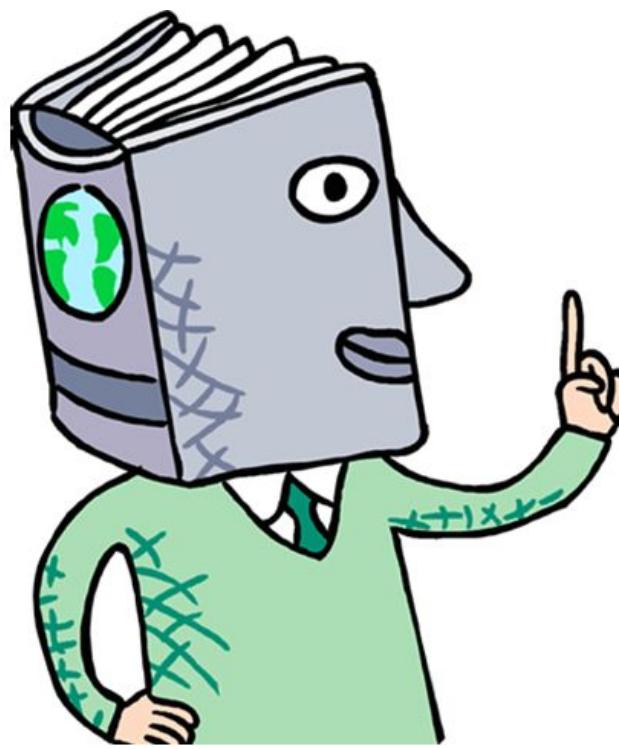


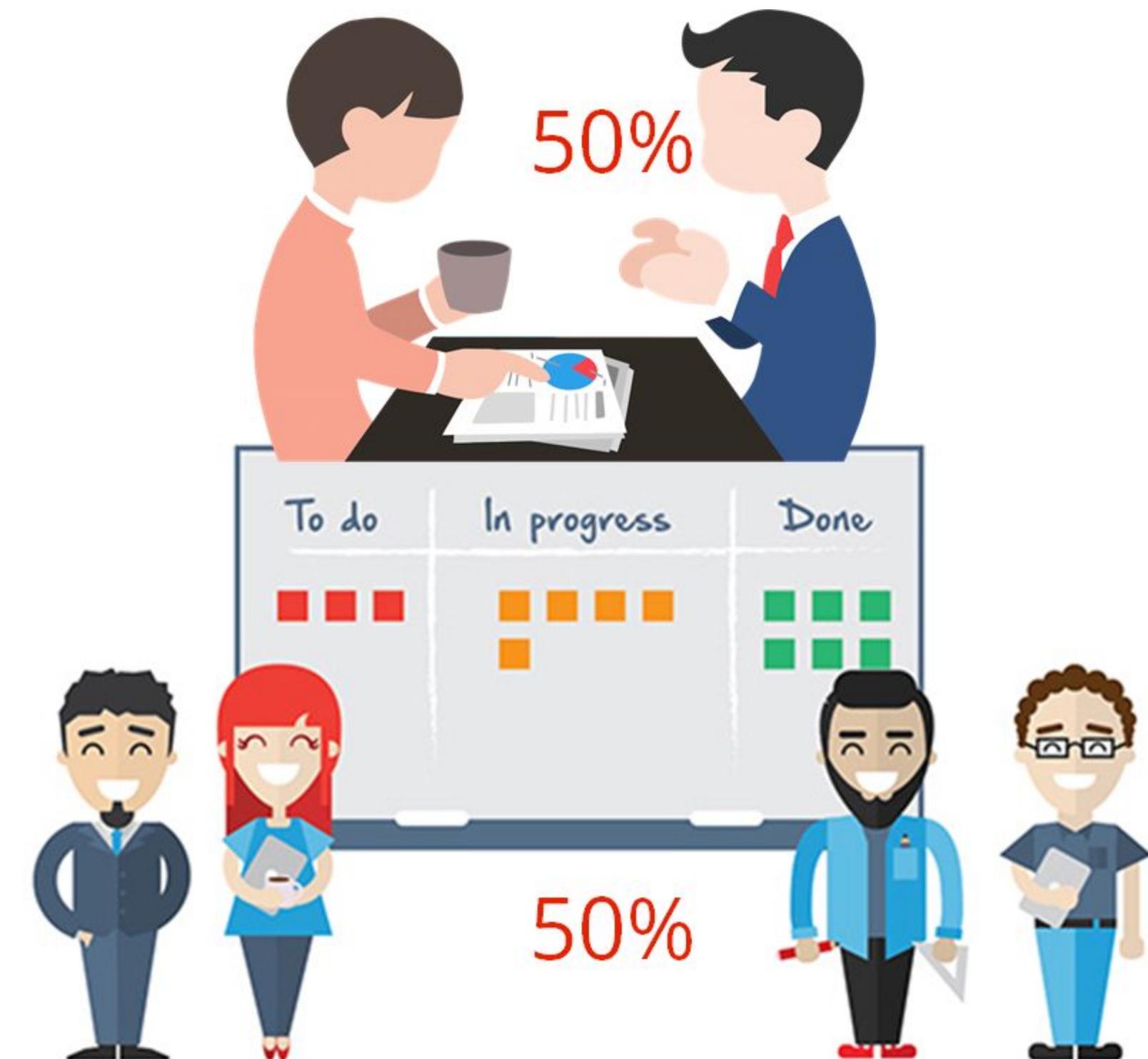
# Product Owner Key Attributes



Decisive 做決定



Knowledgeable 知識淵博



Available  
在客戶和團隊身邊



Empowered 得到授權



Accountable 當責

How many teams can a Product Owner support?

