="list-style-type: none">1. What are the expected outcomes for current process?	
	<p><u>Root Cause Analysis: (Tools like: Fishbone or 5 Whys)</u></p> <ol style="list-style-type: none">1. What happened this time? (more details)2. What is the root cause(s) of the problem? (Remember, the causes could be more than one, so if you are using 5 Whys, it could have different routes)3. Make sure you take all relevant factors into consideration (time, technology restriction, talents availabilities...)	
Plan	<p><u>Owner:</u> <u>Participants:</u> <u>Date Reported:</u></p>	<p><u>Date Reviewed:</u> <u>Status:</u> To Do or Doing or Done</p>
	<p><u>Countermeasures: (S.M.A.R.T. and these are experiment/hypothesis and need to be validated)</u></p> <ol style="list-style-type: none">1. Decide and make sure it's clear: Who will do What by When...2. Propose countermeasure(s) to address each root cause3. Make sure the countermeasure(s) will prevent the recurrence of the challenge4. Try the countermeasure that can yield the biggest value first	
	<p><u>Confirmation (Results):</u></p> <ol style="list-style-type: none">1. Actual result of each countermeasure (experiment)2. How is the new behavior now?3. Does it resolve the challenge?4. How will above countermeasures be verified and measured?5. Make sure you don't fix A break B	
Check	<p><u>Follow-Up (Learning):</u></p> <ol style="list-style-type: none">1. Does the countermeasure help?2. What have we learned?3. Are there other areas we should Adjust to reflect what we learned?4. What remains to be accomplished?5. Do we inform the team/department/organization about the changes?	
Solutions		
Act - Adjust		

Scrum Team Situations

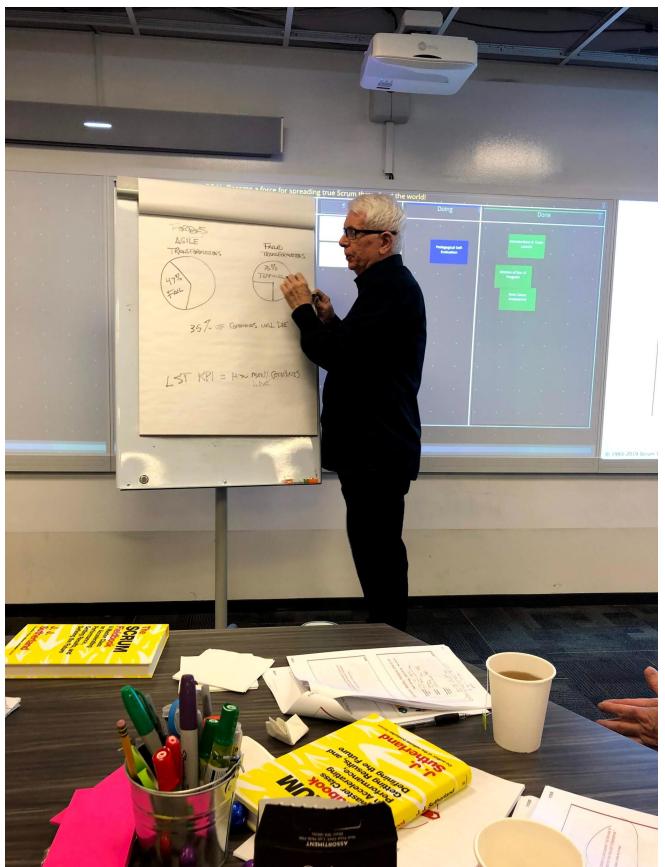
Scrum團隊的狀況劇

As a Scrum Master, I need to analyze Scrum Team challenges I might encounter, so that I know how to prepare and execute an action plan for improvement

身為Scrum Master, 我需要分析我可能會碰到的Scrum團隊的挑戰,
所以我可以在我知道如何做好準備並做出行動計畫來改善

Roundtable Discussion 圓桌討論

- Your team will be given situations to explore (or pick 2 of your choices)
你的小組會收到要去探究的狀況劇（或選2個）
- **Each case should be discussed** within the timebox
每個個案要在時間內討論完
- Scrum Masters, read each card aloud and **facilitate** a conversation around each potential problem
Scrum Master要大聲唸出每張卡片，並且引導繞著每個可能的問題做討論
- **Record the possible solutions** to share each case with the main group
記下可能的解決方案和大家分享個案
Hint: Which **Pattern** would help to improve the situation? 提示：哪個模式能協助改善情況呢？



MURAL



Case 1: The Poor Product Owner

蹩腳的 Product Owner

羅貝塔是你們團隊的PO，她幾乎花了全部的時間在拜訪客戶，因此，她指定另一個管理層的人湯姆，做為她不在時的代理人

連續三次Sprint，湯姆都沒能讓任何最終使用者或利害關係人參加Sprint回顧會議，而且，湯姆無法視覺化想像去呈現要完成的產品，總是要等看到可運作產品後才知道

每次Sprint回顧會議後，湯姆總有許多想法，並告知團隊去改變下一組產品待辦項目的需求

這會讓團隊做何感想？

身為一個PO，你能做什麼？

Case 2: The Defect Effect

瑕疵效應

團隊正在做一樣新產品，過去他們已經創造許多出色的產品，並且需要逐漸去升級

在Sprint執行到一半時，終端使用者發現他們最後的產品有一瑕疵，團隊中有多位是當時負責該待辦項目的成員

現在，每個人正忙於計畫中的待辦項目，而眼看新發布日期馬上就要到了，團隊似乎不確定誰應該去承擔解決這問題

團隊有哪些可行的行動途徑？

身為一個PO，你的建議是什麼？

Case 3: The Design Discord

設計意見衝突

派特和泰勒經由對培訓的投入，成為跨職能團隊，當他們被交辦一項新產品設計任務後，兩人設計出兩個截然不同且獨特的設計選項

當他們向團隊展示設計時，他們兩位似乎都相信各自的設計才是最棒的，並且才是對的方向

他們是如此熱情於自己的觀點，因而產生激烈爭論

身為一個PO，該如何做出選擇設計的決定？

身為一個SM，你能如何處理衝突？

Case 4: The Constant Carry-Over

經常把工作留到下次做

Sprint結束了！但很不幸的是，團隊只完成計畫交付七個待辦項目中的三項

對團隊而言，這已逐漸成為重複模式，造成發布計畫持續被調整，也明顯地讓管理階層很不高興

可能的根本原因會是什麼？

身為SM或PO，你將會如何幫助團隊？

Case 5: The Young Yes-Men

年輕的應聲蟲

每當團隊做待辦優化或在Sprint規劃會議討論待辦，法蘭克和鮑伯這兩位資淺成員都只會同意團隊中最資深的哈立德

每當被問到時，他們似乎總是抗拒說出他們自己的看法

可能的根本原因是什麼？

身為SM，你如何鼓勵他們大聲說出自己的想法？

Case 6: The Micro-Manager

老闆管太多

普麗西拉之前是公司的工程師，現在是最資深的副總裁，她分派了一個小專案給團隊

這是她三年來一直想完成的專案，但似乎總是半途而廢

每次普麗西拉經過團隊辦公區時，她總是會詢問進度如何，現在她決定最好來參加Daily Scrum

普麗西拉對於讓團隊了解成品看起來、感覺起來是怎樣，以及如何被完成，深感興趣

哪些事可能可以向普麗西拉說呢？

身為PO，你如何幫助團隊處理這狀況？

Case 7: The Daily Scrum

在一個九人團隊，有六位三十歲以下，三位超過五十歲，在最近的兩次Sprint，你已經注意到每天早上10:30開的Daily Scrum無法發揮功能了

即使公司每天的上班時間是早上10點到下午6點，但員工不是遲到，就是完全缺席

即使來了，他們似乎心不在焉，而且他們的貢獻僅僅只是關於目前工作項目的狀態報告而已

這樣行為的可能原因有哪些？

身為SM，你能如何幫助矯正這狀況？

Case 8: The Talkative Tech Lead

多話的技術主管

貝瑞以前的頭銜是公司的資深技術主管，當公司轉型去跑True Scrum後，他變成只是團隊的一份子

在他現在的團隊，貝瑞似乎獨攬控制所有的討論，並且否決或去挑其他每個人意見的毛病

你也注意到團隊成員現在會在進行工作前去找貝瑞討論他們該如何做事

這樣的狀況會產生哪些潛在的問題呢？

做為一個SM，你能如何指導團隊呢？



Case 9: The Reticent Retrospective

沉默無言的回顧會議

公司已完成重組，為了節省成本，也聘用許多海外約聘人員

現在團隊七個成員中，有四位來自其他國家，最近兩次Sprint，可以說是徹底失敗，而且團隊士氣似乎非常低落

但在回顧會議時，每個人都給快樂指數滿分五分，但似乎沒人對團隊的改善有任何建議

這樣行為的可能原因有哪些？

身為SM，你能如何鼓舞大家參與會議呢？

Case 10: The Isolated Instigator

孤獨的煽動者

胡利歐每天準時上班，他參加每早的Daily Scrum，但結束後，為了不被干擾，他馬上戴上耳機

他躲進自己的小世界裡，做他認為只特定給他或他的專長的事

除了煽動員工外，他幾乎沒有提出任何意見，也從不協助其他團隊成員的待辦項目，因為他認為那些不在他的專長領域

這樣的行為會導致哪些可能的後果？
身為SM，你能如何鼓舞員工的敏捷態度？

Case 11: The Collocation Conundrum

遠距難題

你們公司是一個跨國大型企業，你的三位團隊成員是在總部上班，兩位在相差三個時區的外地上班，最後兩位是在相差一個時區的家裡上班

規劃Scrum會議的協同合作變得很麻煩，然而在總部裡，有許多人具備與你團隊成員類似的技能

這樣的安排會產生哪些可能的問題呢？

身為SM，你有什麼可行的建議？

Case 12: The Operations Obstacle

營運障礙

丹的團隊已做到超高效能，並累積了十二個等待交付的”已完成”待辦項目

營運部門需要大量的文件才能交付給終端使用者

身為Product Owner，他因為無法向顧客發布已通過測試的零瑕疵產品，而感到非常氣餒

這可能會讓團隊做何感想？

身為一個Scrum Master，你能建議公司做什麼改變去提高流程效率？

Case 1: The Poor Product Owner

The PO of your team, Roberta, spends almost all her time visiting clients. As such, she has named someone else from management, Tom, to be her proxy while she is away.

Tom has failed to get any end-users or stakeholders in the room for the Sprint Review for 3 Sprints in a row. Tom has trouble visualizing things and can only do so after seeing the working product.

After each Sprint Review, Tom gets a lot of ideas and advises the team to change the requirements of the next set of product backlog items.

How might this make the team feel?

As a Scrum Master, what could you do?

Case 2: The Defect Effect

The team is working on a new product. They have created many great products in the past and are required to upgrade them over time. A defect is discovered mid-Sprint by an end-user in their last product. The backlog item from which the defect came was worked on by multiple members.

Everyone is currently working on planned backlog items for a new release whose date is pending shortly. The team seems unsure of who should take on the issue.

What are some possible action paths for the team?

As a Scrum Master, what is your recommendation?

Case 3: The Design Discord

By investing in training, the team has become cross-functional. When given a new piece of the product to design, Pat and Taylor, have come up with two very different and unique design options.

When presenting them to the team, they both seem to believe that their design is the best and the right way to go.

They are so passionate about their views, that a heated argument occurs.

How could the decision of which design to choose be made?

As a Scrum Master, how could you handle the interpersonal conflict?

Case 4: The Constant Carry-Over

The Sprint is over! Unfortunately, the team only completed 3 out of the 7 backlog items they planned to deliver.

This is becoming a repeating pattern for the team causing the release plan to be constantly adjusted and is making management visibly unhappy.

What could be the possible root causes?

As a Scrum Master, how will you help the team?

Case 5: The Young Yes-Men

Whenever the team does Backlog Refinement or tasks out backlog items at Sprint Planning, Frank and Bob, the two junior members, always and only agree with the ideas of Khalid, the most senior team member.

Whenever asked, they seem resistant to ever give a voice to their own opinions.

What could be the possible root causes?

As a Scrum Master, what possible ways could you encourage them to speak up?

Case 6: The Micro-Manager

The team has been given a pet project from Priscilla, a former engineer and one of the most senior Vice Presidents in the company. It is a project she has been trying to get done for 3 years but always seems to fall by the wayside.

Every time Priscilla passes by the team space, she asks for an update and has now decided that it might just be better for her to attend the Daily Scrum.

Priscilla is very interested in having the team understand exactly how she wants it to look, feel, and be built.

What are some possible things that could be said to Priscilla?

As a Scrum Master, how could you help the team handle the situation?

Case 7: The Daily Scrum

A 9-person team has 6 members under the age of 30 and 3 members over the age of 50. For the last two Sprints, you have noticed dysfunction at the Daily Scrum, which is held at 10:30a. Even though company working hours are 10a - 6p daily, people have been coming late or missing it entirely. If they are present, they seem distracted and barely contribute anything more than a status report on just their current work item.

What could be some causes for this behavior?

As a Scrum Master, how could you help rectify the situation?

Case 8: The Talkative Tech Lead

Barry used to have the title of Senior Tech Lead in the company. When they switched to doing true Scrum, he became just another team member.

On his current team, Barry seems to dominate all the discussions and shoots down or finds fault in everyone else's ideas.

You have also noticed that team members are now going to Barry to discuss how they will do things before ever working on them.

What are some potential problems that could arise from this situation?

As a Scrum Master, how could you coach the team?

Case 9: The Reticent Retrospective

The company has done a reorganization and to cut costs has hired many contractors from overseas.

The current team of 7 now has 4 from another country. The last two Sprints have been almost total failures and morale seems very low. Yet at the Retrospective, everyone rates their happiness a 5 out of 5 and no one seems to have any suggestions for team improvement

What are some possible causes of this behavior?

As a Scrum Master, how can you encourage participation?

Case 10: The Isolated Instigator

Julio comes to the office every day on time. He participates in the morning Daily Scrum, but immediately afterwards, he puts on his headphones so as not to be disturbed.

He retreats into his own little world consumed with tasks he feels are specific just to him or his domain knowledge.

He rarely offers any comments other than to stir people up and never offers to assist other team members with backlog items he considers outside of his area of expertise.

What are some possible outcomes of this behavior?

As a Scrum Master, how can you encourage a more agile attitude?

Case 11: The Colocation Conundrum

Your company is a multinational giant. Three of your team members sit at headquarters. Two of your team members sit off-site, three time zones away. The final two, work from home only one time zone away. Planning the Scrum events has become troublesome as has collaboration. There are many other people at headquarters with similar skills to those of your team.

What are some possible problems that could arise from this arrangement?

As a Scrum Master, what possible suggestions you could make?

Case 12: The Operations Obstacle

Dan's team has become hyper-productive and has 12 "Done" items waiting to be shipped.

Operations requires extensive documentation to push anything to an end user.

As Product Owner, he is becoming very frustrated at having tested, defect free product, that he cannot release to his customers.

How might this make the team feel?

As a Scrum Master, what changes could you suggest the company make to streamline this process?

Descaling 去除惡習

As a member of a Scrum Team, I need to remove bad habits before I scale Scrum to more teams, so that I don't magnify the dysfunctions

身為Scrum團隊的成員，我需要在我大規模化Scrum團隊之前，改掉壞習慣，
所以我不會擴大障礙

Descaling - Key Takeaways 關鍵重點

- A **True Scrum Team** is equivalent to 5 lazy Scrum Teams
一個真Scrum團隊等於5個慢吞吞的Scrum團隊
- **Descaling is required - ending bad habits is essential before scaling, or you will only scale bad habits!**
必須去除惡習 - 在展開大規模之前，終結不良惡習十分重要，否則您只會放大惡習
 - **Stabilize the teams** 穩定團隊: NO ADDING OR REMOVING MEMBERS
 - **Stop Interruptions** 停止插件干擾: FOCUS IS KEY
 - **Institutionalize Kaizen Thinking** 養成改善的想法: CONSTANTLY LOOK TO IMPROVE 不斷尋求改進
 - No off-backlog work or split team members
不做沒列在待辦事項的工作或分裂團隊成員
 - Prioritize **Tech Debt & Defect Fixing**
優先排序技術債和瑕疵修改



Hyper-productive Patterns That Will Double Velocity

高產能模式會加倍速率

- **Teams that Finish Early Accelerate Faster** 儘早完成交付的團隊加速更快
- **Small, Stable Teams** 穩定的小型團隊
- **Ready Backlog** 準備就緒的產品待辦
- **Yesterday's Weather** 昨天的天氣
- **Swarming** 蜂擁行動
- **Interrupt Pattern** 插件緩衝模式
- **Good Housekeeping** 良好的即時清除管理 (Daily Clean Code)
- **Happiness Metric** 幸福指標
- **Quantum Entanglement** 量子糾纏
- **Emergency Procedure** 緊急程序
- **Scrumming the Scrum** - 用Scrum來改善Scrum
- **T-shaped People** T型人
- **In Sprint Testing Sprint** 裡面測試
- **Point-Based Estimation** 以點數為依據的估算

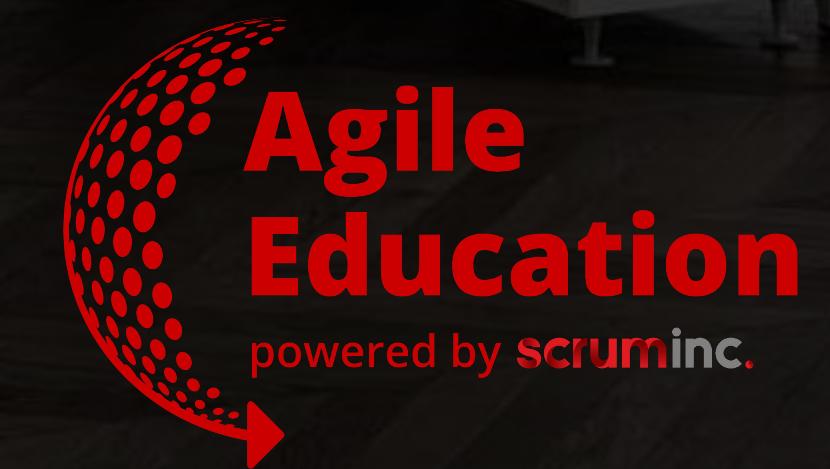
www.scrumlop.org

An Agile and Digital Transformation Journey

It's like changing the airplane engine while flying

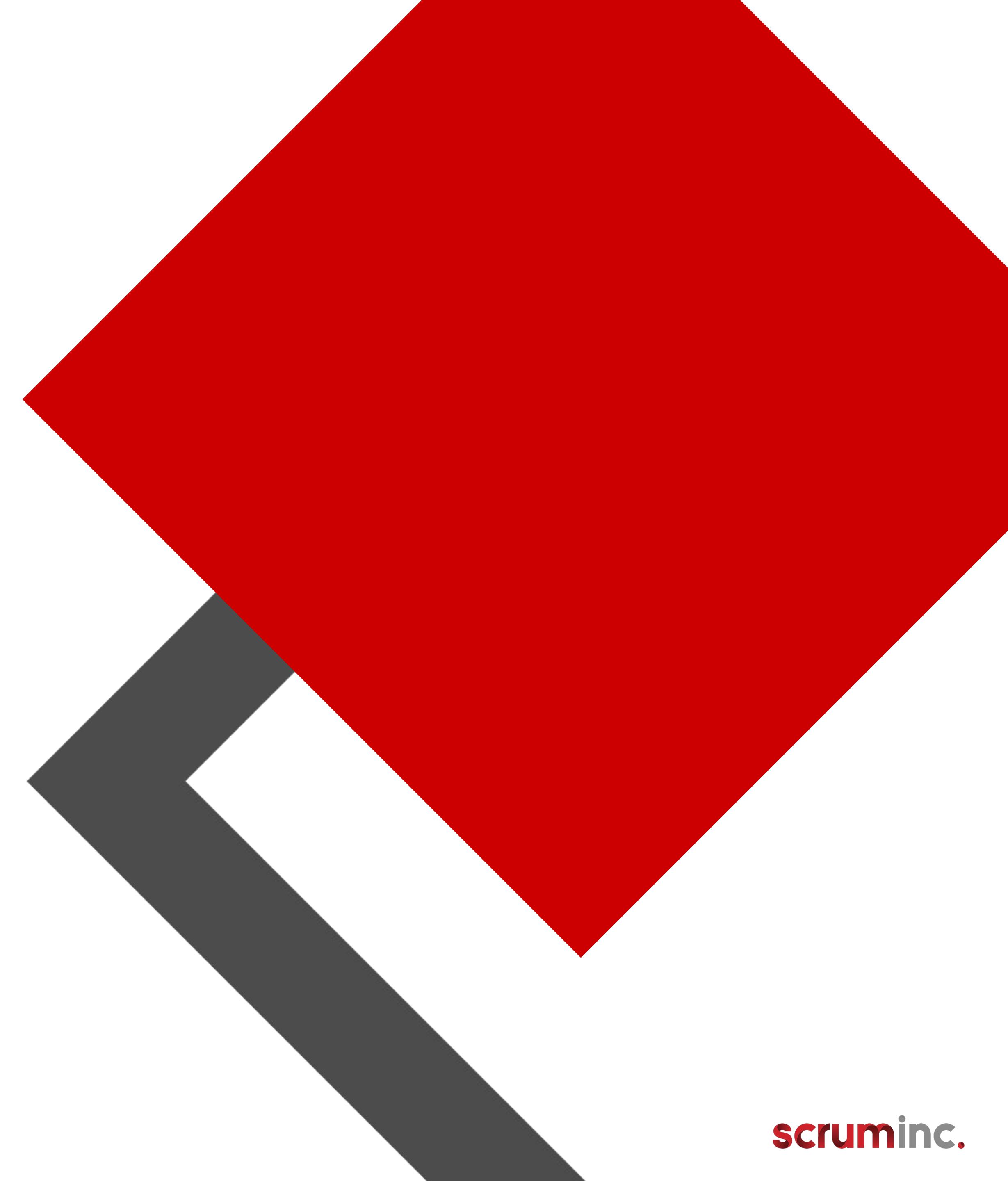
May 2022

Andrew Lin



Before

Where business as usual



Before: Challenges & Opportunities

Challenges:

- Organization (a non-profit media/publisher with 200+ employees) with outdated technology
- Department silos
- Business as usual, keep the old lights on
- And no new product release for many years
- “We did whatever they told us to do”

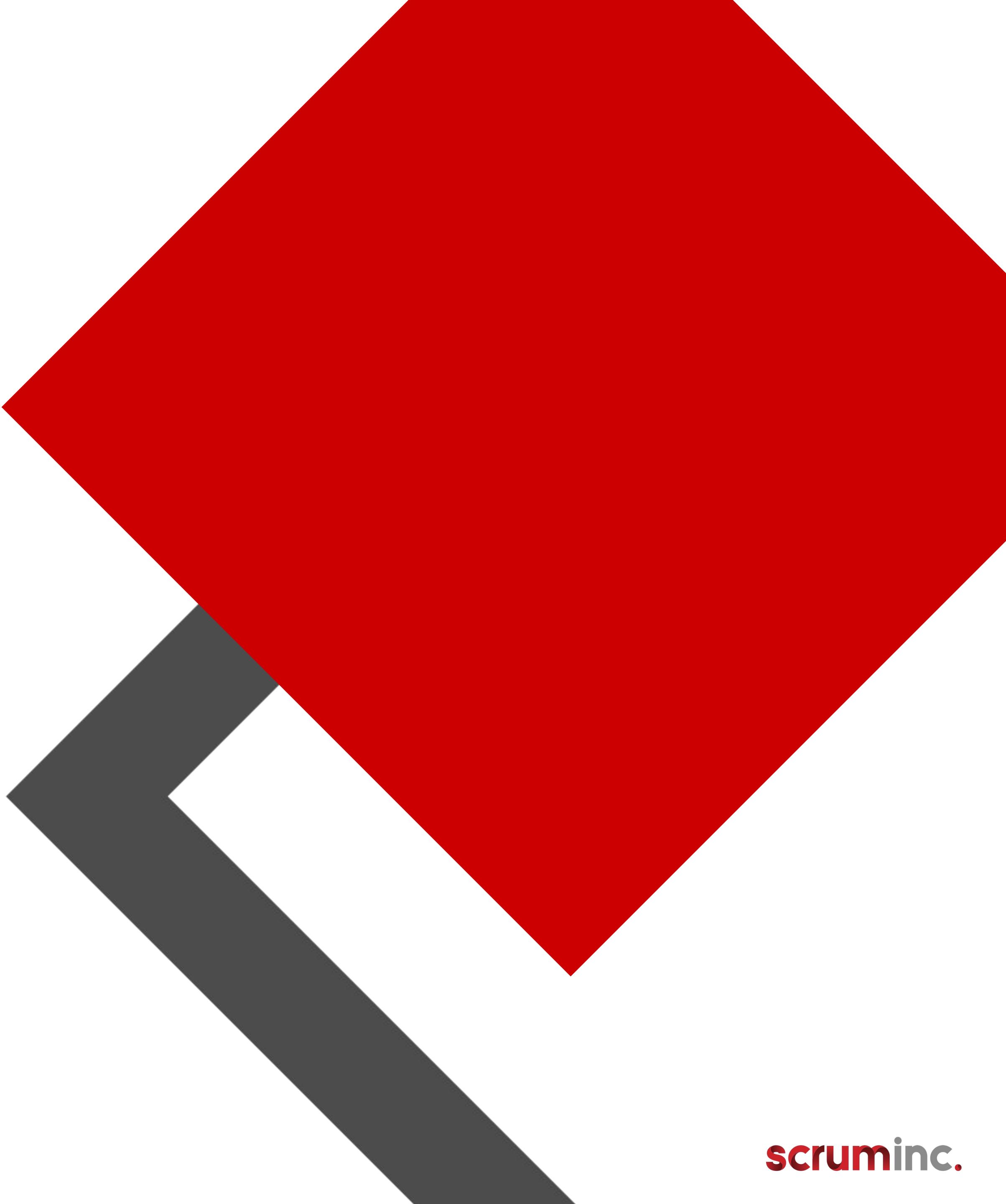
Opportunities:

- Team level Scrum and Training
- Reference model
- Cross-Teams Coordination
- Start Product Owner cycle
- Stop Wasting and Start Delivering
- From very outdated technology to catching up and now cutting-edge technology



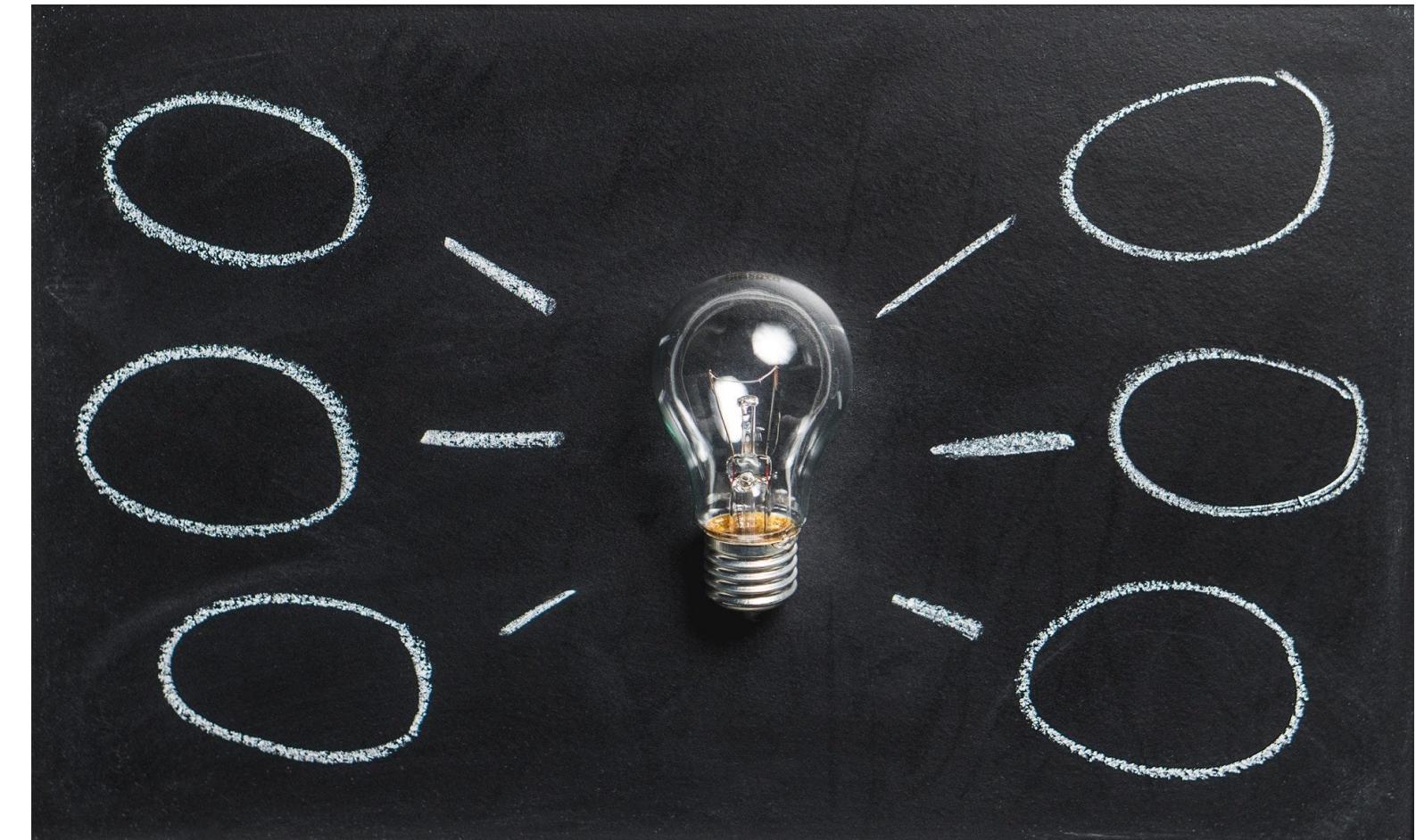
Current

New products and features are coming out faster than customers can handle



Current: Innovation, customer loves the new products, and teams are happy

- Kaikaku in Product Owner Cycle:
 - EMS, 20+ Product Owners
 - From Vision to Release: **New major product release twice every year**
 - New feature **every Release and every 2 weeks**
- Kaizen in Scrum Master Cycle:
 - Continuous Improvement (Teams **Say-Do ratio ~ 90%**)
 - Cross-Team Coordination
 - Some SM Components but no EAT
 - Learning organization
 - From business as usual to twice the value in half the time
- Keep the lights on
- **Regain 40% of the organization revenue in 2 years**



Digital Transformation Milestones

- 7 years ago
 - Everything is running on IBM/Lotus Domino
- 2 Years moves Web to Drupal, CMS back-end with a ECMS with NoSQL
 - Keep this monolithic architecture running for 3 years while planning for next generation architecture
- 2 Years moves Web from Drupal to **Serverless and Microservice** architecture running on AWS with 2 Scrum Teams
 - **Impacts:**
 - Improved **Scalability**
 - Better **resilient and error isolation**
 - Better data security
 - Integrate with **DevOps**
 - Easier to understand, maintain, and troubleshooting
 - Be able to **track** user's journey
 - **More selections** for technologies and programming language



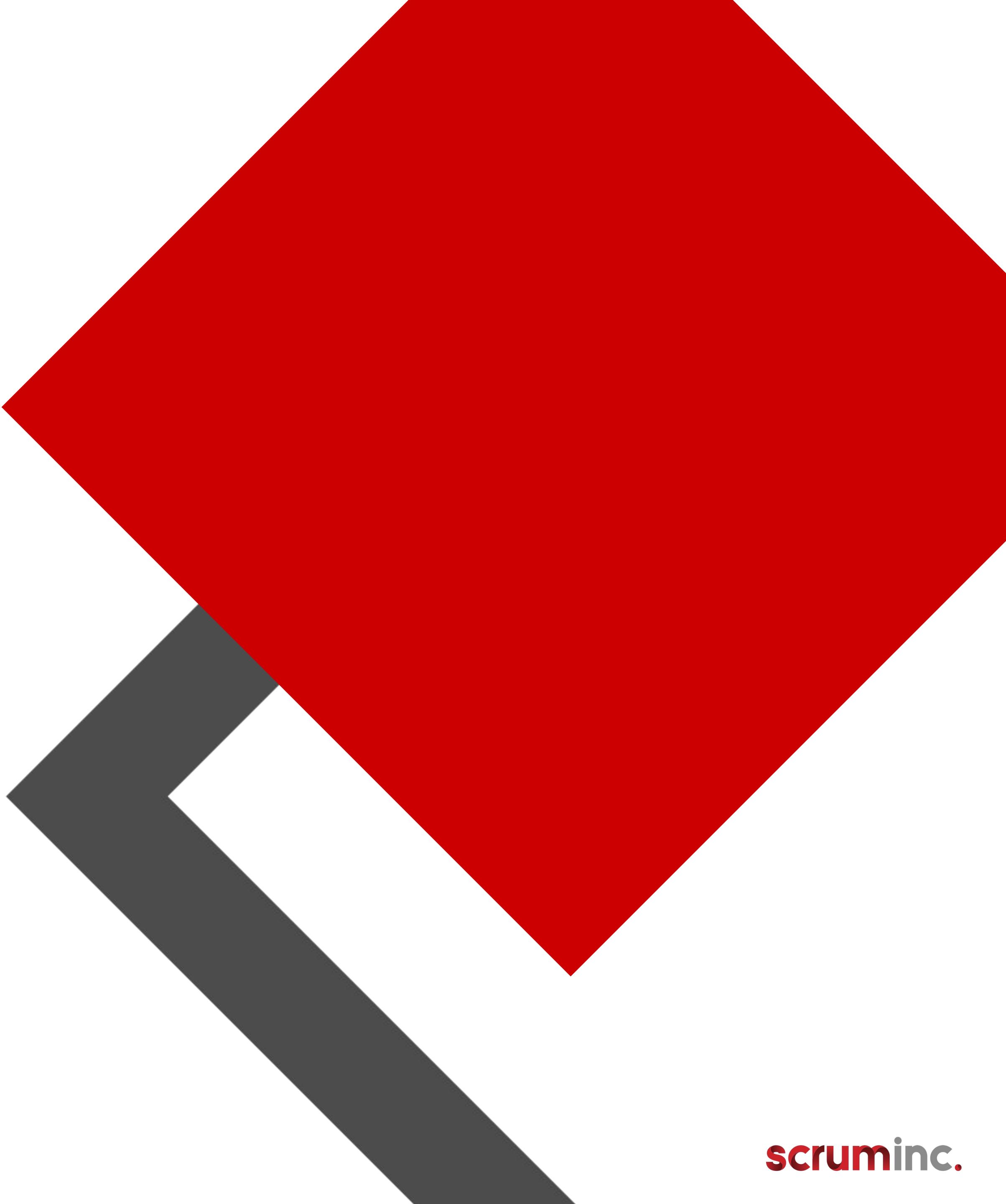
Future:

- Better alignments with products/talents/expectations
- Learn to discontinue products
- Feedback
- EAT
- Promote quick decision making
- Continue Refine technology stacks
- From collecting data and presenting data to Prediction and Make it Happen
- and Keep it going



What we learned

From Scrum teams, to transform, to scale, and beyond



What we learned:

- Digital Transformation starts with Agile
- Digitally enabled organization is a Scrum at Scale organization:
 - From monolith to microservice - **Break something big into smaller pieces**
 - Track everything - **Feedback, Continuous Improvement, Kaizen**
 - Push anytime - **Cutting waste, speed, quality, and Deliver**
 - Innovation - **Short iteration, validation, and Kaikaku**
- It's a journey, it's about **Change**, It's like changing the airplane engine while flying, and you need to continue serving drinks and snacks
- **Enjoy**



A Brief Intro of Scrum@Scale

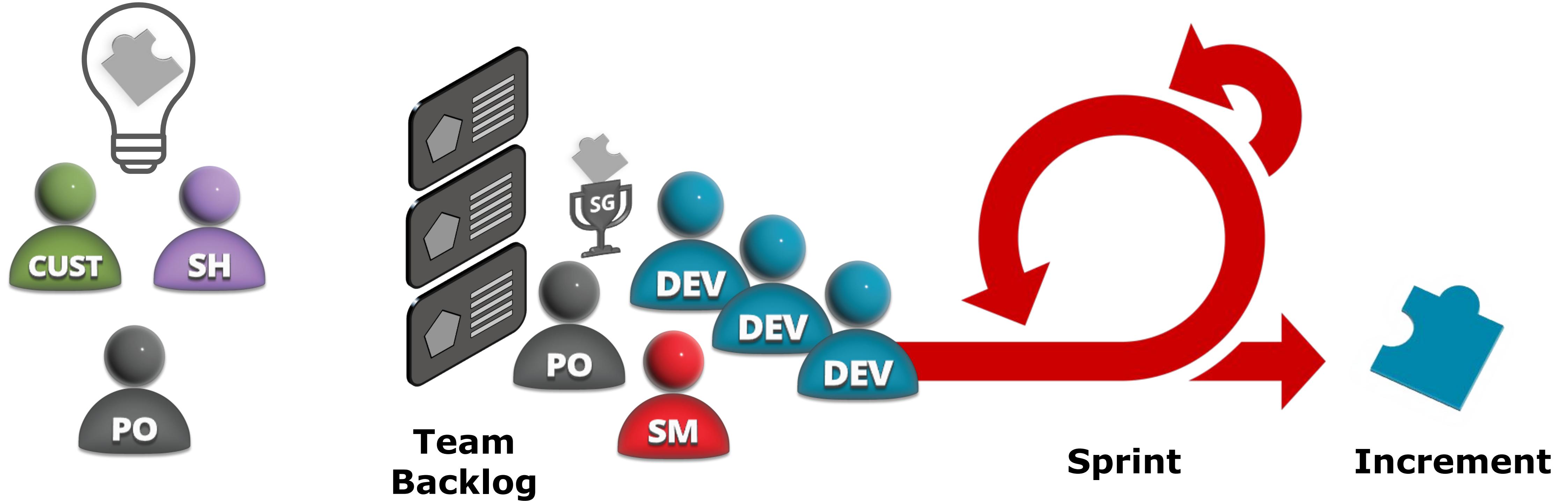
大規模化/多團隊Scrum

As a team of teams (SoS), I want to deliver an integrated product increment and remove impediments so that we can deliver twice the value in half the time