# What do Scrum Masters do? SM的工作

Leadership

Product  
Owner

Scrum  
Master

Developers

Waste

- Scrum Masters are the **keepers of good Scrum** 守護者
- Scrum Master helps the team to **get the work done fast and more effective at a sustainable pace** 幫助團隊以可持續的速率、更快速、更有效地完成工作
- Scrum Masters are **not your boss** 不是你的主管
- Scrum Masters help Product Owner and stakeholders to **forecast a realistic timeline** for the product release 協助產品負責人和利害關係人預測產品發布的實際時間表
- Scrum Masters are **accountable for Team Happiness** 對團隊快樂負責

# The Scrum Master is accountable for The **PROCESS**

流程是SM的當責  
**(the implementation of Scrum 導入Scrum)**

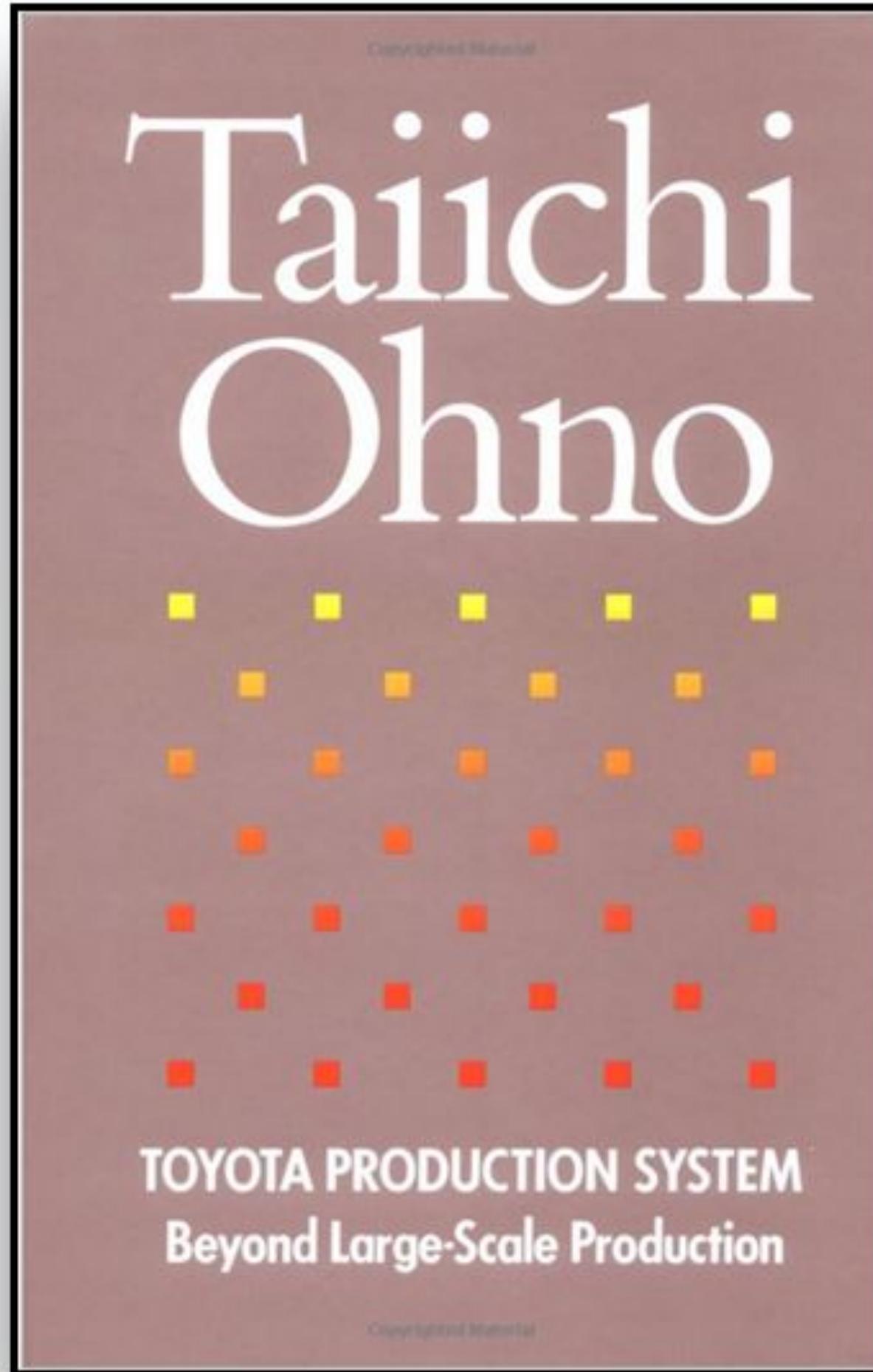
- Scrum Masters are “**leaders who serve**” 僕人式領導者
- **Coach the Team and Organization** to enhance value creation with Scrum 指導團隊和組織透過Scrum增強價值創造
- **Facilitate** the events effectively and repeatedly 有效且重複地引導活動
- **Protect** the Scrum Team **from interruptions** 保護Scrum團隊不受干擾
- Help the team to **make their work visible** to the stakeholders, customers, and organization 幫助團隊向利害關係人、客戶和組織透明化他們的工作
- Facilitates the creation of **process improvements** (to **ensure impediments are removed**) 建立流程改善以利阻礙移除



Scrum is a **simple framework** that requires consistent **discipline** to execute effectively  
Scrum 是一個簡單的架構，需要一致的紀律才能有效執行

# Having no problem is the biggest problem of all

最大的問題是沒有問題



“No one has more trouble  
than the person who  
claims to have no trouble.”

- Taiichi Ohno, Toyota

Production System

"說沒問題的人  
問題最多"



大野耐一

豐田生產方式系統化的靈魂人物

# What is the Role of Management in Scrum?

Leadership /  
Management

Product  
Owner

Scrum  
Master

Developers

Waste

- You still have Management 你仍然有管理階層
- Scrum would like to **transform the Management into Leadership** 將管理階層轉變為領導
- Management will work if your work is simple 傳統的管理階層在簡單工作環境會起作用
- If you want **creativity, innovation, in a complex environment**, then you need leadership 如果你想要在複雜的環境中發揮創意、創新，那麼你就需要領導
- Leadership makes sure **the teams have all the resources** (the people, the skills, the tools, the consistent leading practices...) they need 領導確保團隊擁有他們需要的所有資源(人員、技能、工具、一致的做法...)
- Leadership **holding SM, PO Developers accountable** 領導要求 SM、PO、Developers當責

# Who does What in Scrum?

- If you've read the Scrum Guide, you may have noticed that "**Project Manager**" is **missing** 專案經理不見了, So is the title "Manager"
- Almost everything your company does to create and deliver products and projects will still get done; what changes is **who does it**. 幾乎所有為產品及專案交付所做的事仍繼續完成, 改變的是誰做了這些事
- Some of what your organization does is **waste** - let's **get rid of that!** 組織所做的一些事情是浪費 - 一起來扔掉吧!
- In Scrum, the title PM doesn't exist because the function is **split amongst the different roles** 在Scrum團隊, PM的工作被分配到不同的角色



# Exercise: Map the work to the role

- As a team, write down all responsibilities of a traditional Project Manager / Management  
寫下傳統專案經理的所有職責
- Put one responsibility (example: Hiring) on each Post-it  
將每一個職責分別寫在不同的便利貼上
- Do this in the AGILE way
  - Everyone writes sticky notes 每個人都要寫
  - Product Owner coordinates 由PO協調完成
- Now put these Post-Its into 5 areas (移動便利貼 ~ 將所有職責分配到以下五類)



Leadership /  
Management

Product  
Owner

Scrum  
Master

Developers

Waste

# Exercise: Map the work to the role



# Key Points

- Scrum Team has **3 Roles / Accountabilities**. 團隊有 3 個角色/當責
- The **organization supports the Scrum Teams** 組織支持 Scrum 團隊
- **Scrum Team is responsible for delivering** a potentially releasable increment each Sprint 團隊負責在每個Sprint中交付可能可發布的增量
- Scrum Team is responsible for the **Quality** 團隊對品質負責
- Scrum Team is **cross-functional** and **self-managing** 團隊是跨職能且自我管理的
- **Developers** accountable for the **How** 開發人員對如何進行當責
- **Product Owner** accountables for the **What** 產品負責人對什麼負責
- **Scrum Master** accountables for the **Process** 對流程當責
- Scrum helps the **Management transform into Leadership** 幫助管理階層轉變為領導

# Agile Management

---

As a member of a Scrum Team, I need a clear understanding of what to expect from managers and what they expect from each role, so that we can work well together

身為Scrum團隊的一員，我要清楚了解對管理者該有何期待，以及他們對每個Scrum團隊角色的期待，所以我們才能成功地一起工作

# What is the Role of Management? 管理階層的工作

Leadership

Product  
Owner

Scrum  
Master

Developers

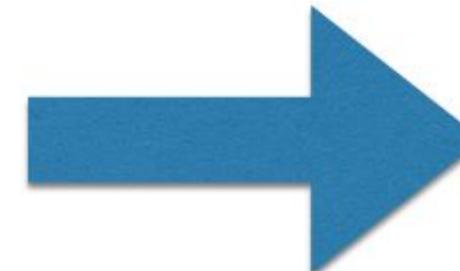
Waste

# The Role of Management: Becomes Leadership

## 管理者成為領導者

During an Agile transformation, the role of **MANAGEMENT evolves** 在敏捷轉型中，管理者的角色產生演變  
They change from managers of work to **LEADERS** who maintain a vision and enable the teams  
他們從經理人轉變為維護願景並帶動團隊成功的領導者

- Define Organizational **Structure**
- **Determine what work** needs to be accomplished
- **Decide who and how tasks** will be accomplished at a team or individual level
- Create a **culture that fears failure** to try to **eliminate risk**
- **Define Standards**
- Decide who should **receive certain information**
- Develops **Rules**
- Host meetings to **solely check people are doing what they asked**



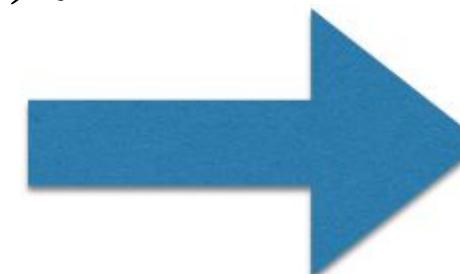
- Provide challenging **prioritized goals** for the teams
- **Eliminate organizational debt**
  - Create a business plan/organization that works
  - Provide all resources the teams need
- **Remove impediments** identified by the teams
  - Assure teams are set up to maximize production
  - Remove waste - Value Stream Mapping
- **Hold Product Owners accountable** for value delivered per point
- **Hold Scrum Masters accountable** for process improvement and team happiness
- **Hold Development Teams accountable** for quality increase and empower technical debt remediation

# The Role of Management: Becomes Leadership

## 管理者成為領導者

During an Agile transformation, the role of **MANAGEMENT evolves** 在敏捷轉型中，管理者的角色產生演變  
They change from managers of work to **LEADERS** who maintain a vision and enable the teams  
他們從經理人轉變為維護願景並帶動團隊成功的領導者