身為Scrum團隊之一(SoS), 我想要交付一個整合的產品增量並去除障礙,
所以我們才可以用一半的時間交付兩倍的價值

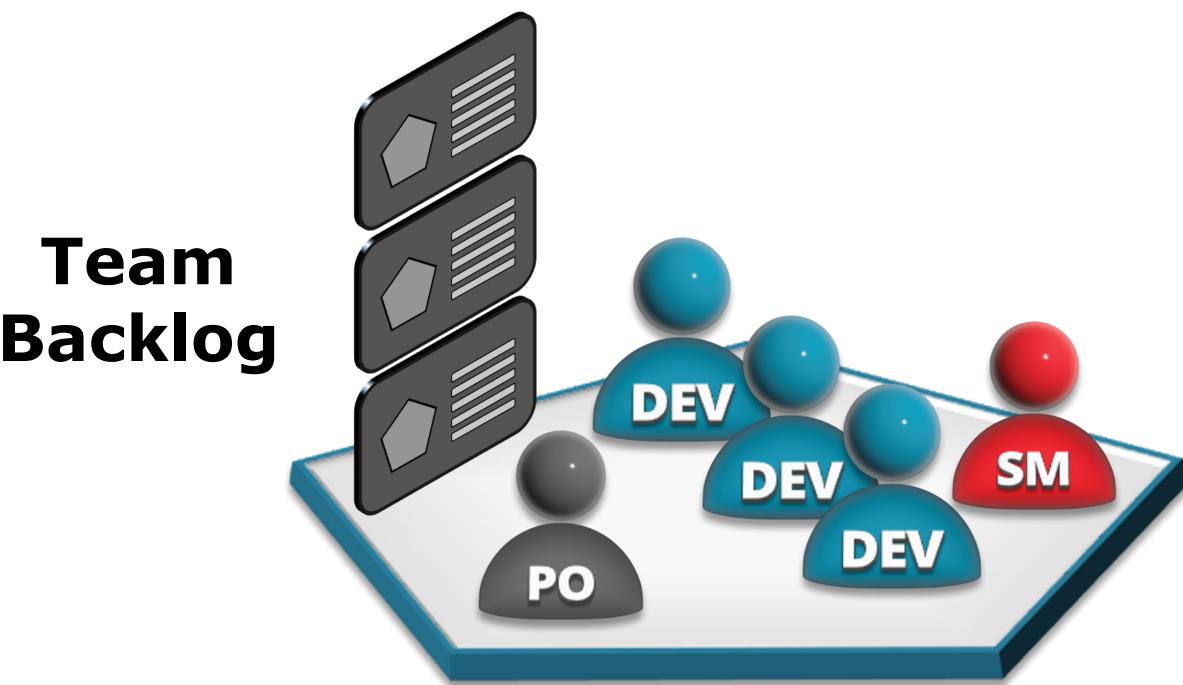
Delivering with Scrum

用Scrum來交付



Delivery with Scrum@Scale

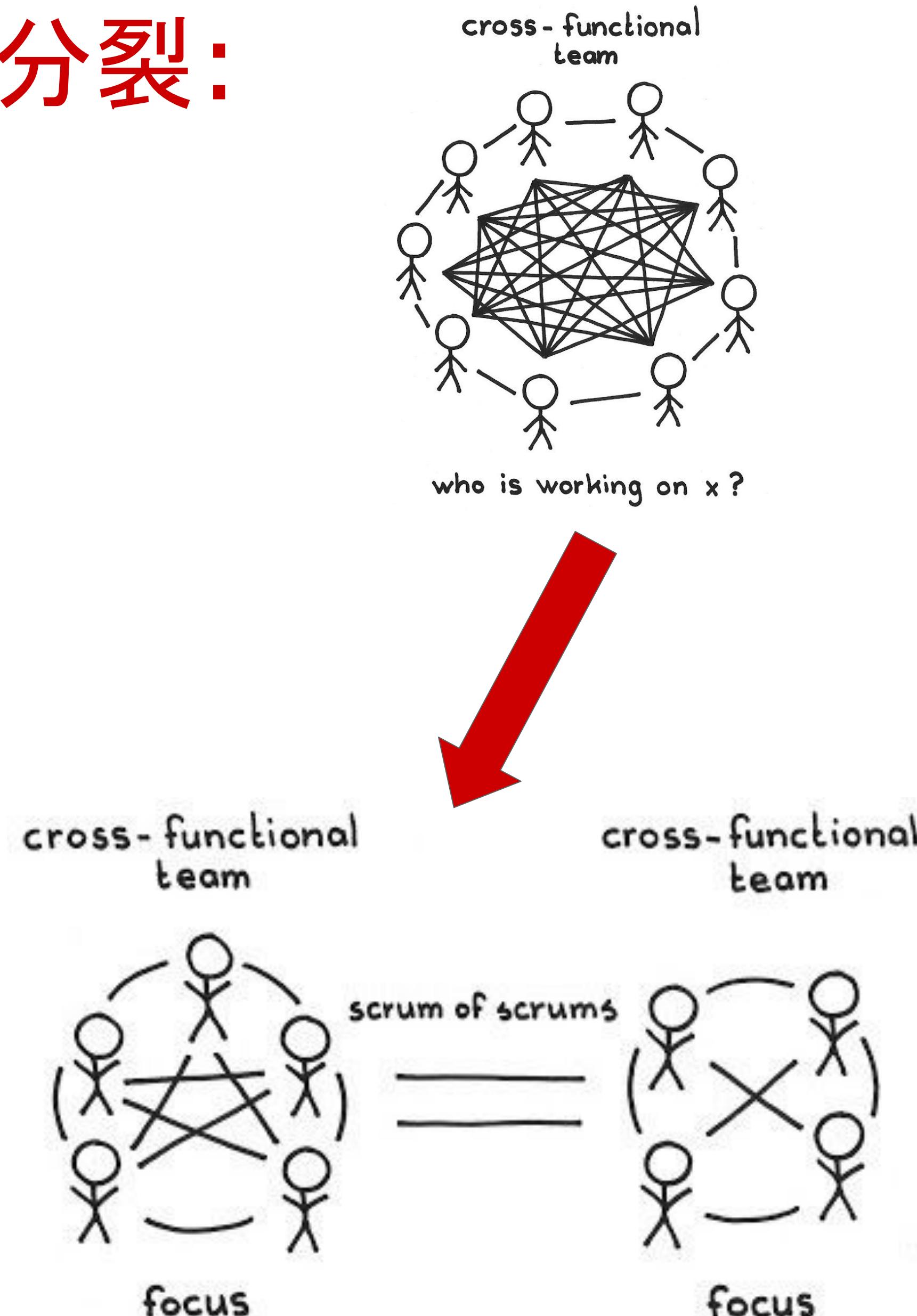
用大規模化多團隊Scrum來交付



If a Scrum Team is an independent path to production, why would we have to scale at all?
如果Scrum團隊是獨立生產, 我們為什麼還要大規模化/多團隊呢?

Scaling Pattern: Mitosis 有絲分裂： Self-Organizing at Scale

- One should **grow a Scrum Team in an incremental, piecemeal fashion**, but eventually the team just becomes too large to remain efficient (應該在增量的、逐個的方式下成長Scrum團隊，但終究團隊會變得太大而無法抱持效率)
- **Small Teams are the most efficient**, but sometimes you need to grow (小團隊最有效率，但有時需要發展變大)
- Differentiate a single large Development Team into two small teams after it gradually grows to the point of inefficiency - **about seven people** in the old team (當一個大的開發團隊漸漸發展到缺乏效率時，將團隊分為兩個小團隊 - 大約七個人留在原來的團隊)
- Members of separate teams should continue to coordinate with each other informally, and as necessary, through the **daily rhythm of Scrum of Scrums events** (不同團隊的成員應該繼續和每個人非正式地，在必要的情況下，在每天Scrum of Scrum會議的節奏中協調/協作)

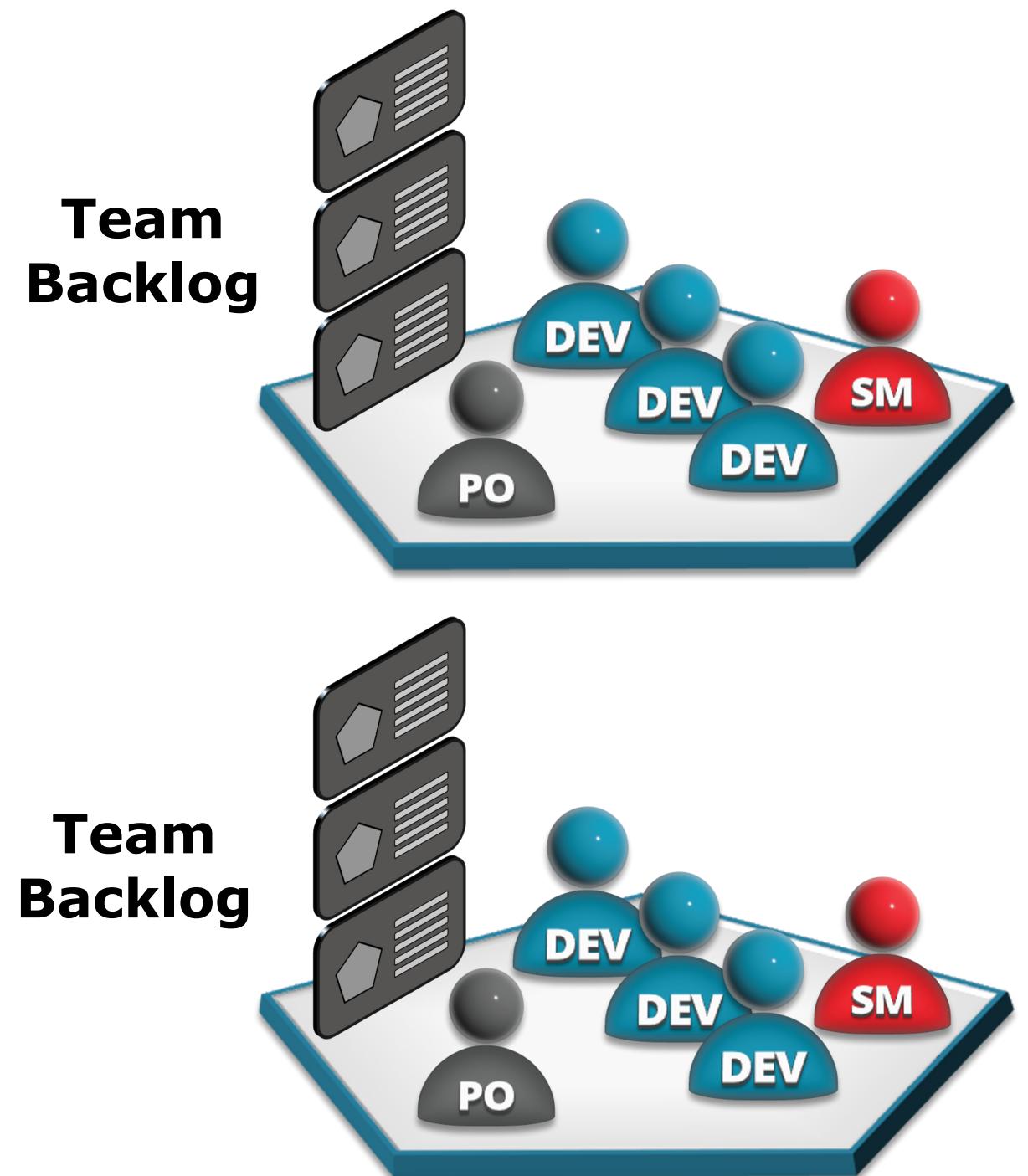


Delivery with Scrum@Scale

用大規模化多團隊Scrum來交付

What is a common issue that arises when teams work independently of one another?

當多團隊各自獨立作業時，通常會出現什麼樣的挑戰？



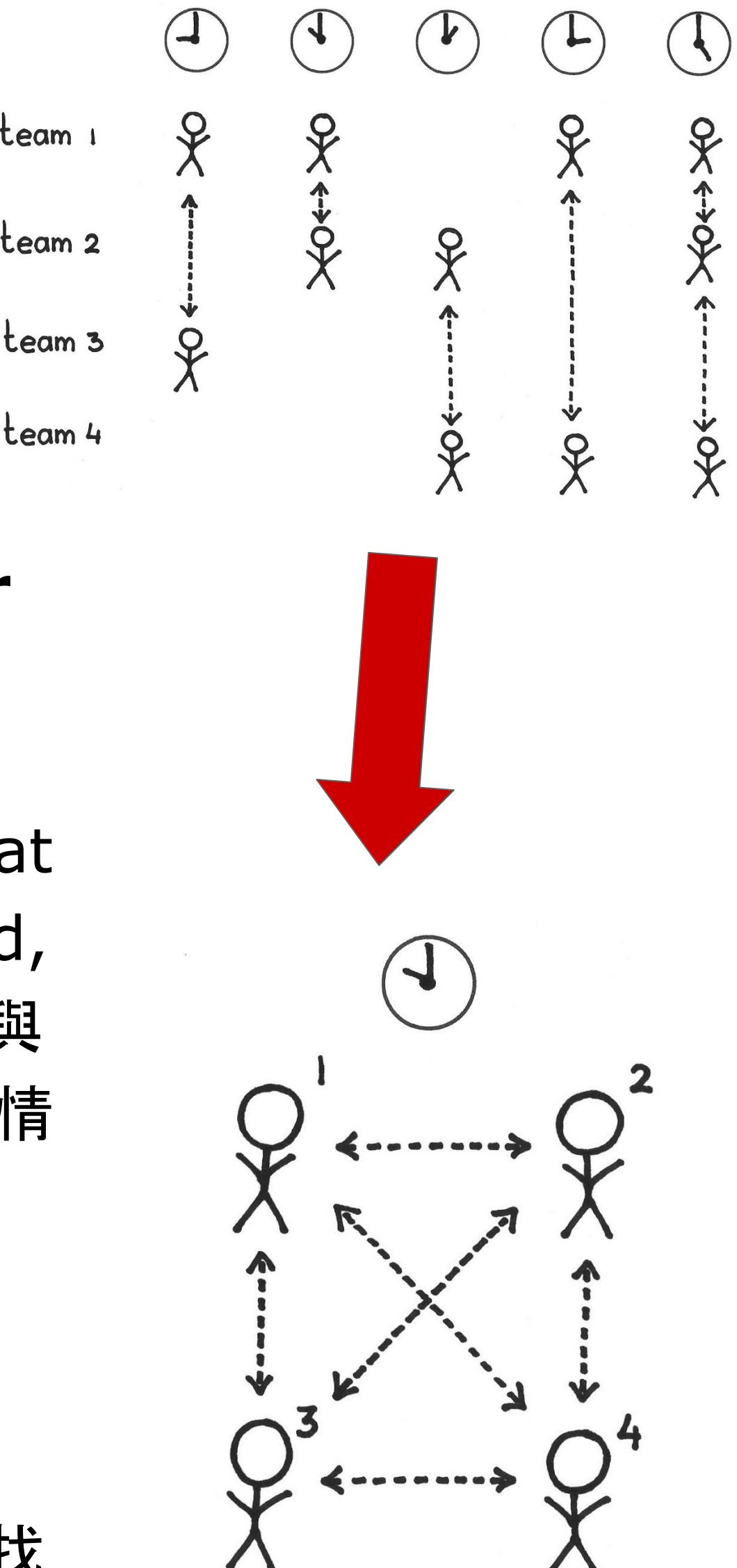
**How do we fix this?
如何解決/改善？**

Scrum of Scrums:

Grouping Teams that NEED to work together

將需要一起工作的團隊組合在一起

- a Scrum Team is working on a single product with multiple Development Teams. The Development **Teams need to coordinate dependencies and shared work.** 一個Scrum團隊擁許多個開發團隊在做一個產品，開發團隊需要協調之間的相依性及分工
- When multiple teams work independently of each other they **tend to focus on their own concerns** and lose sight of any common goals. 當多團隊各自獨立工作時，他們多半會專注自己所在乎的、忽略共同的目標
- Organizations might revert to a command-and-control approach in the false belief that agility only works at the scale of one team, but complexity has grown, not diminished, in this circumstance. Hierarchical control increases delays and ... 組織可能會走回命令與控制的方式，誤以為敏捷只能用在一個團隊的大規模化，但是錯綜複雜已經發生，而不是消除，在此情況下，層層階級的管控會增加延誤及 ...
- Therefore: **Give the right and the responsibility to collaborate on delivering common goals** identified by the Product Owner to the Development Teams themselves. **Permit the teams to figure out the best way to coordinate their efforts.** 因此，給予權力及責任來為PO及開發團隊自己設定的共同目標而合作，准許多團隊自己找出能協調彼此工作的最佳方法

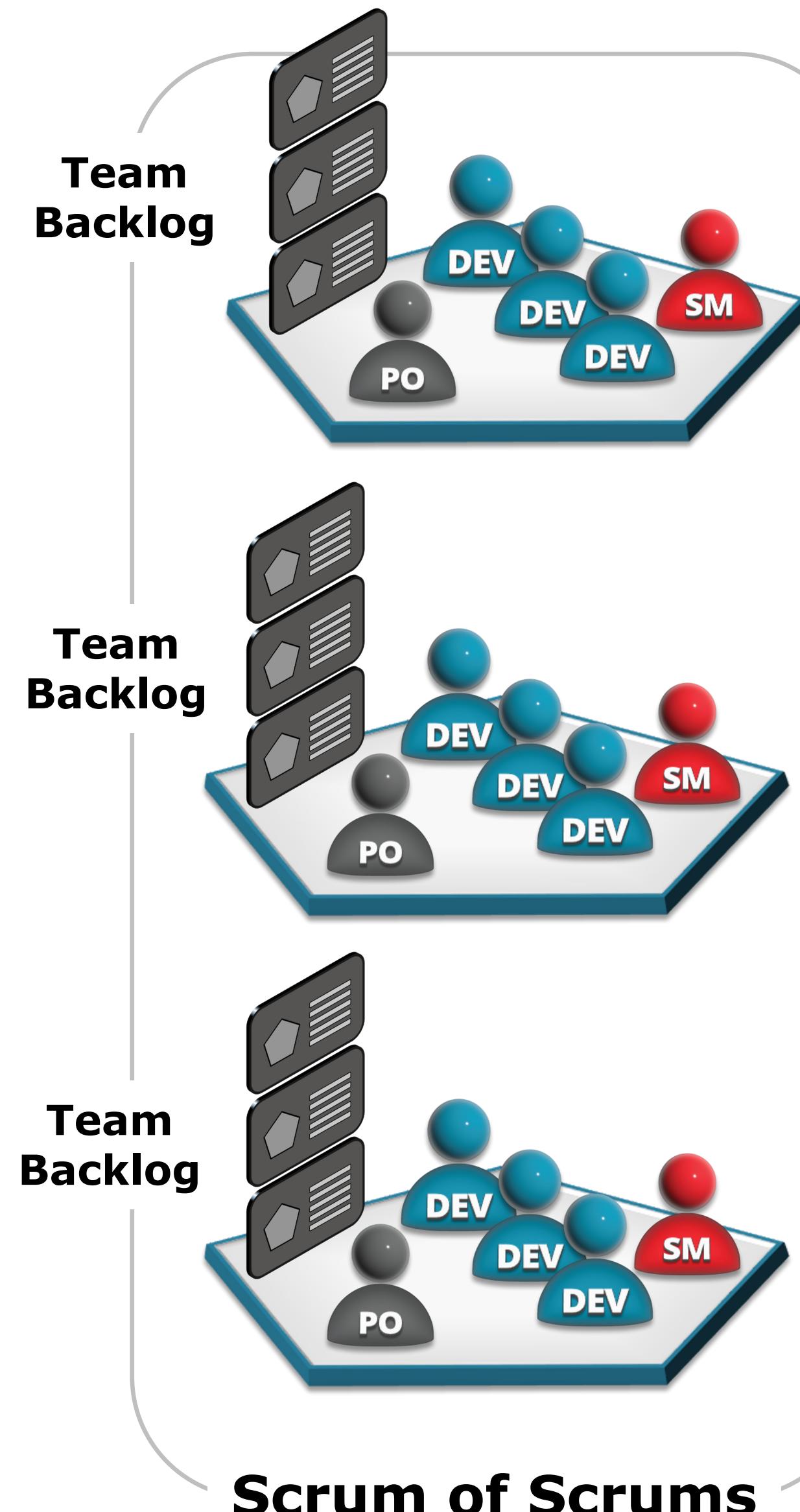


Delivery with Scrum@Scale

用大規模化多團隊Scrum來交付

How do we coordinate WHAT is in each Team Backlog?
我們要如何協調彼此團隊待辦中的什麼呢？

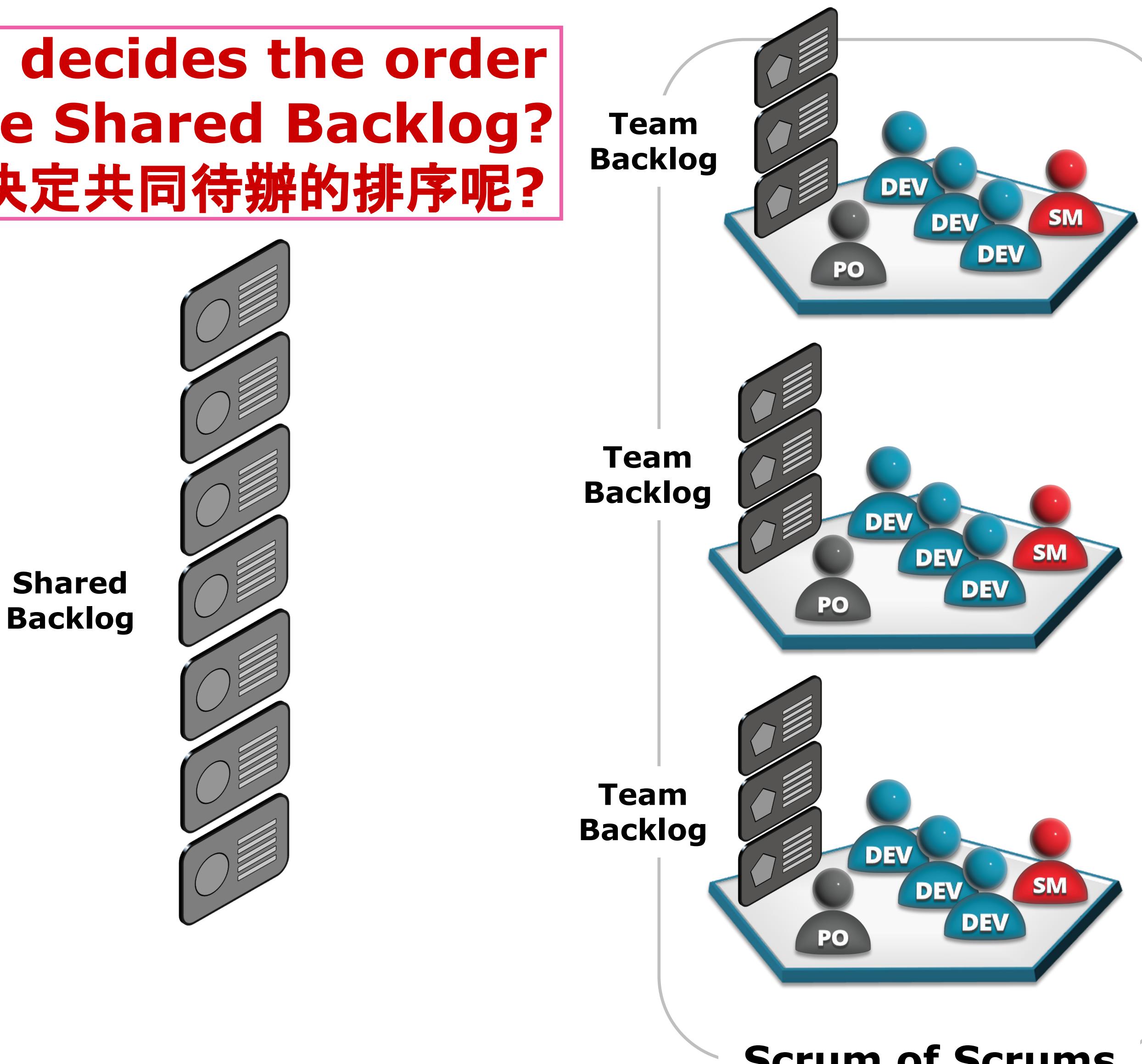
How do we prevent the duplication of effort?
我們要如何避免重複相同的工作呢？