- 定義組織結構
- 決定哪些工作需要完成
- 決定誰以及什麼工作要如何由團隊或個人完成
- 建立害怕失敗的文化來消除風險
- 定義標準為何
- 決定誰能接收那些訊息
- 建立規則
- 會議目的用來檢查人們是否按照要求來做事



- 提供團隊具挑戰性的**優先目標**
- **消除組織債**
  - 提供團隊所需要的所有資源
  - 創建可行的商業計劃/組織
- 移除團隊所定義的**障礙**
  - 認為團隊是為了完成**生產最大化**而存在
  - 移除浪費 - 價值流圖
- 每一個點數所做的交付是**Product Owner**的當責
- 流程改善和團隊的快樂是**Scrum Master**的當責
- 交付品質的提升和技術債的補救是**開發團隊**的當則

# Management vs Leadership

## 管理 vs 領導

### Management:

- Hierarchy Operating System 階層營運系統
- **Stable and Efficient 穩定且效率高**

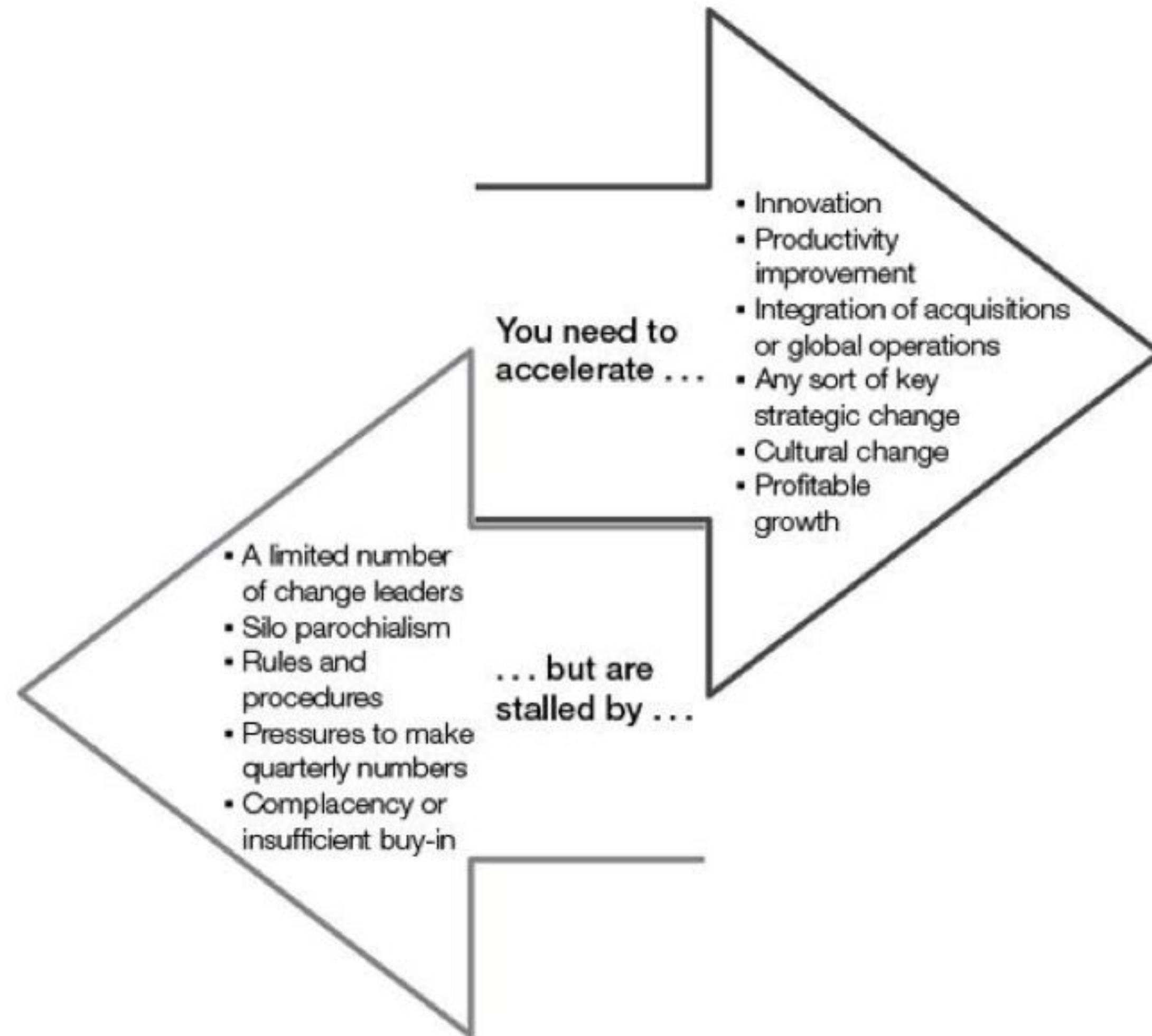
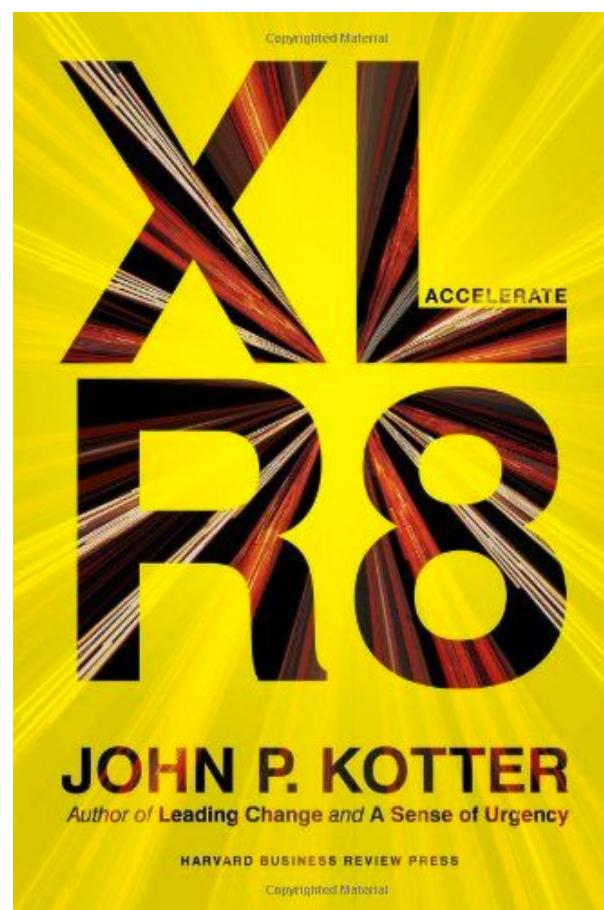
### Leadership:

- Network Operating System 網狀營運系統
- **Agility and Speed 敏捷及速度**
- Make sure **every team is following the same Scrum practices**  
**確保每個團隊都遵守相同的Scrum運作**
- Holding SM (Process Improvement流程改善), PO (Product related產品相關),  
and Developers (Quality交付品質) accountable 每個人為其職責負責
- And **help the teams to remove these impediments** that they can't  
resolve by themselves 協助團隊移除他們無法自己移除的障礙物

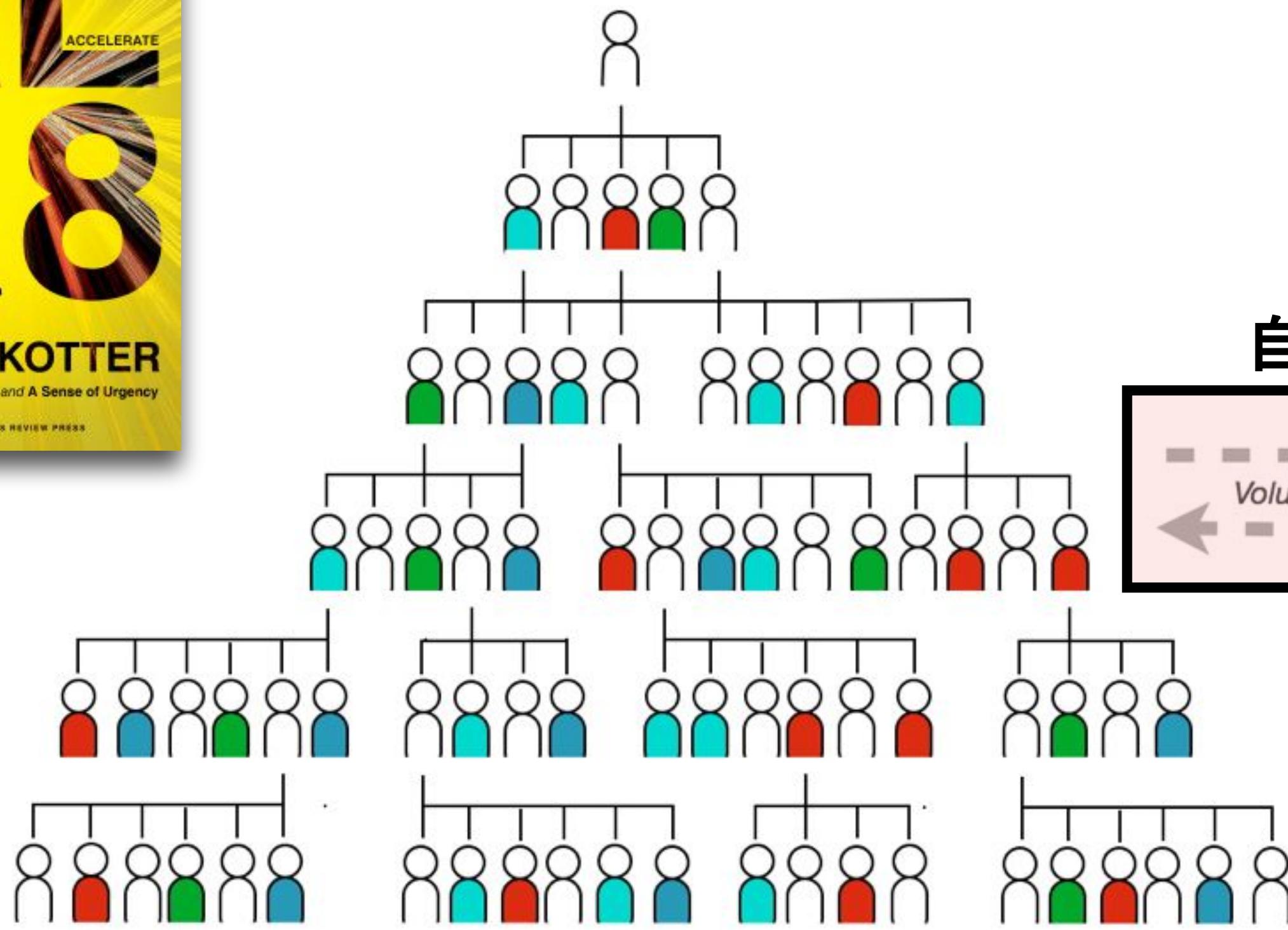
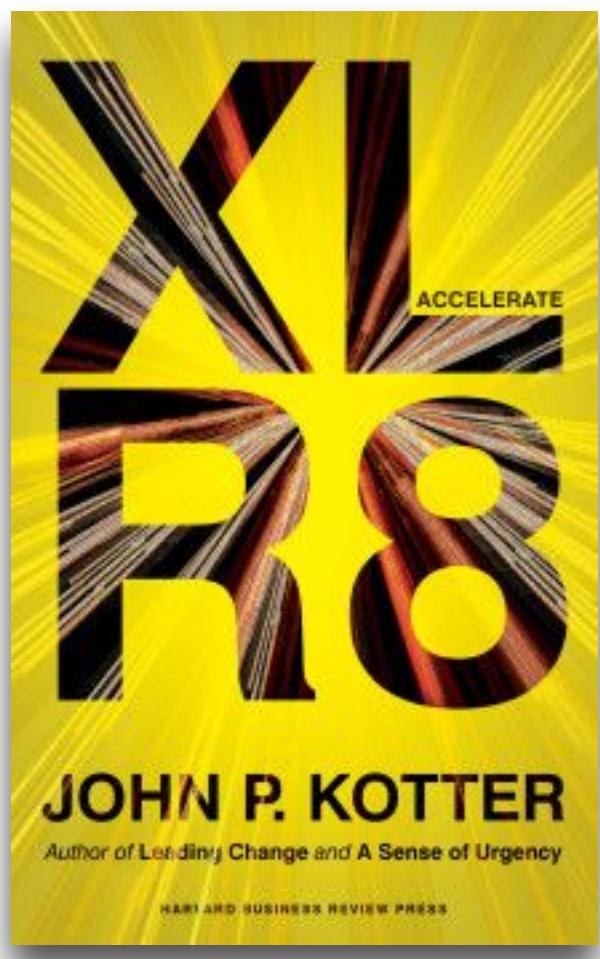