Delivery with Scrum@Scale

用大規模化多團隊Scrum來交付

**Who decides the order
of the Shared Backlog?**
誰來決定共同待辦的排序呢？



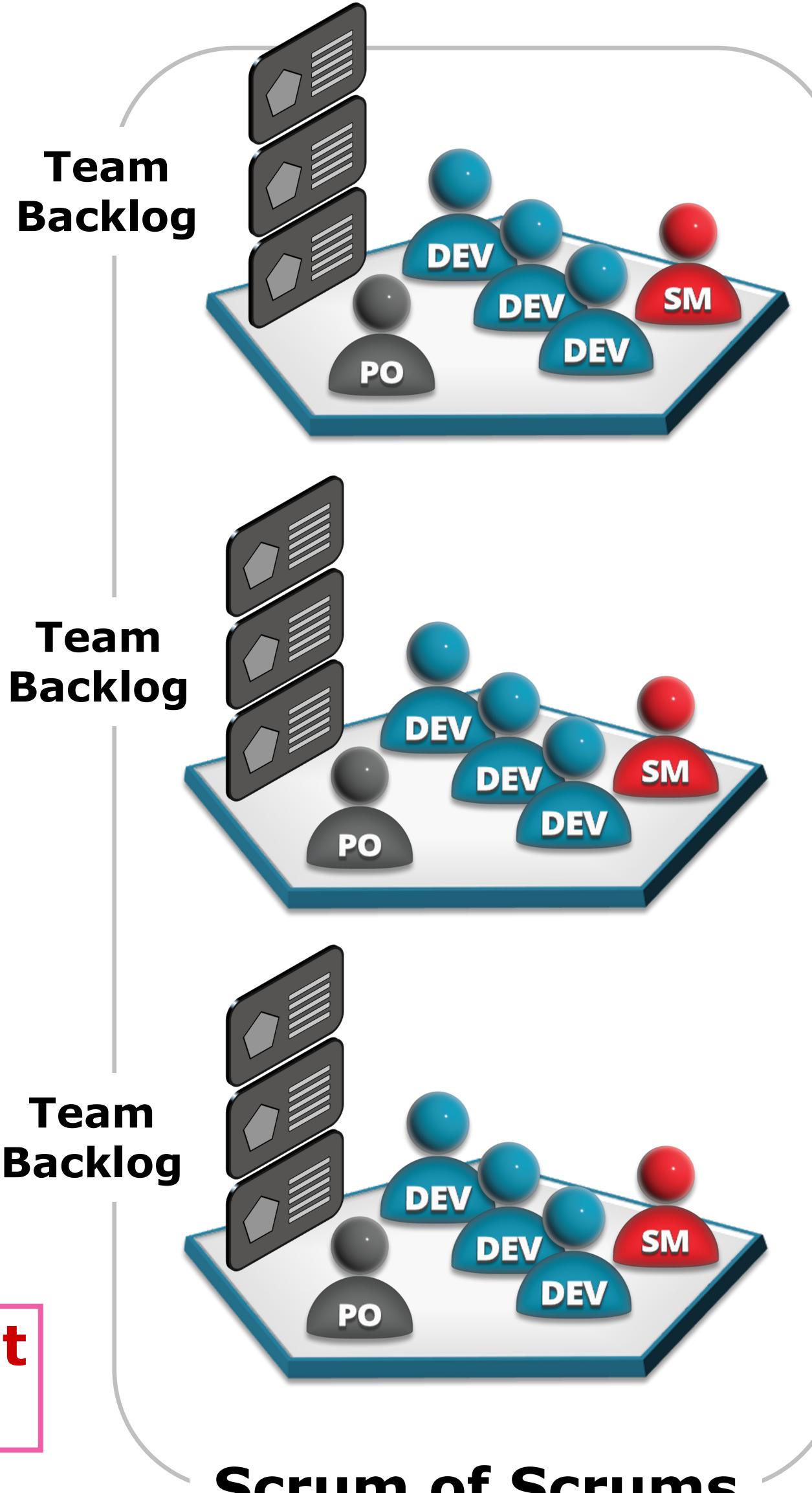
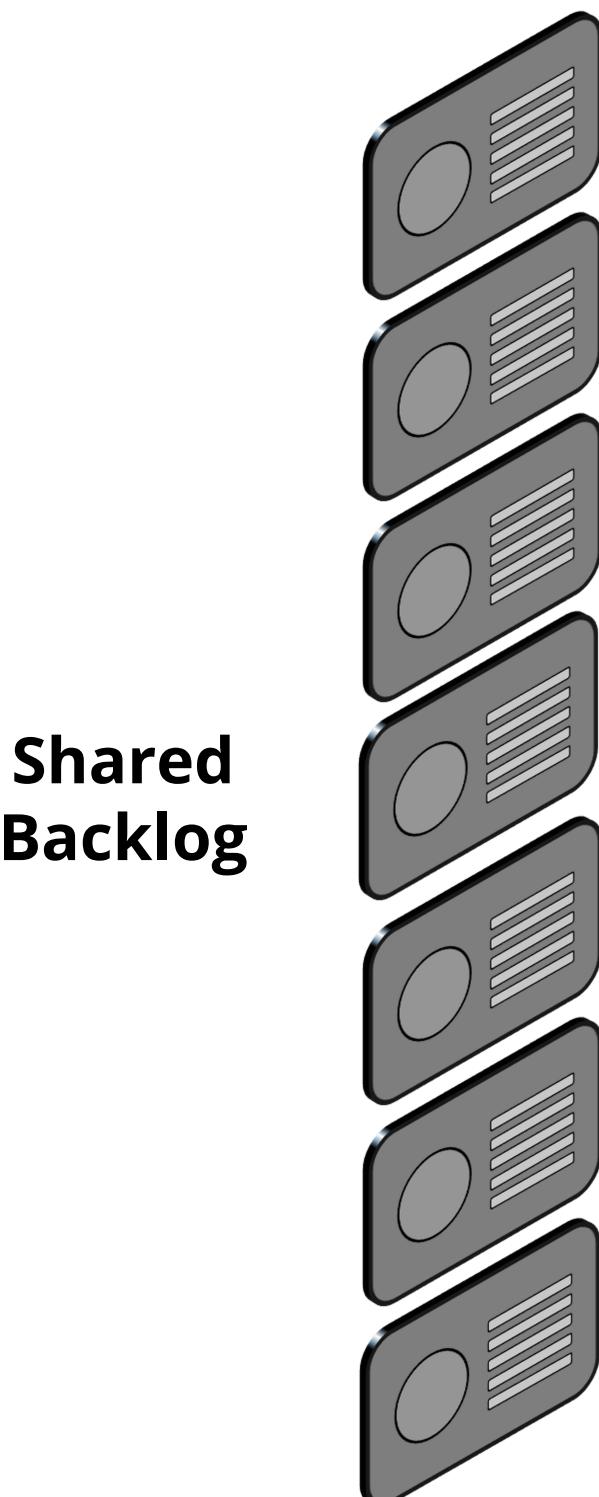
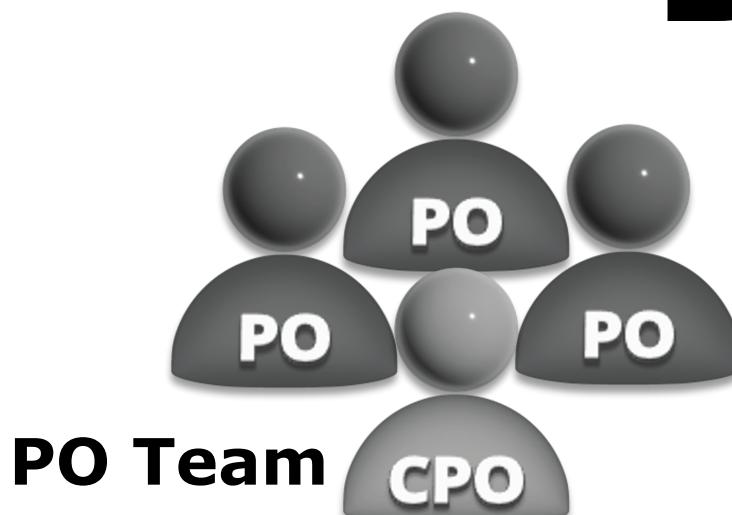
Coordinating 'WHAT' to Deliver 協調該交付什麼： Product Owner Team

- A single Product Owner is accountable for **all value** such as **return on investment** and should handle **market analysis, product discovery, stakeholder management, customer feedback**, and most other market-facing work, while also **enabling the team to build the right thing.**
單一的Product Owner要為所有價值例如投資報酬當責，並做市場分析、產品探索、利益相關者管理、顧賀意見回饋及其他市場相關的大部分工作，同時也要讓團隊能開發出對的商品
- **Not getting the Product Backlog Items Ready** may result in the Development Team building the wrong things...it invites the Development Team to work on lower value Product Backlog Items or start working on Product Backlog Items that are not Ready.
沒把產品待辦項目準備的話，會導致產品開發團隊做出錯的東西 ... 會讓產品開發團隊做低價值的產品待辦項目，或開始做還沒準備好的產品待辦項目
- **Create a Product Owner Team, led by the Chief Product Owner**, whose members together carry out product ownership... it is important that there be a Chief Product Owner (CPO) who has final authority over the ordering of the Product Backlog. 建立一個由Chief Product Owner帶領下的Product Owner團隊，成員一起實現產品的所有權 ... 有一位Chief Product Owner(CPO)是重要的事，他對產品待辦有最終的決定權
- The Product Owner Team realizes the Vision **by ordering the Product Backlog Items** through **a single Product Backlog**. 產品的Product Owner團隊藉由對一個商品待辦之待辦項目的排序，實現產品的願景



Delivery with Scrum@Scale

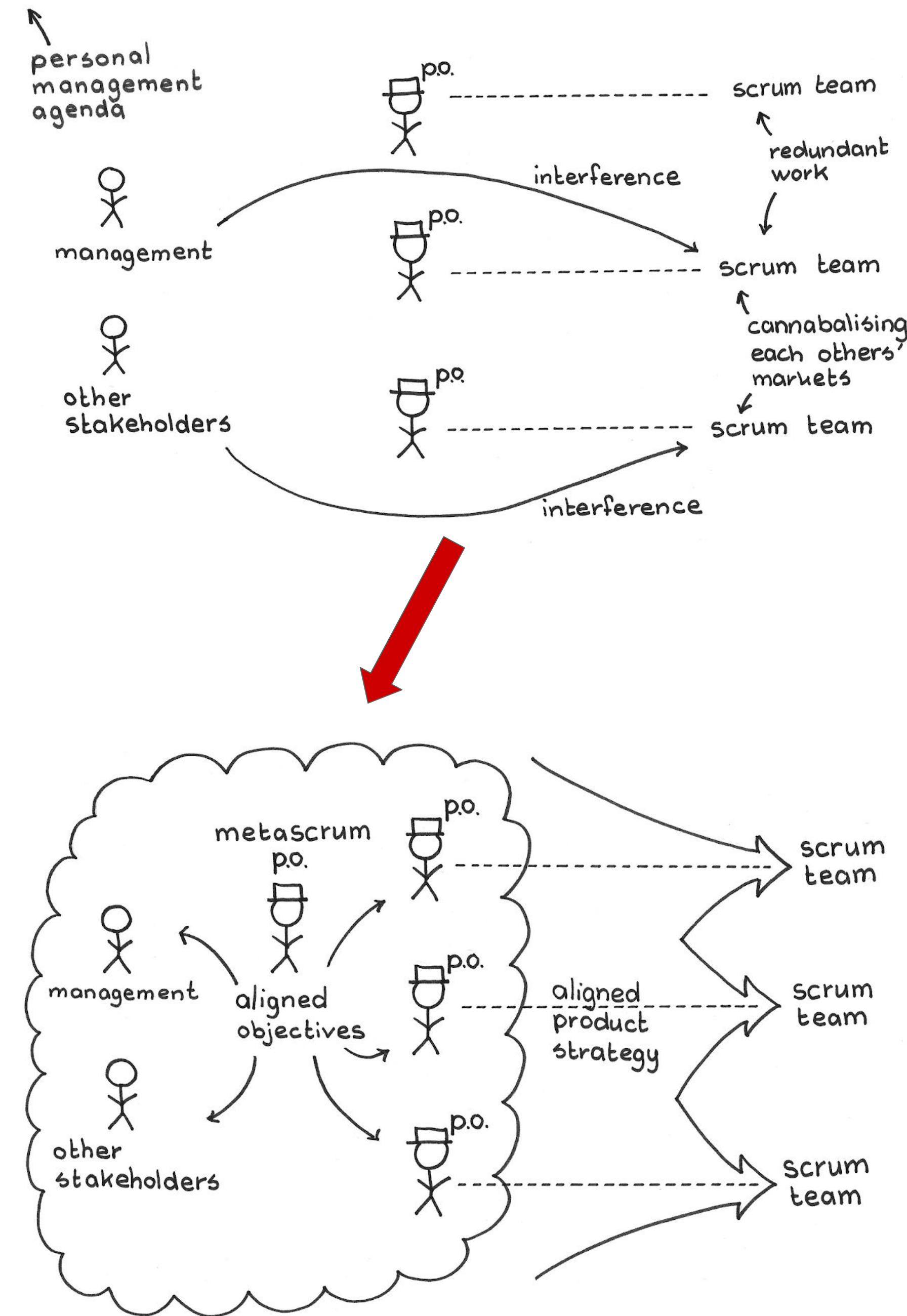
用大規模化多團隊Scrum來交付



**How do we decide what makes it
into the Shared Backlog?**

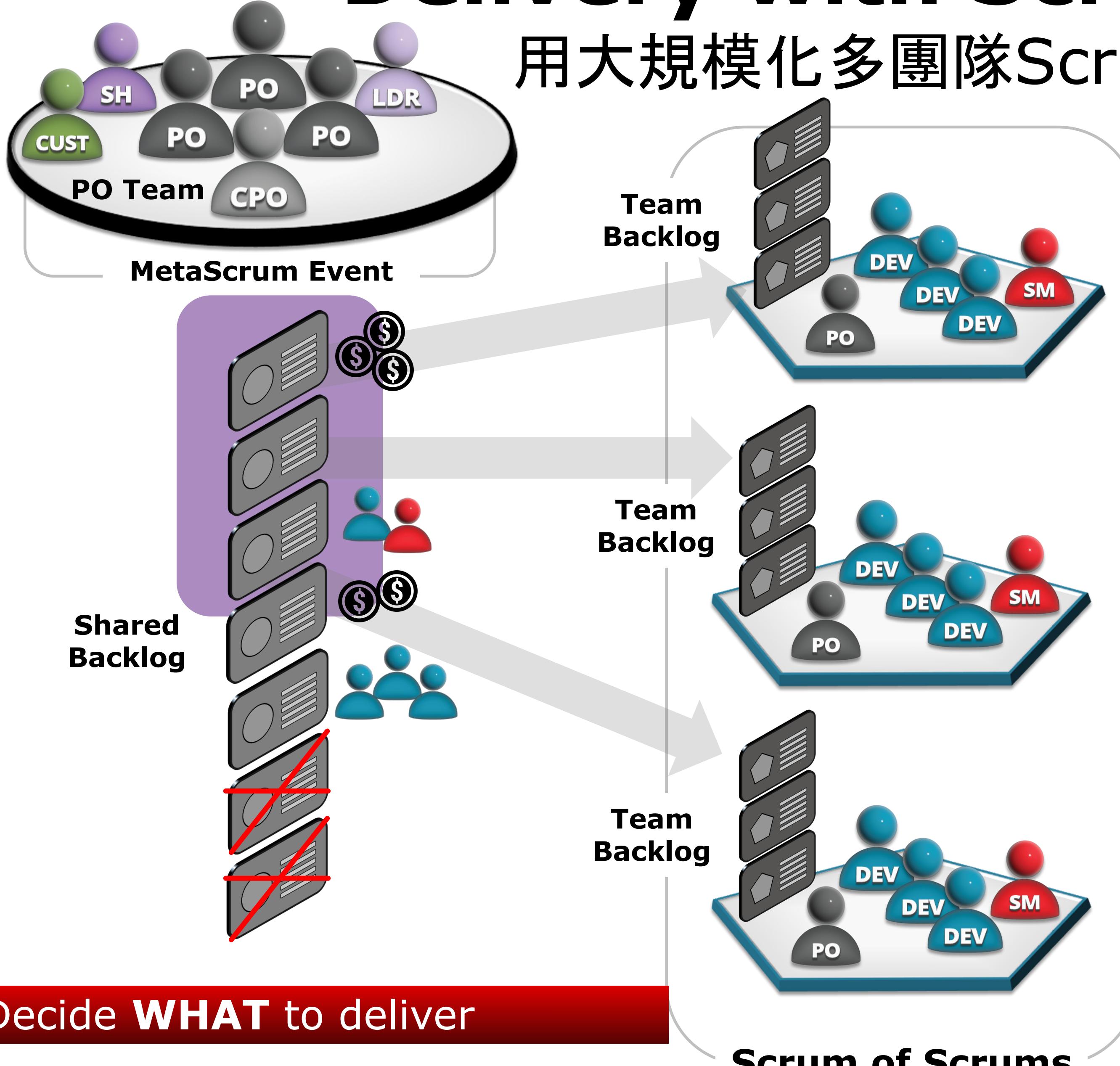
The Agile Steering Body 敏捷委員會： The MetaScrum

- **Scrum Teams are in place, but direction (or the threat of interference) from legacy management structures causes confusion about the locus of control over product content and direction.** 雖然Scrum團隊已就定位, 但來自傳統管理的指導揮(或干預/阻礙的威脅), 導致對產品內容和方向的控制混淆
- **Create a MetaScrum as a forum where the entire enterprise can align** behind the Product Owners' backlogs at every level of Scrum in the organization. 組成一個MetaScrum當作是一個公開討論的地方, 讓整個企業能在Product Owner的待辦之外也密切合作,
- **The MetaScrum is not a body through which management controls product direction, but rather a forum for coordination between products and management.** MetaScrum不是管理階層用來管控產品方向的, 而是用來討論產品與管理階層之間的協調

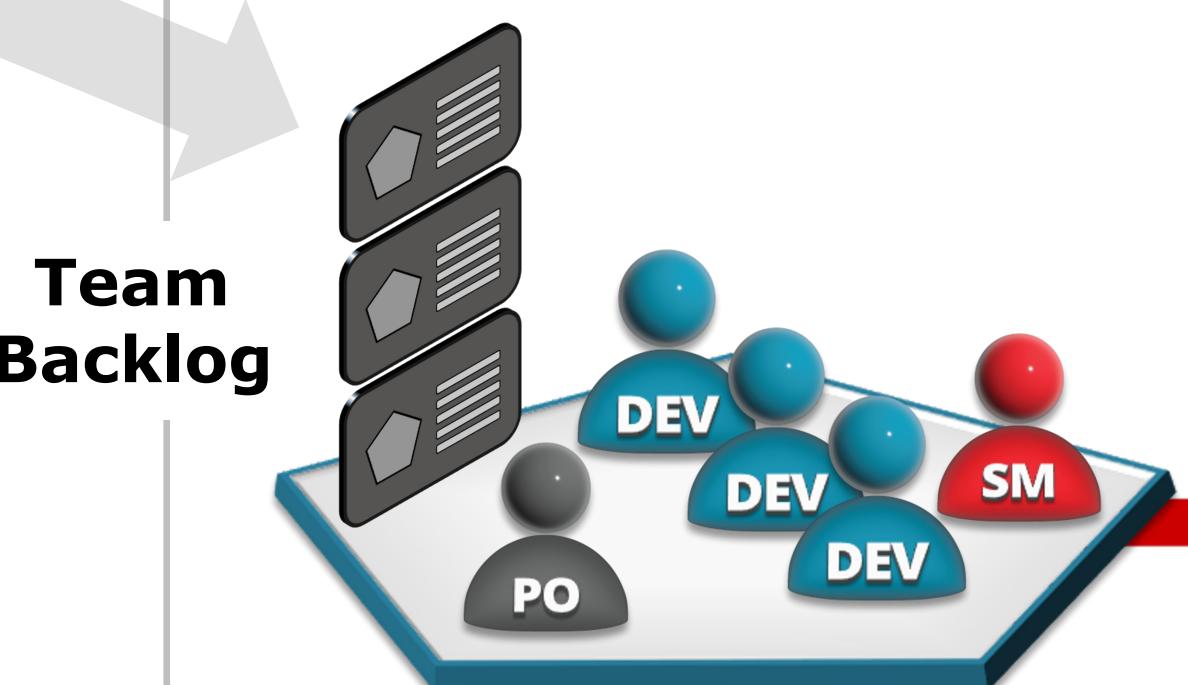
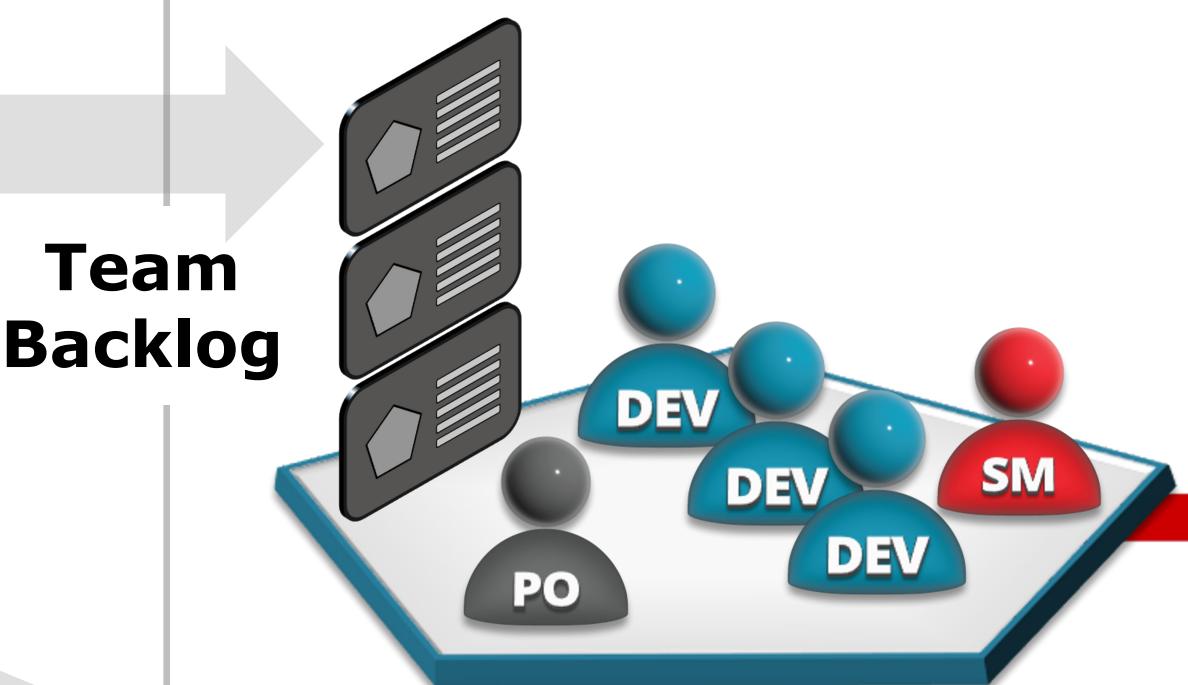
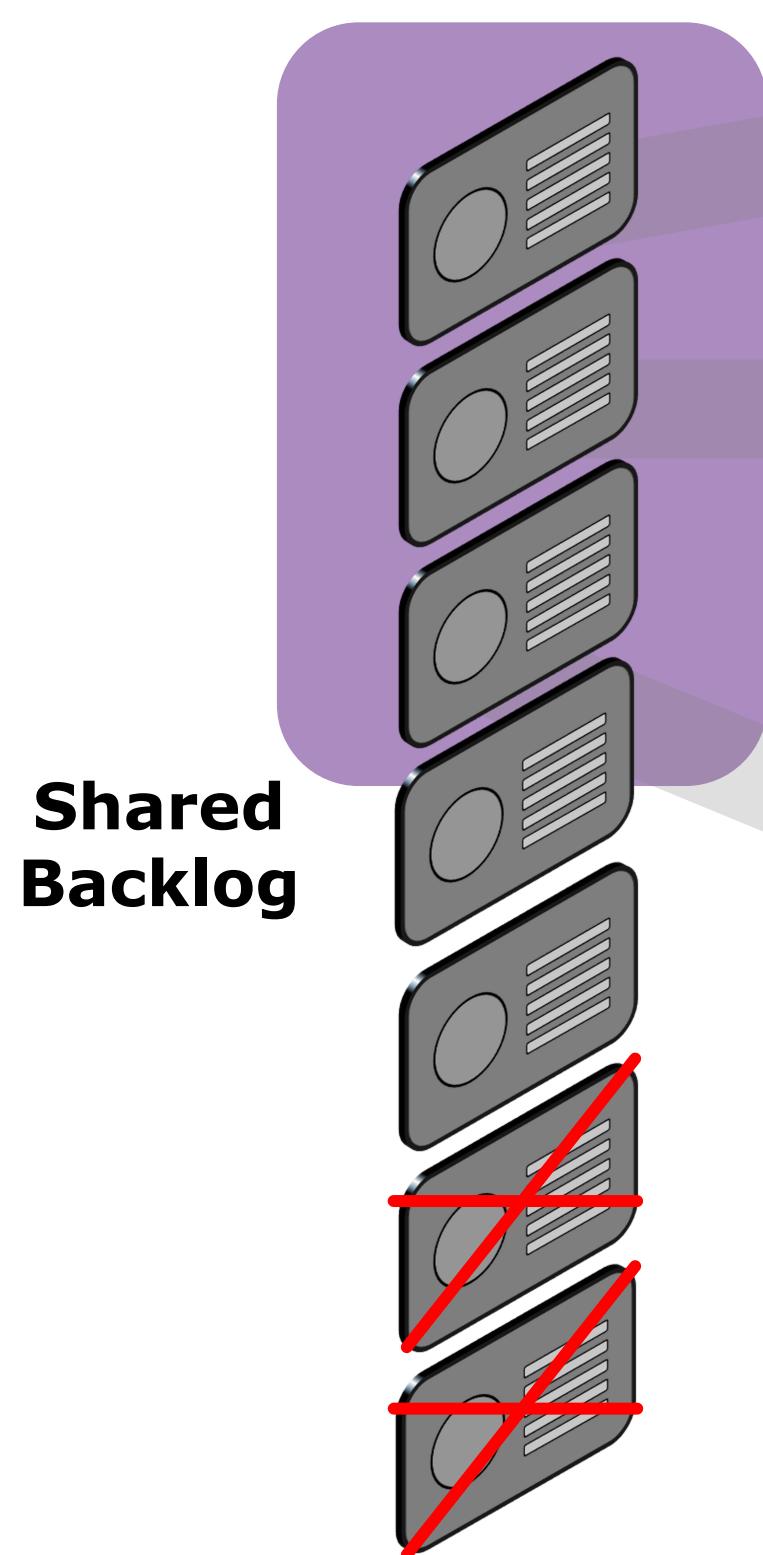
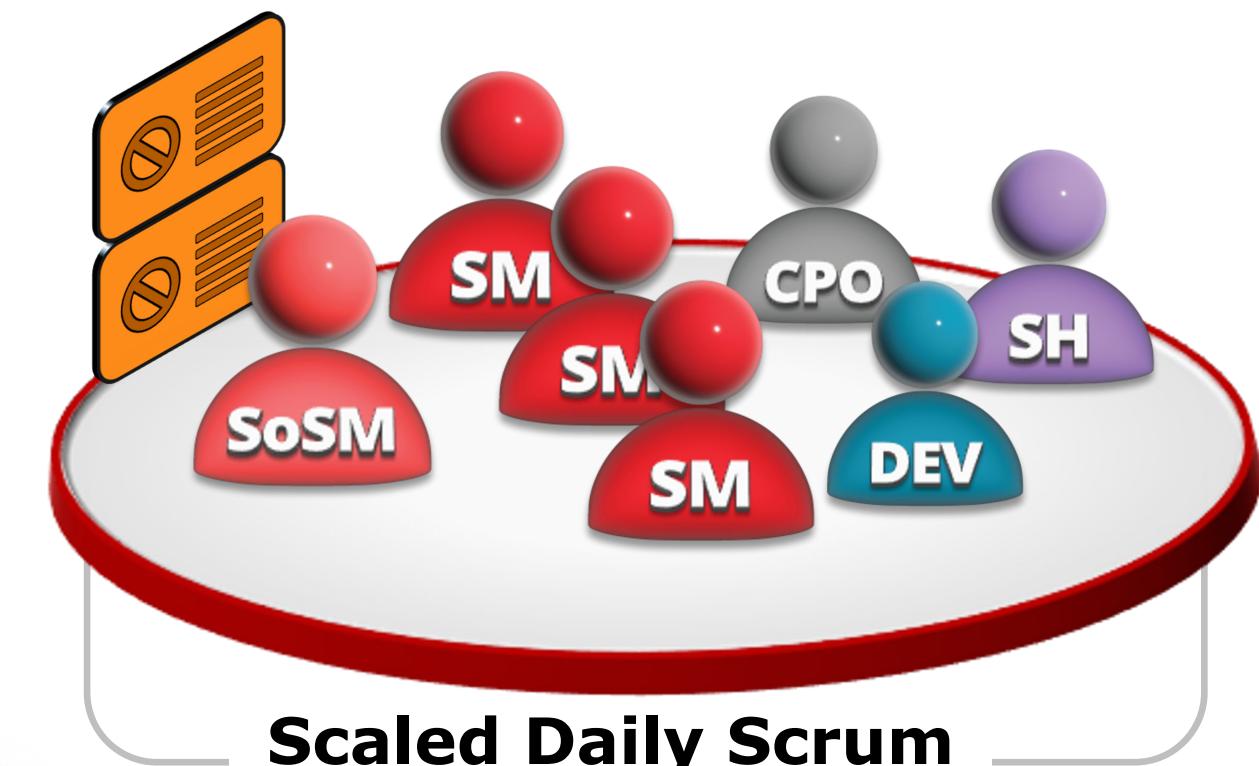
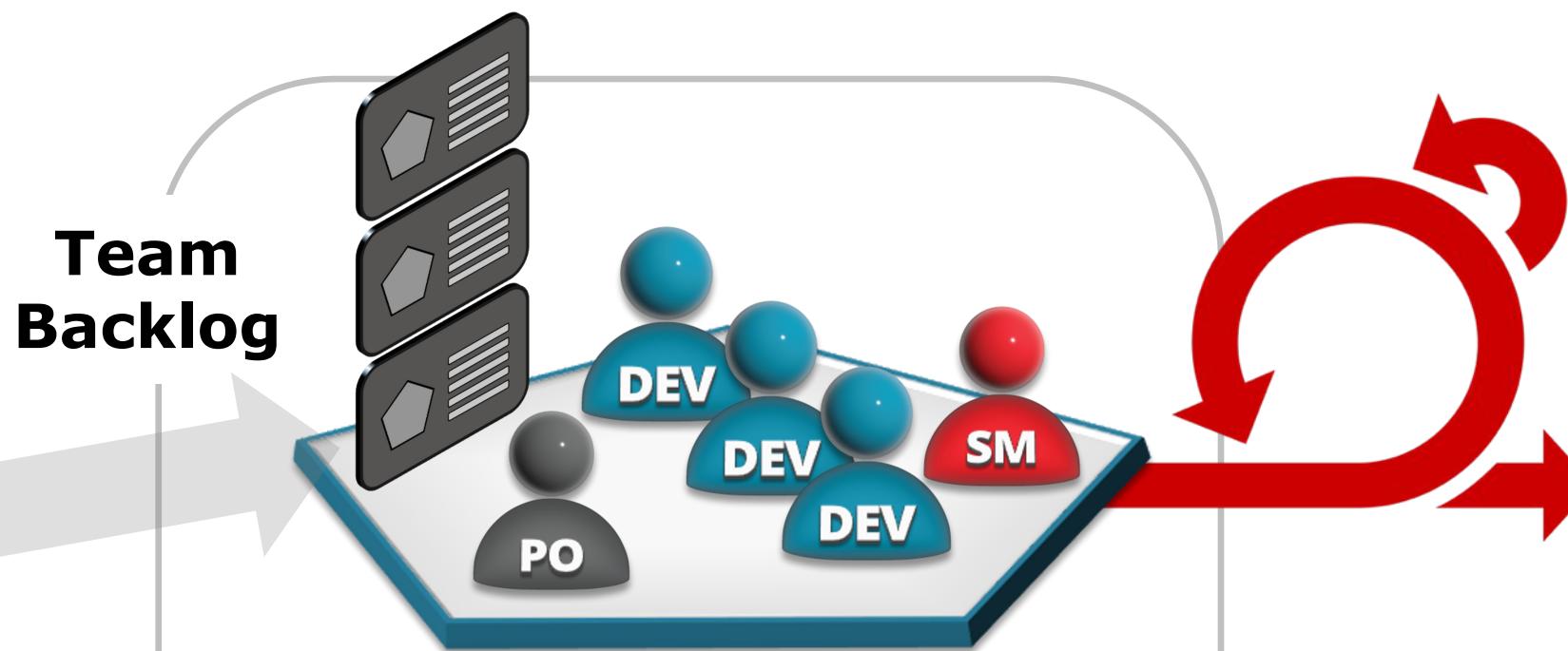
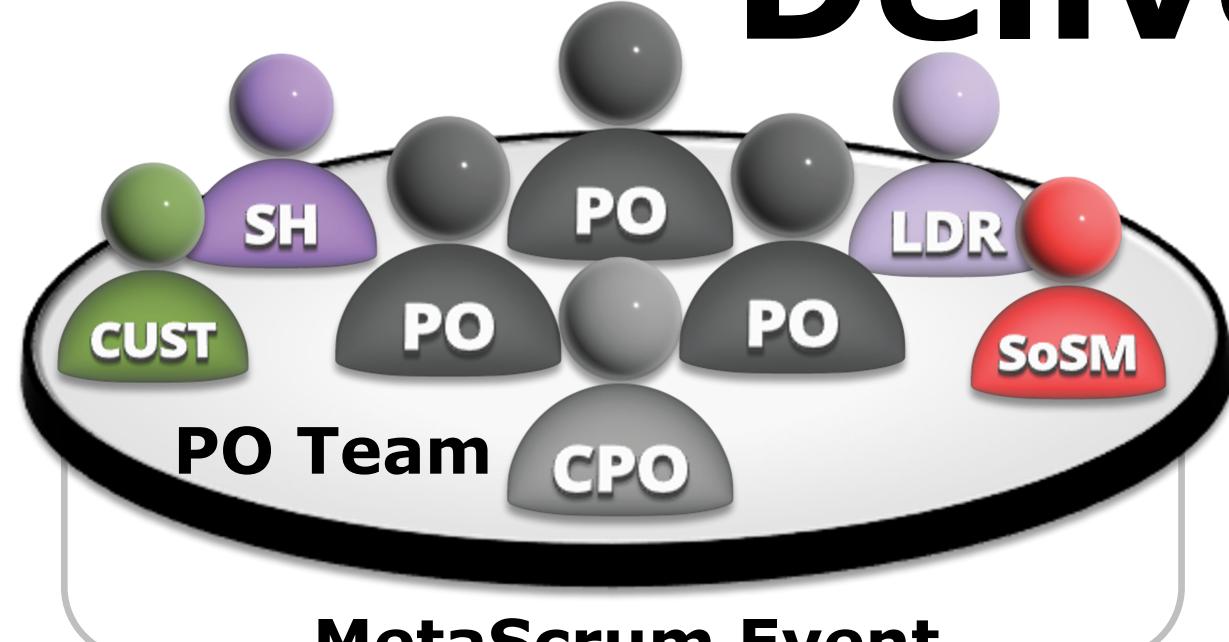