# Install a Dual Operating System To Avoid Failure

## 建立雙作業系統 遠離失敗



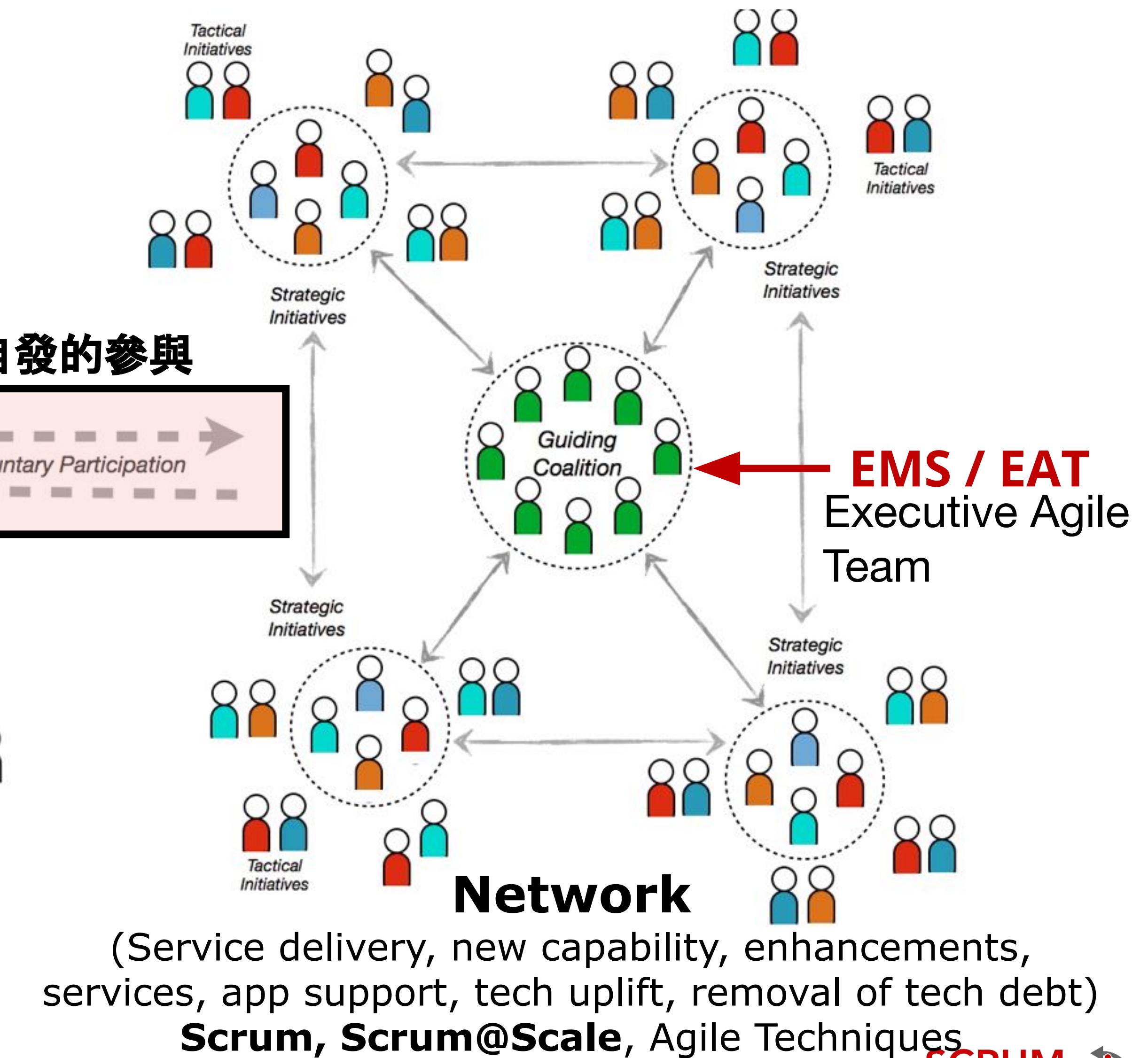
# Creating a Dual Operating System



## Hierarchy 階層

(KLO, Data Center Management, Service Desk)