Delivery with Scrum@Scale

用大規模化多團隊Scrum來交付



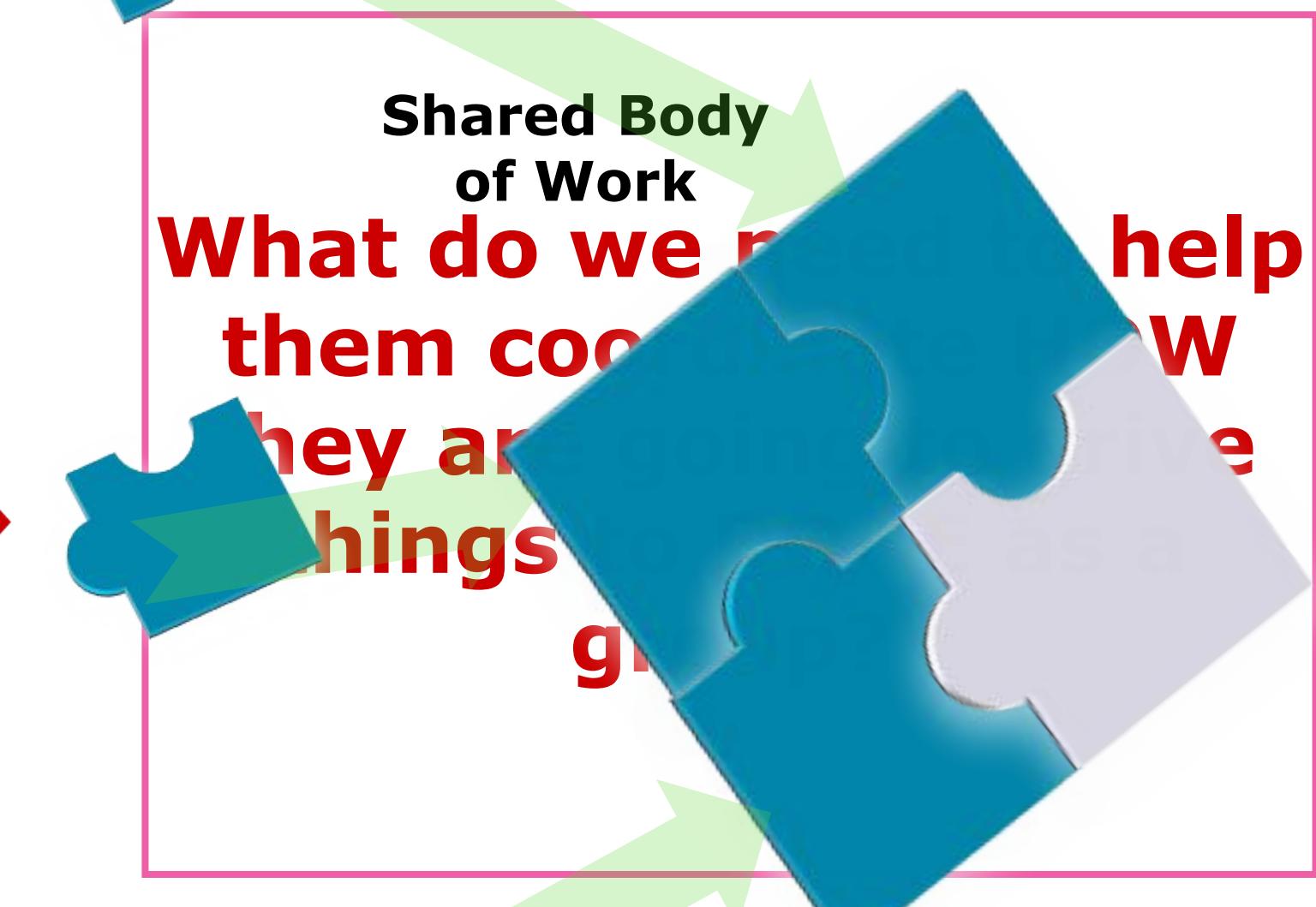
Delivery with Scrum@Scale



Scrum of Scrums

Decide **WHAT** to deliver

Coordinate **HOW** to deliver



Introducing the Scrum@Scale Guide

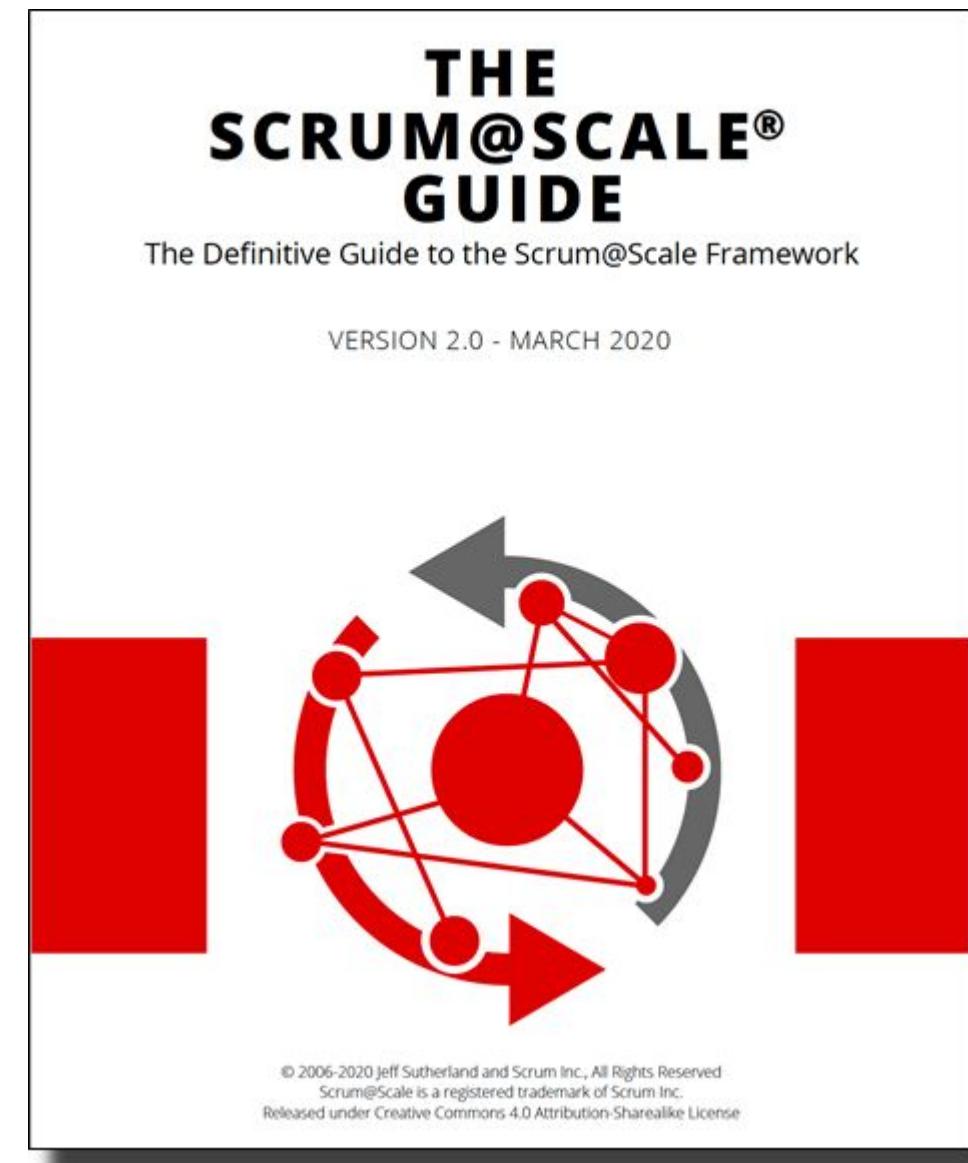
為何組織很難做到敏捷？

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

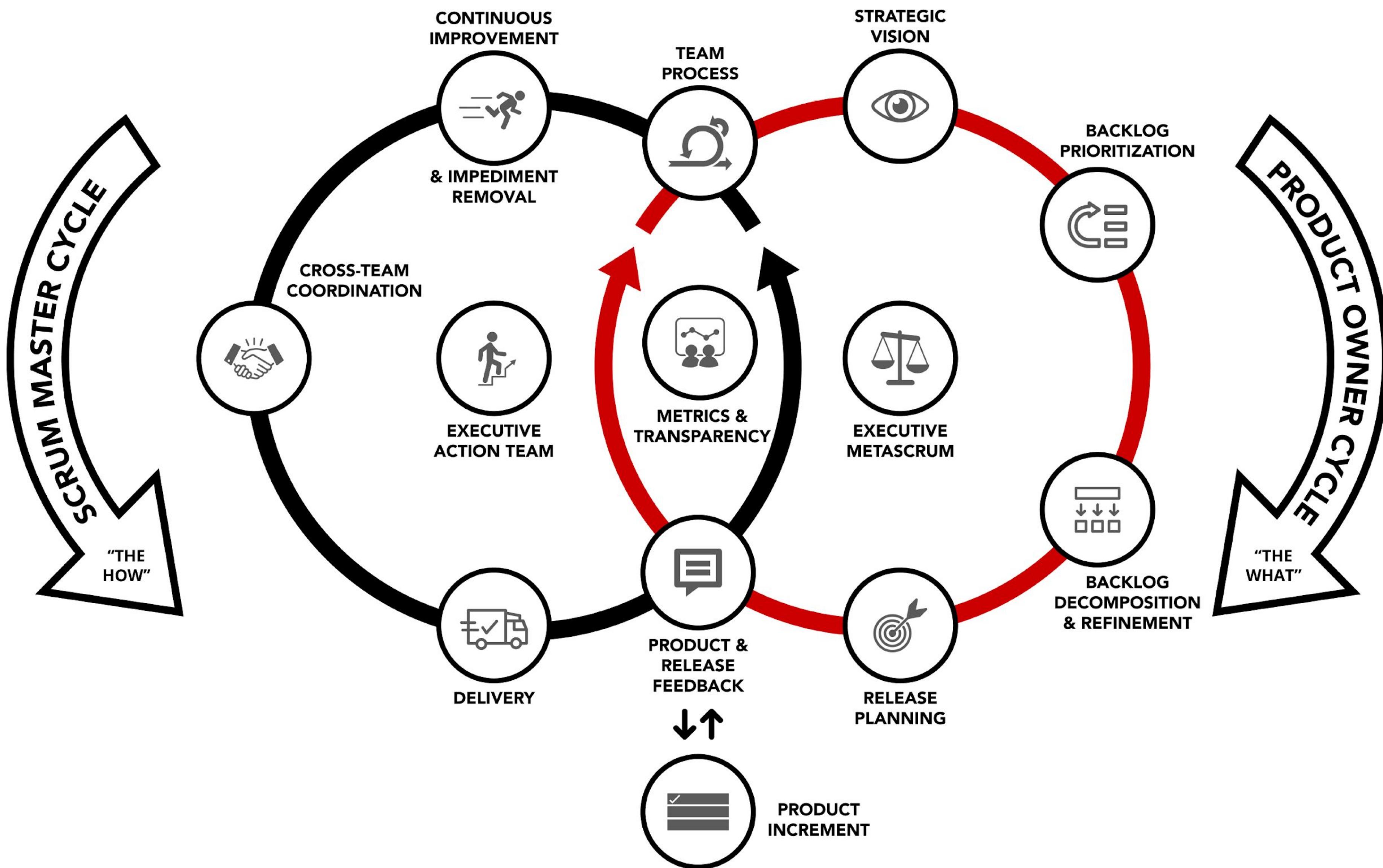
November 2020



Addresses the Four Mega-Issues 解決四大問題

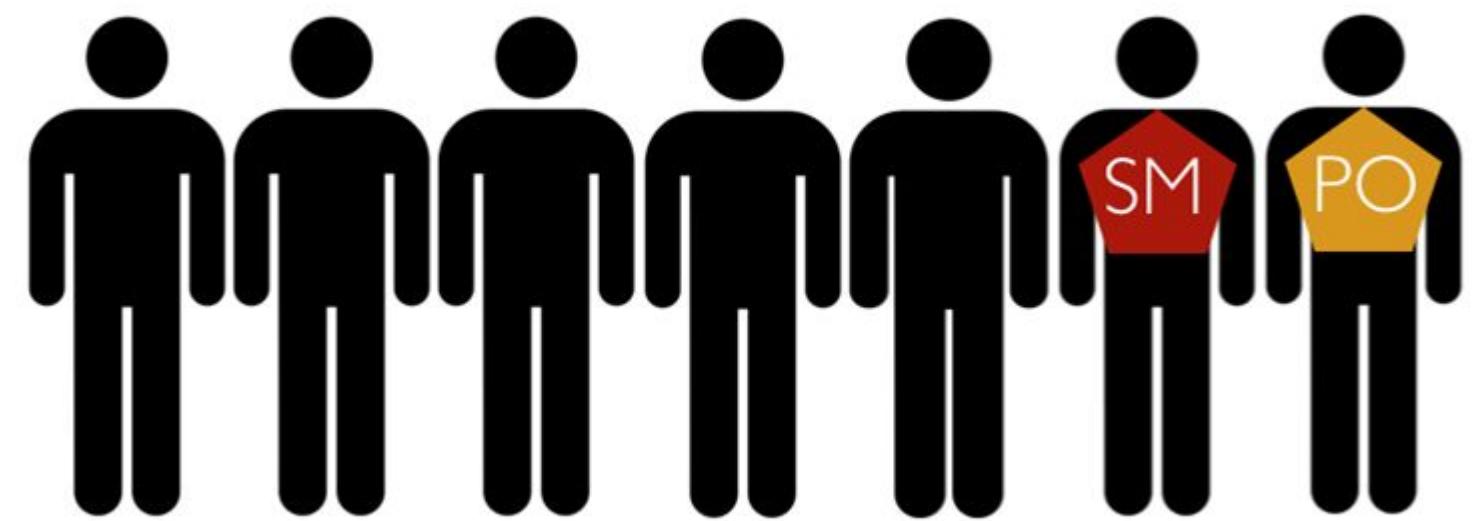
- #1 Prioritize:** Limited resources, ability to **focus**
優先排序: 資源有限，專注的能力
- #2 Deliver:** High-quality, **working product/service**
is primary measure of progress
交付:進步的主要評估在於高品質、有用的產品/服務
- #3 Refactor:** Ability to change fast – both **Product** and **Organization**
重構:產品和組織快速變革的能力
- #4 Organizational Culture:** Value Driven, Agility and Speed
組織文化: 價值導向、敏捷和速度

Scrum@Scale Framework

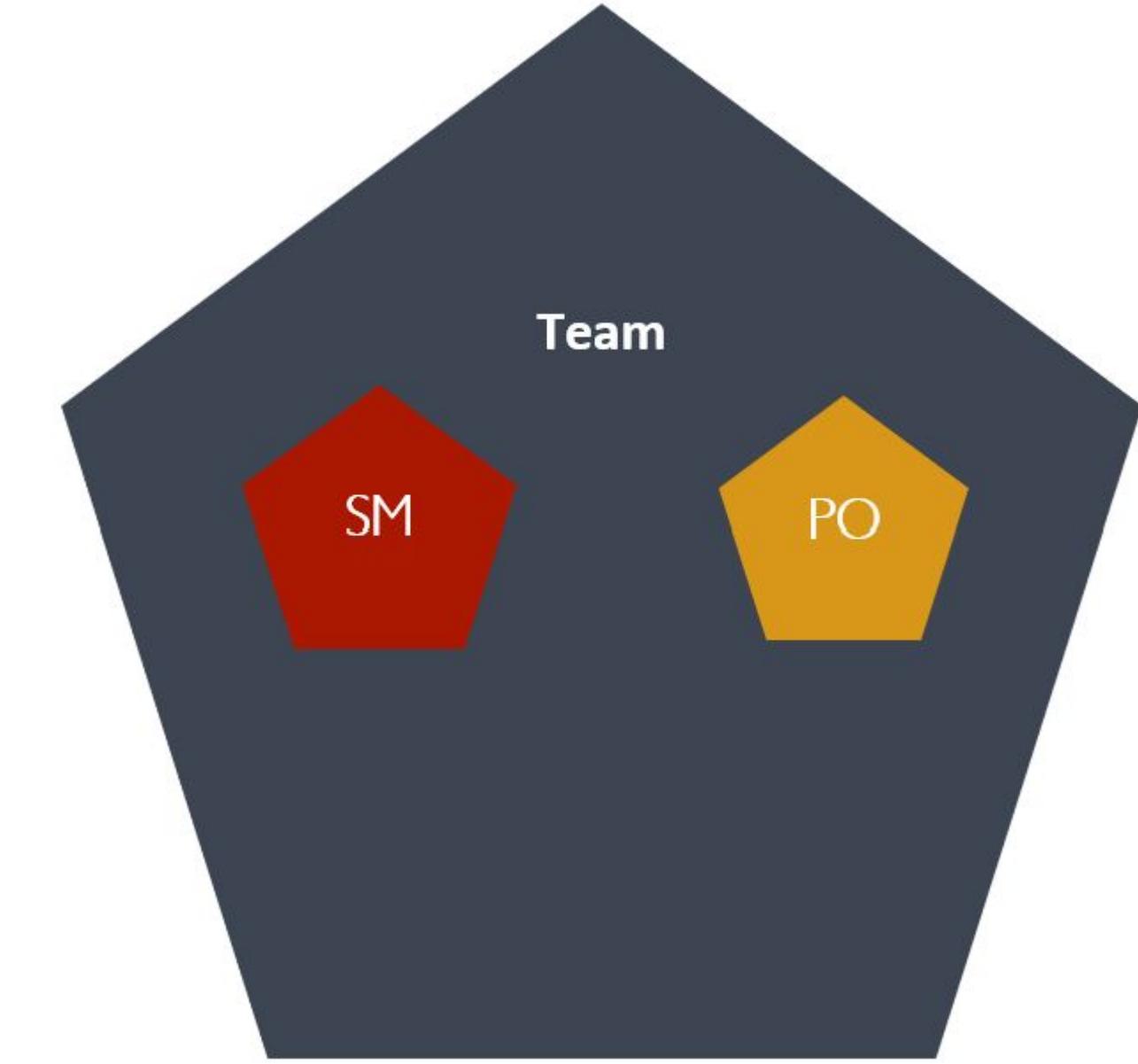


A Cross-functional Scrum Team

跨職能Scrum團隊



=



- 10 or fewer people (ideal size = 4 or 5)
10人或更少(理想人數 4或5)
- **SM** - Removes Impediments & **Keeps True Scrum**
- **PO** - **Team's Known Stable Interface** &
Translates **Vision into Product Backlog**
團隊的穩定界面, 將願景轉化帶入為產品待辦

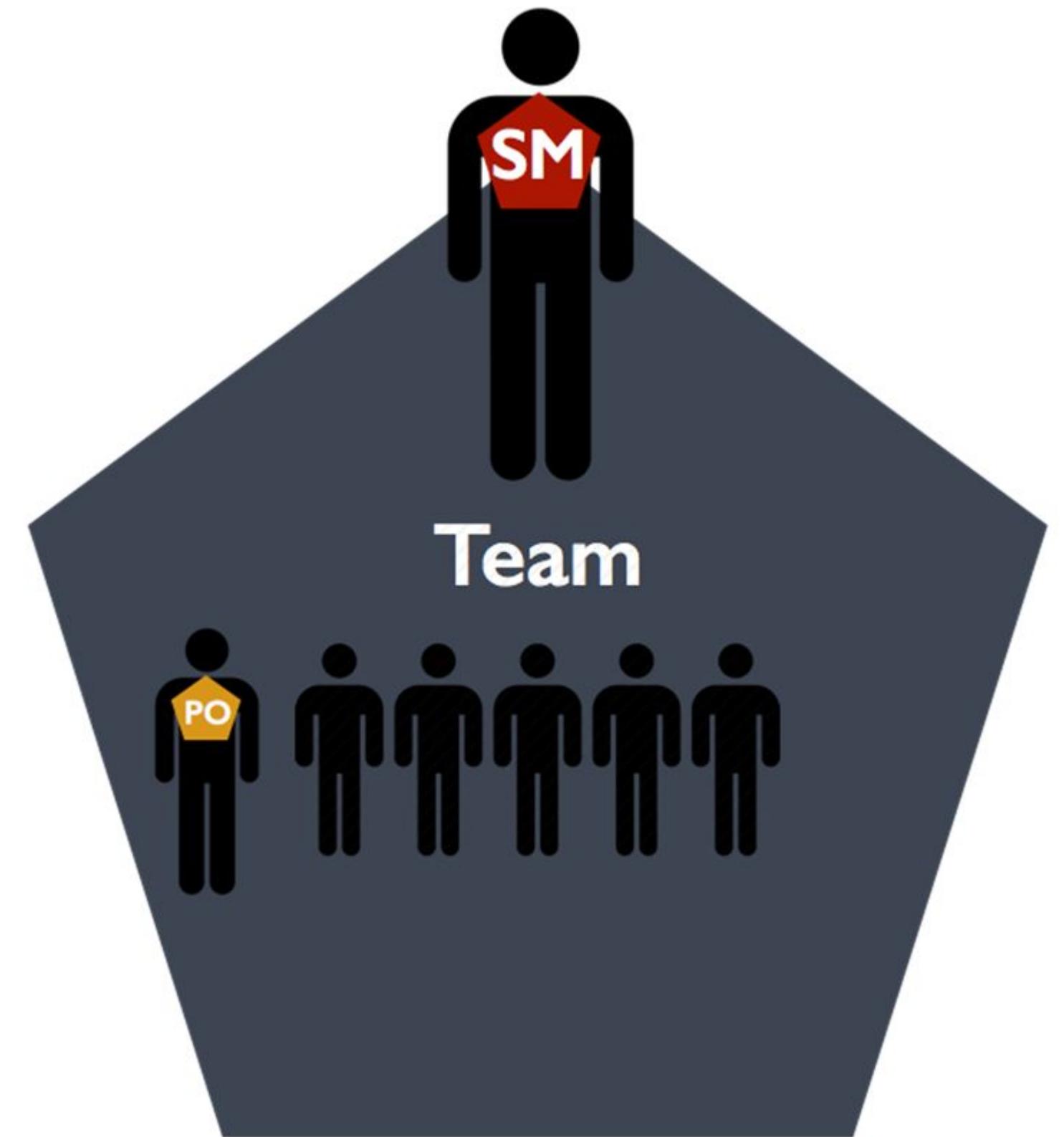
- Transcendent Goals 偉大的目標
- **Cross Functional** 跨職能 ->
Independent Path to Production
- **Self-Managing** 自我管理
- **Stable Teams** 穩定的團隊
- **Accountable to PO** 當責

a pentagon = a team
五邊形圖代表一個Scrum團隊

The Scrum Master from the Team Perspective

站在團隊觀點的Scrum Master

- **A leader who serves** 領導者
- Ensures **work & impediments** are made **visible**
確保工作和障礙都能被看見
- **Protects** the Scrum Team from **interruptions**
保護Scrum團隊免受干擾
- **Coaches** the Scrum Team in Scrum, Lean & Agile
- **Facilitates** Scrum events
- Helps create **process improvements** and facilitate **impediment removal**
幫助流程改進和消除障礙
- Implements the **Scrum & Agile values**
實現Scrum和敏捷價值
- Promotes **Kaizen thinking & waste reduction**
促進思考改善和減少浪費
- Helps to **maintain external radiators** of team progress
協助讓外部持續知道團隊的進度

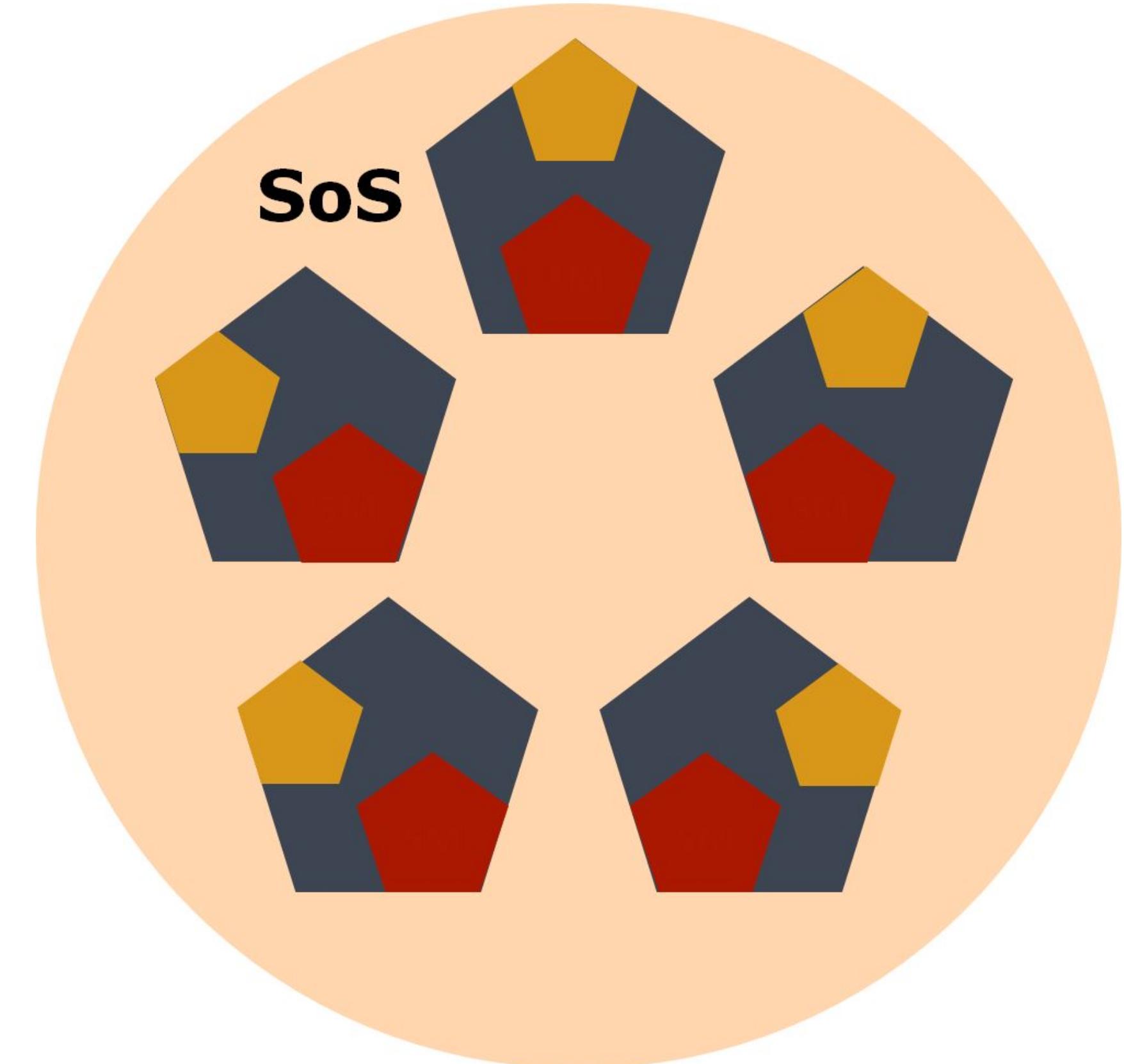


The Scrum of Scrums Perspective

Scrum團隊群(SoS)的觀點

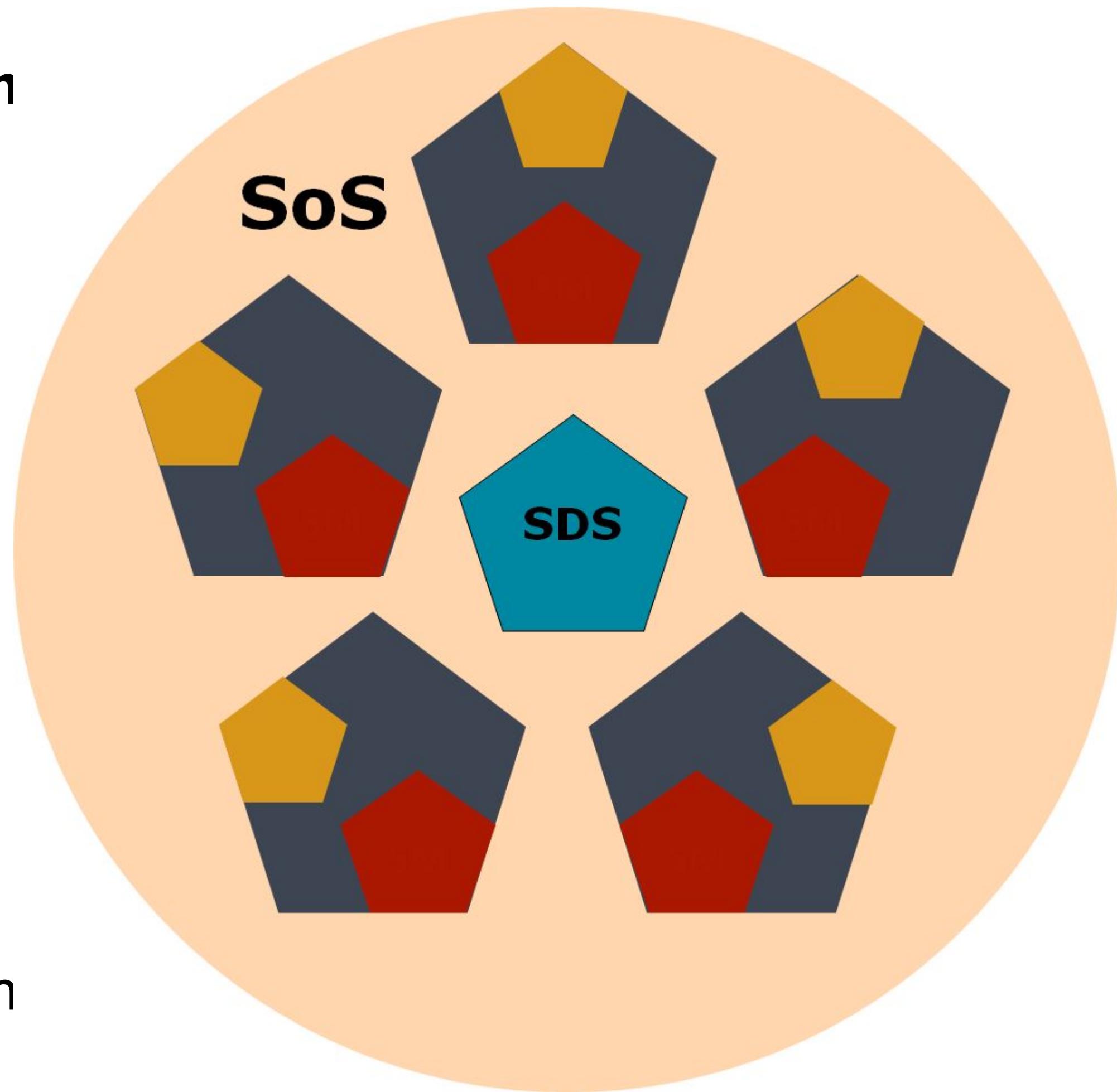
The Scrum of Scrums (SoS) is:

- **Not an event**
不是一個事件
- A set of **teams** that have **a need to coordinate** to deliver together
一群團隊需要協調一起交付
- **A “Team of Teams”**
團隊的團隊
- **Responsible for a shippable increment of product** at the end of a Sprint
and must **deliver the Chief Product Owner's Sprint Goal**
在Sprint結束時負責產品的交付，並且必須與首席PO密切合作，以便達成Sprint目標



Scaled Daily Scrum (SDS) 多團隊每日例會

- This is an event facilitated by a **Scrum of Scrum Master (the SoSM)**
這是由首席SM引導的活動
- Enables **Cross-Team Coordination**
促使跨團隊協調
- The SDS event **mirrors the Daily Scrum** in that it:
如同Daily Scrum
 - is an opportunity to **re-plan as a group** in order to achieve the **Sprint Goal for the SoS**
重新規劃以實現SoS的Sprint目標
 - **surfaces & removes impediments** if possible
讓障礙浮出台面並移除
 - Shares **learning** for Continuous Improvement
分享學習以不斷改進



Example: 5 teams delivering products together

Impediment Template Example

阻礙樣板

Who raised the impediment / Team / Product Area 誰提出阻礙/哪個團隊/產品哪部分	Date Raised 提出的日期
<ol style="list-style-type: none">1. What are you trying to do? 你要做什麼?2. What is the impediment? 遇到什麼障礙?3. What is the impact (of not being able to do this)? (做不到)會產生什麼影響?4. What has been done already? 已經做了什麼?5. What can be done to resolve this (the ideal situation looks like)? 能做什麼來解決?(理想情況下)	

Used in Real Life 實際應用

The screenshot shows a Jira ticket interface. At the top, there's a navigation bar with a logo, the project key 'LAT / SMLA-21', and the title 'Confluence Account doesn't exist for all [REDACTED] IT users'. Below the title are several buttons: Edit, Comment, Assign, More, Ready, Blocked, and Test.

The ticket details are organized into sections:

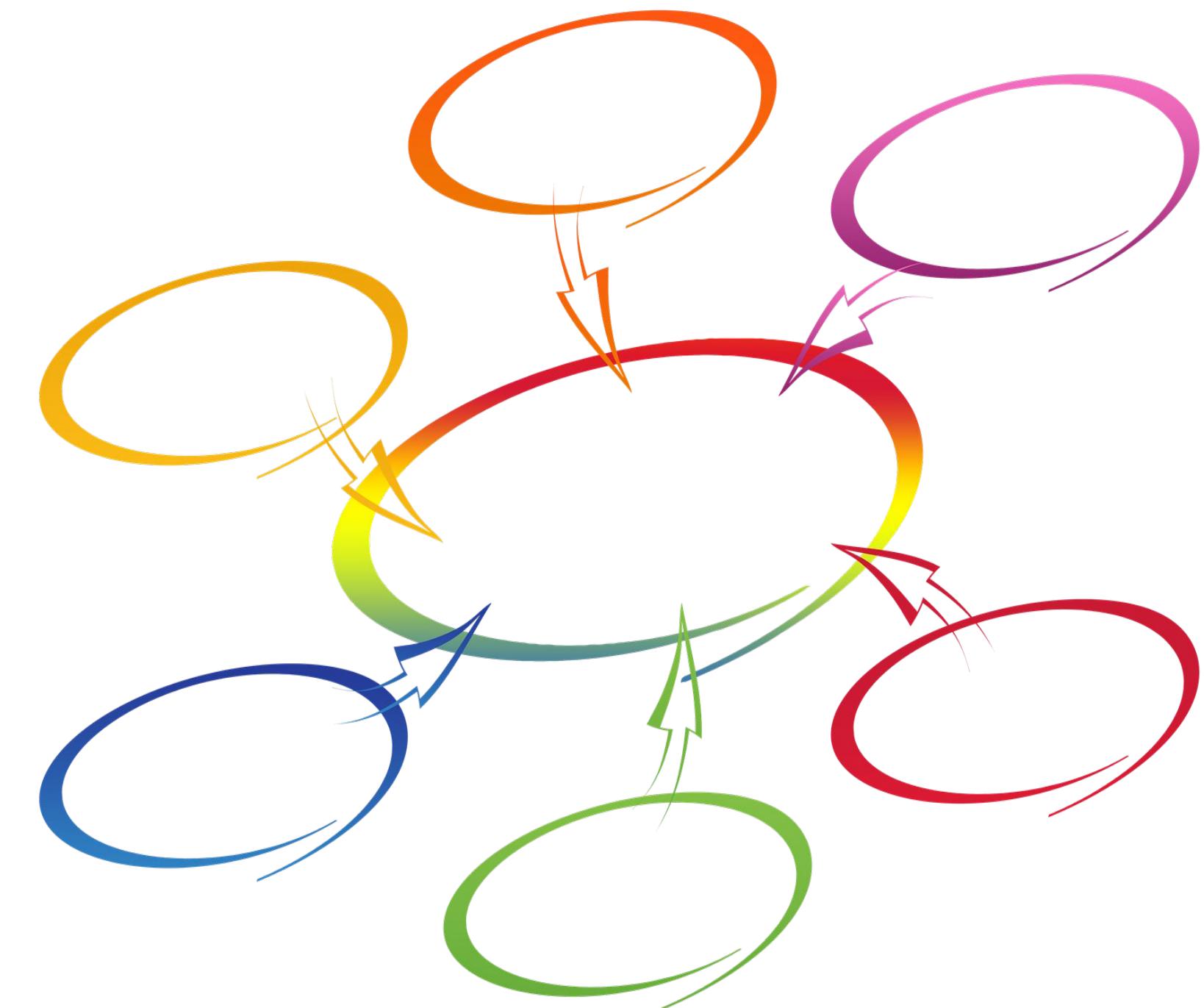
- Details:** Type: Bug, Priority: High, Labels: None. Status: DEV (View Workflow), Resolution: Unresolved.
- Description:** A table with the following rows:

What are you trying to accomplish?	Add all users to IT for [REDACTED] Confluence Space
What is in your way? <i>Please be as specific as possible, including the group(s) you think are in the way.</i>	Some of the users do not have Confluence Account and additionally there are more users who may need access, we do not have a final list.
What is the impact of not being able to do this?	Users will not be able to collaborate on project work without access to the confluence space.
What have you tried already? <i>Please share any documentation.</i>	Nancy has given access to the users who already had confluence account.
What is your vision for an ideal resolution?	1) Ask each lead to provide/confirm the users for their teams. 2) Send an email to all the users to submit request to get the confluence account created by submitting the catalog: https://[REDACTED].service-now.com/ep?id=ep_sc_cat_item&sys_id=211a7ae1db9214906c5edda5ca961957
- People:** Assignee: [REDACTED], Reporter: [REDACTED], Votes: [REDACTED], Watchers: [REDACTED]
- Dates:** Created: [REDACTED], Updated: [REDACTED]
- Agile:** View on Board

At the bottom, there's an 'Attachments' section with a placeholder for file uploads: 'Drop files to attach, or browse.' and a three-dot ellipsis button.

The Scrum of Scrums Master (SoSM) 首席SM

- Must ensure that **impediments** are shared and removed, **knowledge** is spread and standardized, and **dependencies** are discussed and resolved
必須確認障礙被看到及移除、消息統一傳布、相依性被提出討論及解決
- Could be **one of the team SM's** or someone dedicated to the role, example if the company has a separate Operations Department, it could be a senior person from there
可以由其中一個團隊的SM擔任，或是一個專職的人，例如營運部門的資深員工
- It should be someone who has enough power to **remove impediments** and understands the policies and politics of **getting things Done** and Released
應該是有足夠權利可以排除障礙，並且了解能將事情完成並發布的公司政策和辦公室政治的人



What do you discuss at the SDS? 多團隊每日例會要討論什麼？

- **What impediments does my team have that will prevent them or another team from accomplishing their Sprint Goal (or impact the upcoming release?)**
我的團隊有哪些障礙會阻止我的團隊或其他團隊完成他們的Sprint目標(或影響未來的發布)？
- Is my team doing anything that will **prevent another team** from accomplishing their Sprint Goal (or impact the upcoming release?)
我的團隊是否正在做任何會阻止其他團隊完成他們Sprint目標(或影響未來的發布)的事情嗎？
- Have we **discovered any new dependencies** or a way to **mitigate them**?
我們已經發現任何新的相依性和找到降低相依性的方法了嗎？



Daily Scrum and Scaled Daily Scrum

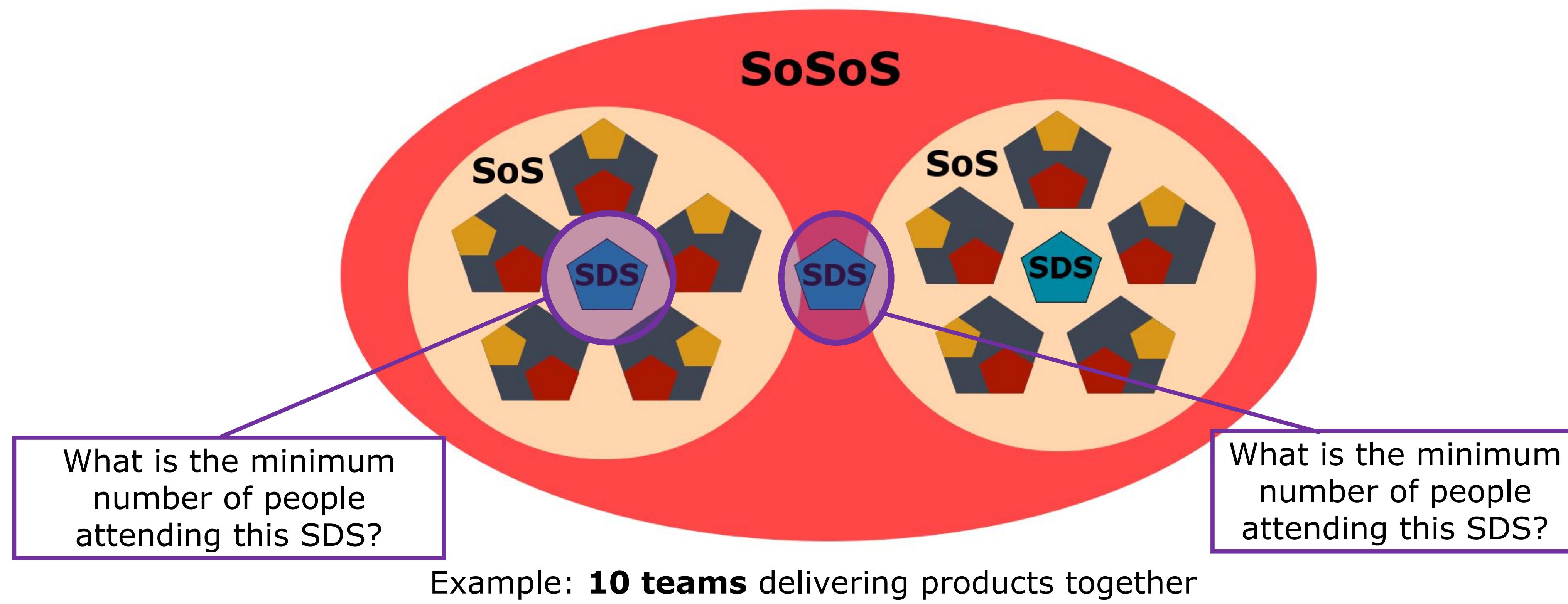
每日例會和多團隊每日例會

- From 8.30 to 9.00 all teams have their **individual Daily Scrum**
從8.30到9.00, 所有團隊都有各自的每日例會
- At 9.00 representatives from all the teams, also the remotes and suppliers, **join the Scrum of Scrums**
上午9:00, 所有團隊的代表(包括遠程的人和供應商)都參加Scrum團隊群
- Scrum of Scrums have a senior manager serving as the **Scrum of Scrums Master (SoSM)**, in charge of systemic impediments
多團隊中的一位資深經理擔任首席SM, 負責涉及全系統性的障礙(排除)



Scrum of Scrum of Scrums (SoSoS) Perspective

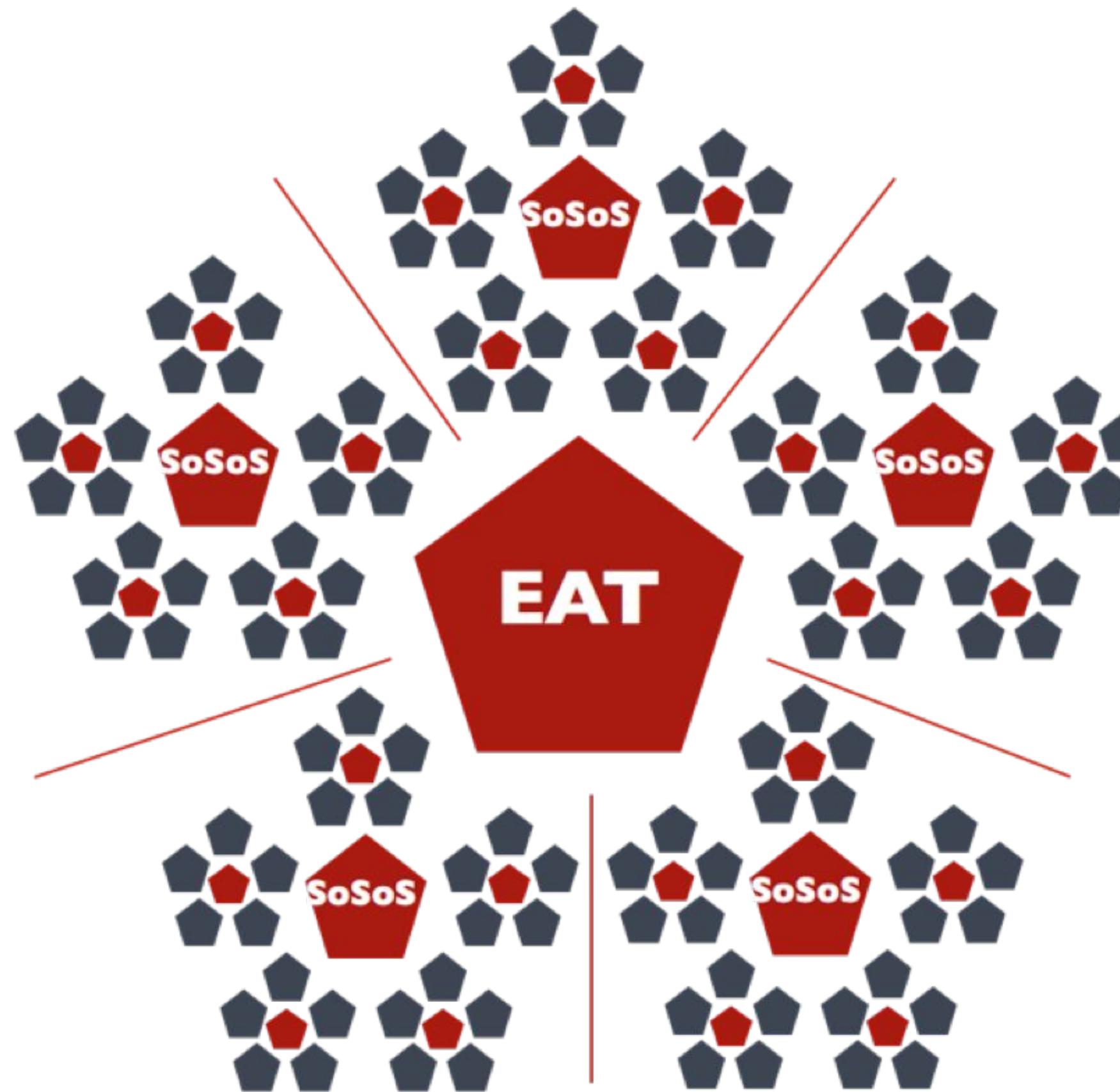
- For **very complex products**, the SoSoS is a **Team of Team of Teams**
針對非常複雜的產品, SoSoS是多團隊的多團隊的團隊
- This is an organic pattern of Scrum Teams that is infinitely scalable (**scale-free architecture**)
這是scrum團隊群的一個根本模式, 是可以無限擴大的(沒有規模限制的架構)