Tools like Kanban, ScrumBan, Lean etc



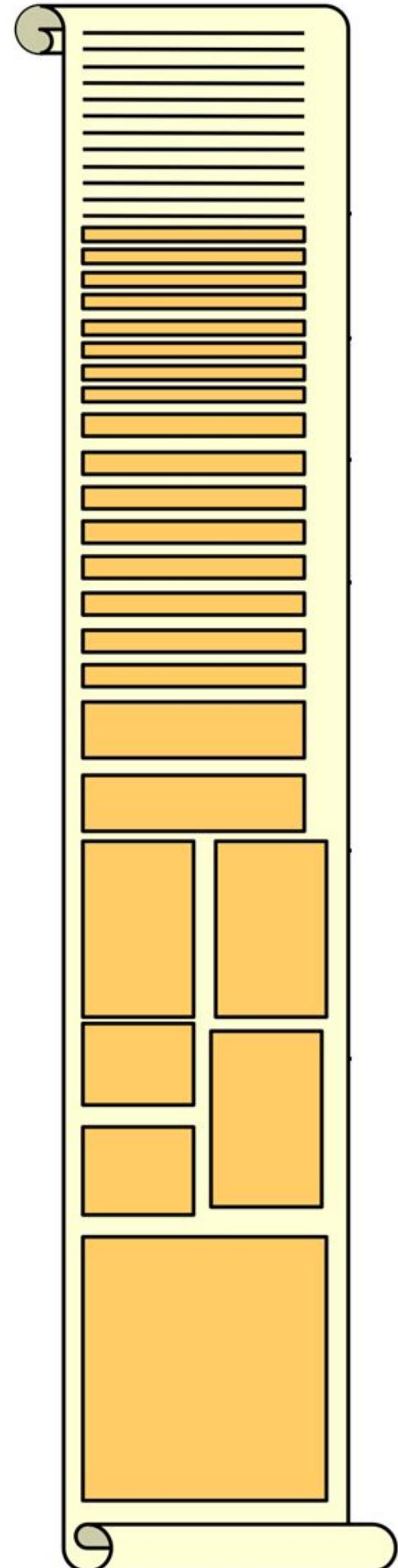
# The Product Backlog 產品待辦

As a member of a Scrum Team, I need to know how the Product Backlog is created, so that I can help my team build a good one