Executive Action Team (EAT) Perspective

Example: 125 Teams in 5 Value Streams

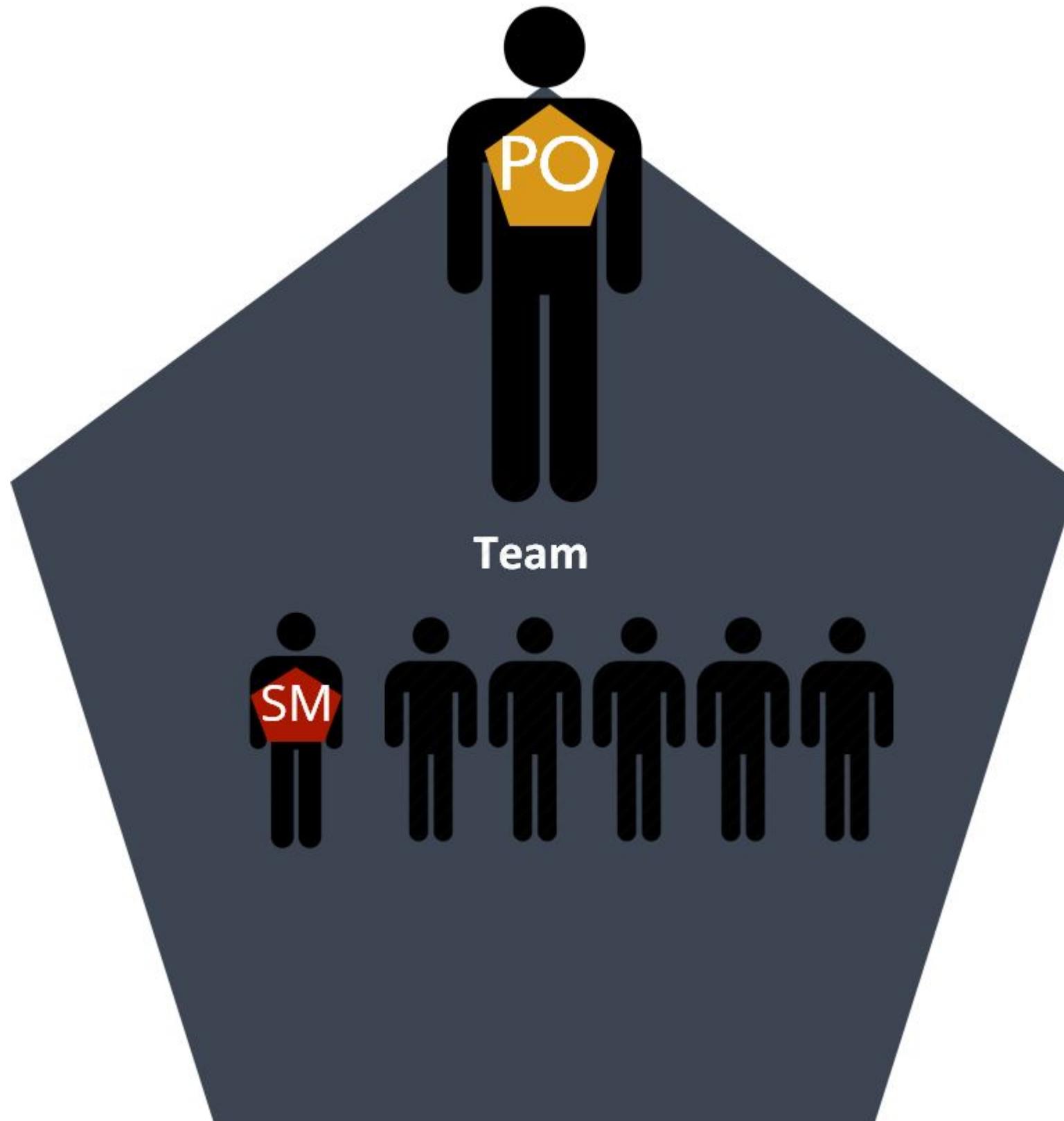
- **EAT** – Eats Impediments the ones that the teams & SoS cannot remove by themselves due to scope, budget, or corporate political power
排除團隊和SoS無法自行解決的障礙
- The **EAT Daily Scrum** is an org-wide SDS (Scaled Daily Scrum)
EAT每日例會是整個組織的多團隊每日例會
- Limits Communication Pathways 減少溝通途徑 (300 vs. 195,000)



- EAT allows **Cross-Value Stream Coordination**
EAT允許跨價值流協調
Ex: 625 People Coordinate in 60 min.
- Measures & improves the quality of Scrum
- Owns **Organizational Transformation Strategy**
負責組織轉型戰略
- **Executes Transformation Strategy** or delegates it to the **Agile Practice**
執行轉型戰略或將其委任給敏捷實踐者

The Team Product Owner

Builds, Refines, Plans 建立、優化、計劃



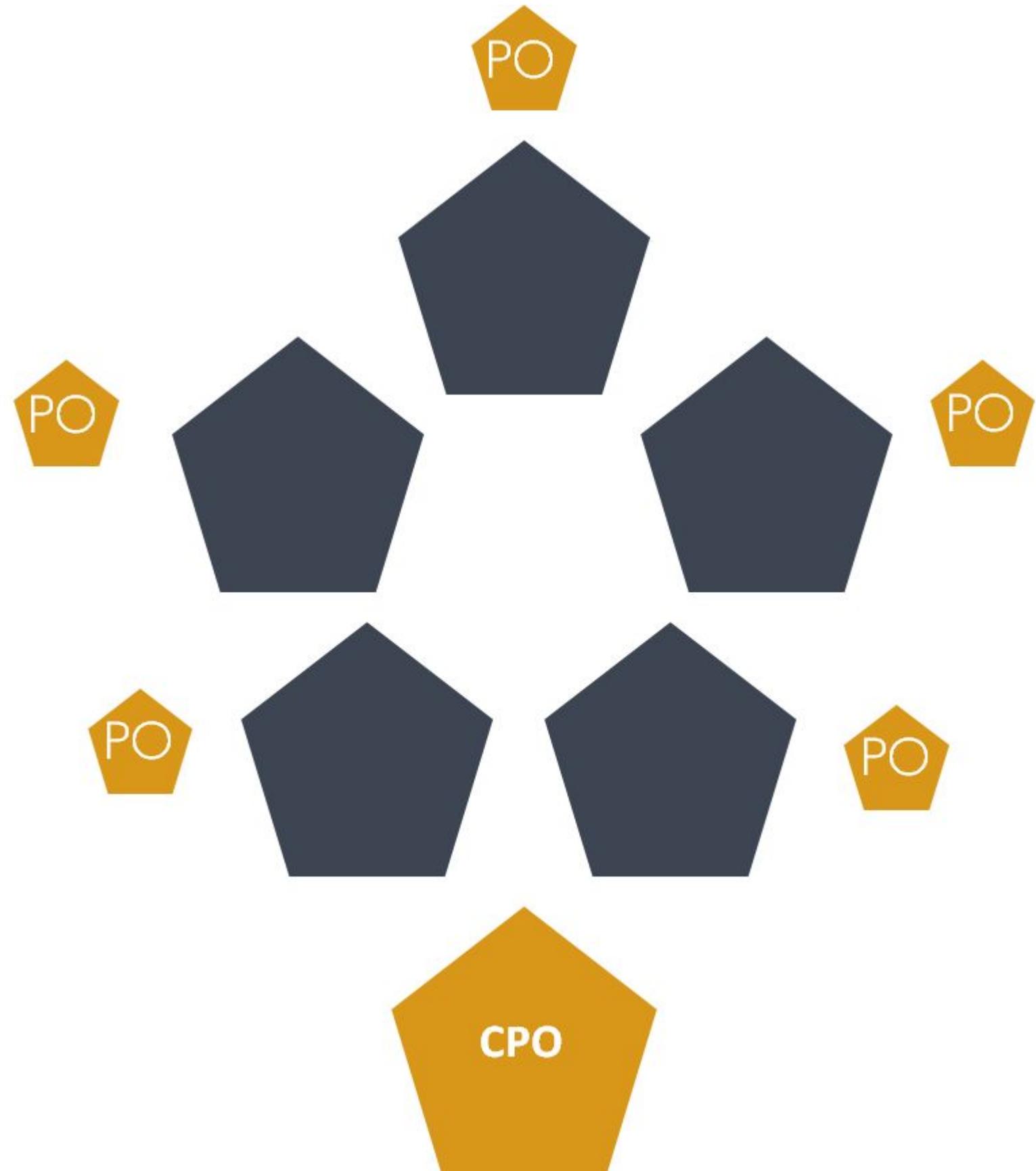
- **Sets Team Priorities**

設置團隊的優先排序

- 50% w/ Customer, 50% w/ Team
50%的時間給客戶、50%的時間給團隊
- Single Backlog 單一產品待辦
 - **Stories** 故事(級別)
 - **Epics** 史詩(級別)
- Focus on What, not How 專注於什麼, 而不是如何
- Systems Thinking 系統思考 - Oversees the Whole
縱觀監督全局
- Sometimes referred to as Line PO
有時被視為直屬PO
- Complete Responsibility Through Trust
- Known-Stable-Interface to the Enterprise
企業的已知穩定介面

Product Owner Team at 1 Level PO群/PO團隊

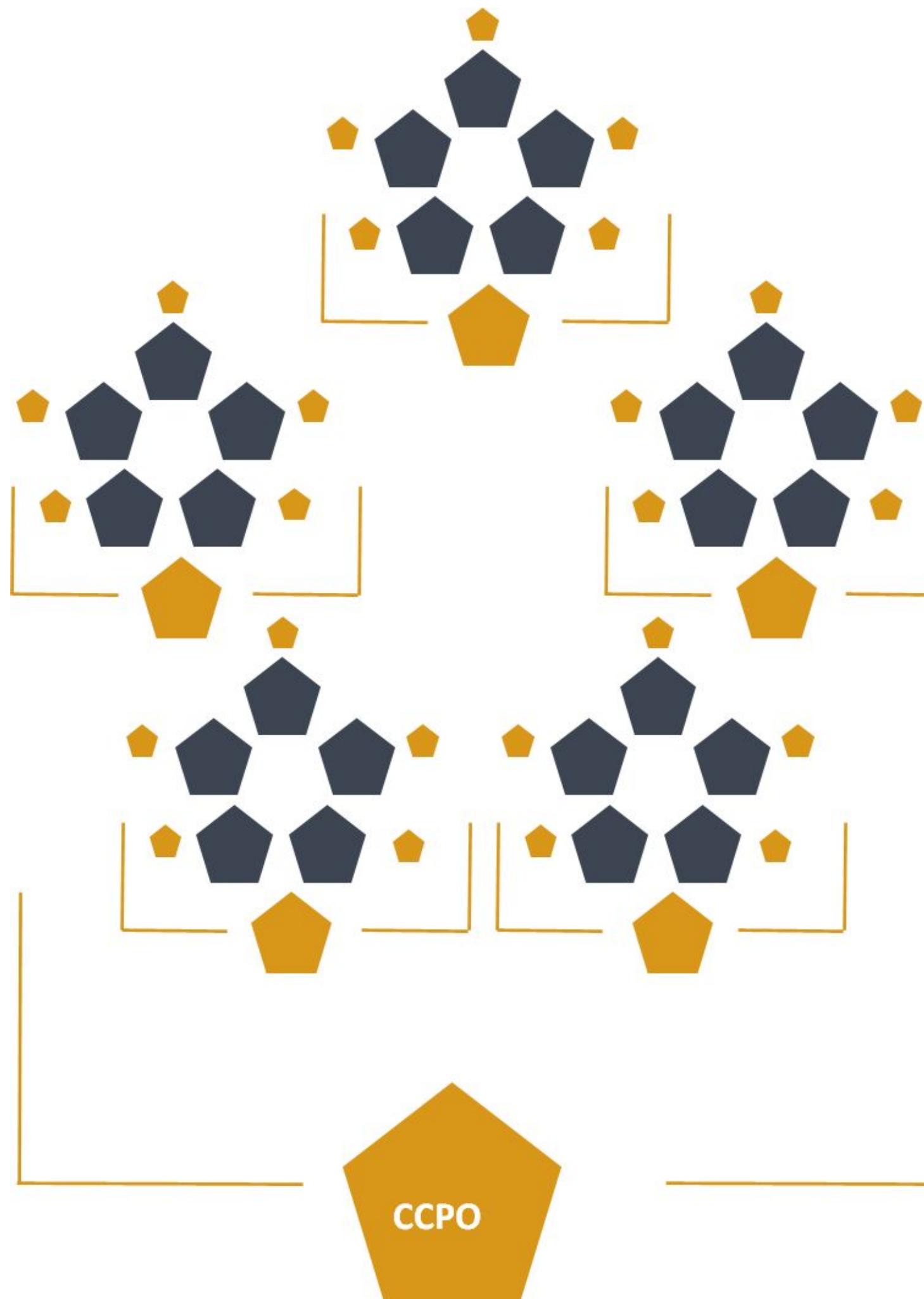
Aligns, Refines, Plans 聯盟合作、優化、計劃



- **Sets Priorities for Multiple Teams**
為多個團隊設定優先排序
- Mirrors Refinement and Planning
跟一個Scrum團隊同樣的優化和計劃會議
- Single Backlog Pulled by Line POs
由直屬PO拉出單一的產品待辦
 - **Epics** 史詩(級別)
 - **Features** 功能(級別)
- **Cross-Team Coordination & Alignment**
跨團隊協調與聯盟合作
- Systems Thinking - Oversees the Whole
系統思考 - 縱觀監督全局

Product Owner Team at Level 2

Aligns, Decomposes, Refines 聯手合作、拆分、優化

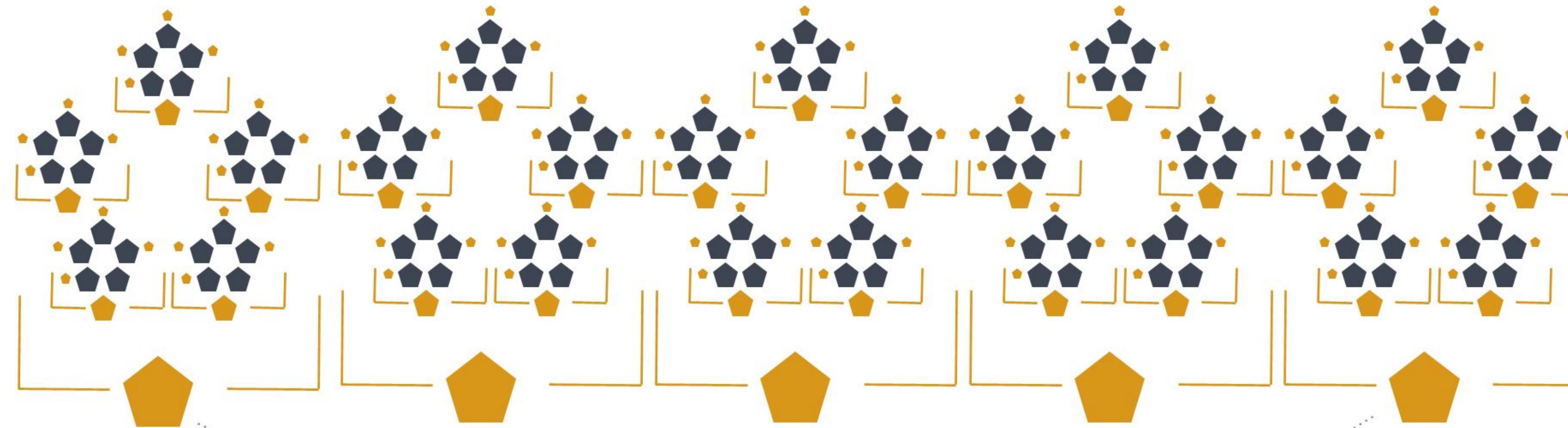


- **Sets Priorities for Multiple Teams**
為多個團隊設定優先排序
- Mirrors Refinement and Planning
跟一個Scrum團隊同樣的優化和計劃會議
- Single Backlog Pulled by CPO team
由首席PO團隊拉出單一的產品待辦
 - **Features** 功能(級別)
 - **Value Streams** 價值流(級別)
- **Cross-Team Coordination & Alignment**
跨團隊協調與聯盟合作
- Systems Thinking - Oversees the Whole
系統思考 - 縱觀監督全局

Executive Meta Scrum (EMS)

Aligns and Sets Strategic Priorities for the Organization

聯手合作、設定組織的戰略排序

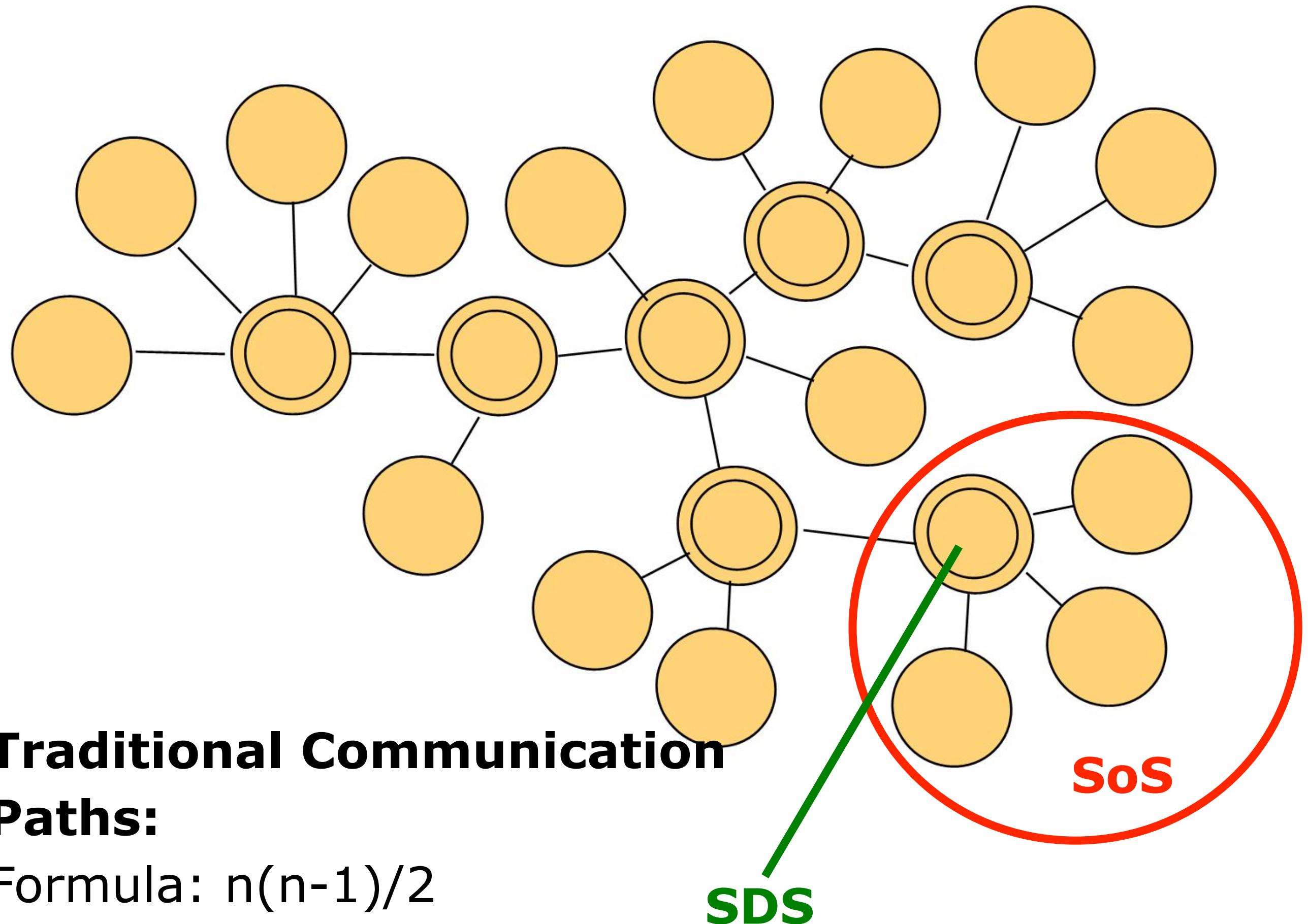


- Owns Organizational Vision 擁有組織願景
- Lead by the Level 1 PO
 - Leader who serves
 - CEO
 - SVP
- Single Backlog Pulled by Level 2 PO
 - Value Streams
 - Initiatives 提出
- Sets Organizational Priorities 設定組織的排序

- Inhales Technical Priorities
- Exhales Organizational Priorities
- Mirrors Refinement & Planning

Scrum of Scrums & the Scaled Daily Scrum

The Scrum Master Role Scaled SM角色大規模化



Traditional Communication Paths:

Formula: $n(n-1)/2$

For 120 people

$$120(119)/2 = 7140$$

Scrum of Scrums Communication Paths:

Same 120 people, 5 person team, 24 teams in total

Same formula: $n(n-1)/2, 5(4)/2 = 10, 24 \text{ teams} * (10) = 240$

plus cross team communication

$$24(23)/2 = 276, \text{ so } 240+276=516$$

That is 516 vs 7140

93% less in communication paths!