身為Scrum團隊的一員，我要了解產品待辦清單是如何產生的，  
所以我才能協助團隊完成好的產品待辦清單

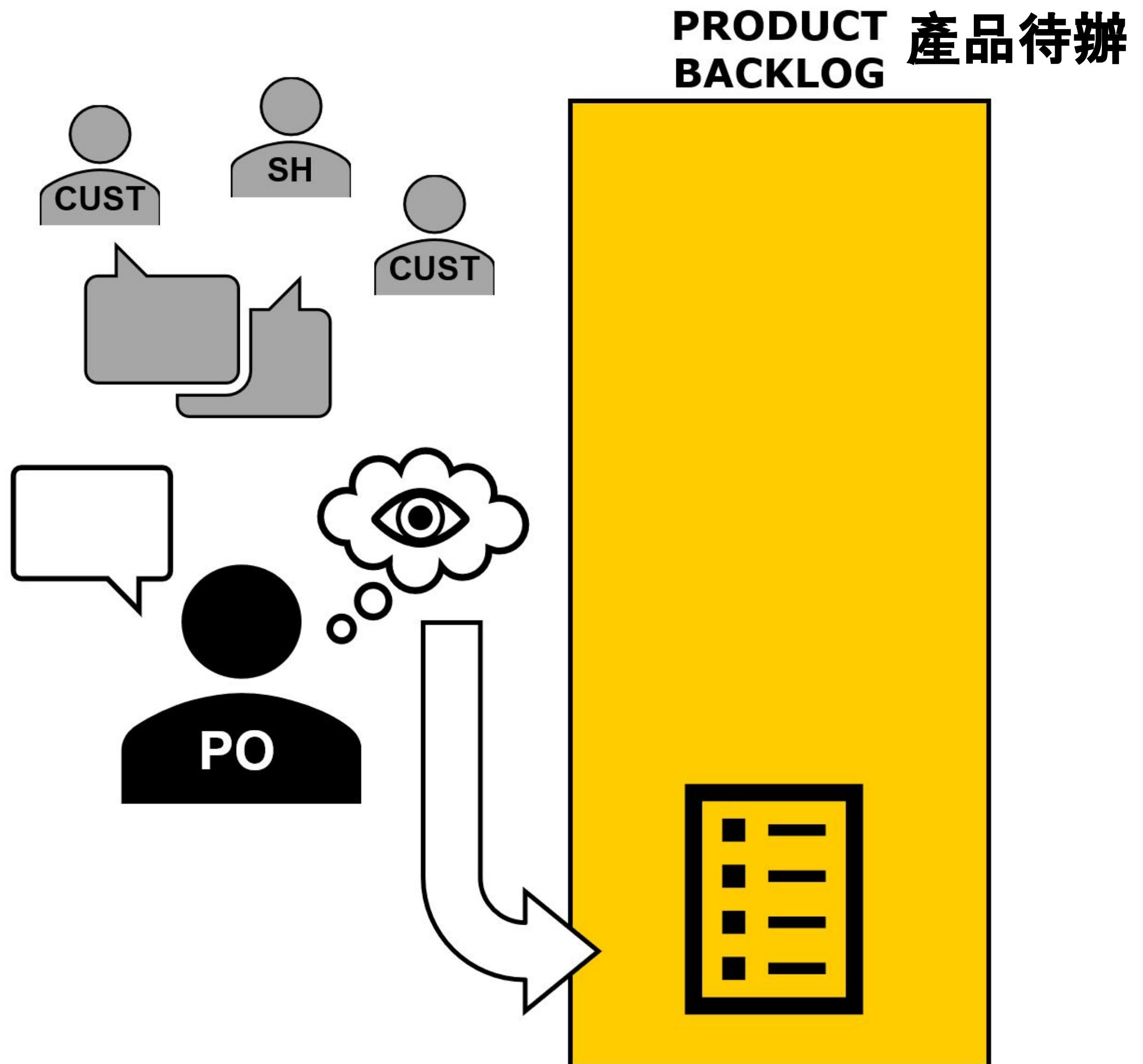
# Product Backlog 產品待辦

- The Product Backlog serves as a collection of **work to be done by the Scrum Team**, known as Product Backlog Items (PBIs), **ordered by business value** 產品待辦清單是Scrum團隊要完成的工作的總和, 就是產品待辦事項(PBI), 依商業價值排序
- **Anyone can add** (make a suggestion or request to the Product Owner) **anything** to the product backlog **at any time**  
任何人都可以隨時(給PO建議或要求)添加任何產品待辦
- The **Product Owner is the final authority** on **ordering** the Product Backlog PO  
是產品待辦排列順序的最終拍板定案者
- The **Product Goal** is the **COMMITMENT** for the **Product Backlog**. The Scrum Team must fulfill (or abandon) a Product Goal before taking on the next 產品目標是針對產品待辦的承諾
- There is **only one Product Backlog** which is shared across teams **working on the same product** 只會有一個產品待辦, 分享給所有為相同產品交付而做的團隊遵循
- The majority of Scrum teams use **User Stories as PBIs** 大多數Scrum 團隊使用"用戶故事"當作待辦事項(PBI)



**Why we refer to the Product Backlog as a “wish list”?**  
**為什麼我們將Product Backlog產品待辦視為“願望清單”？**

# Scrum Framework

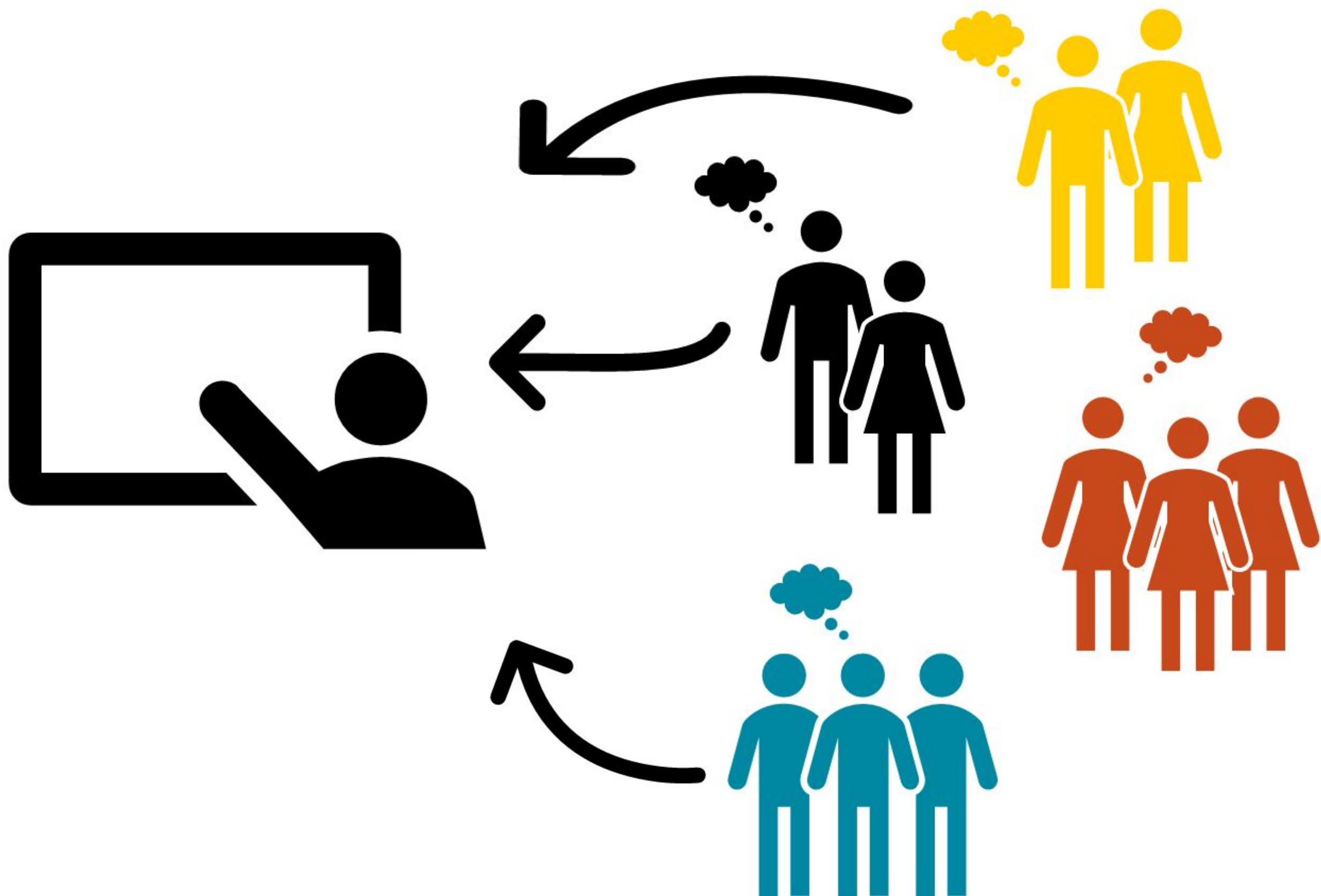


# Product Backlog: Anyone can Add

## 產品待辦: 每個人都可以添加

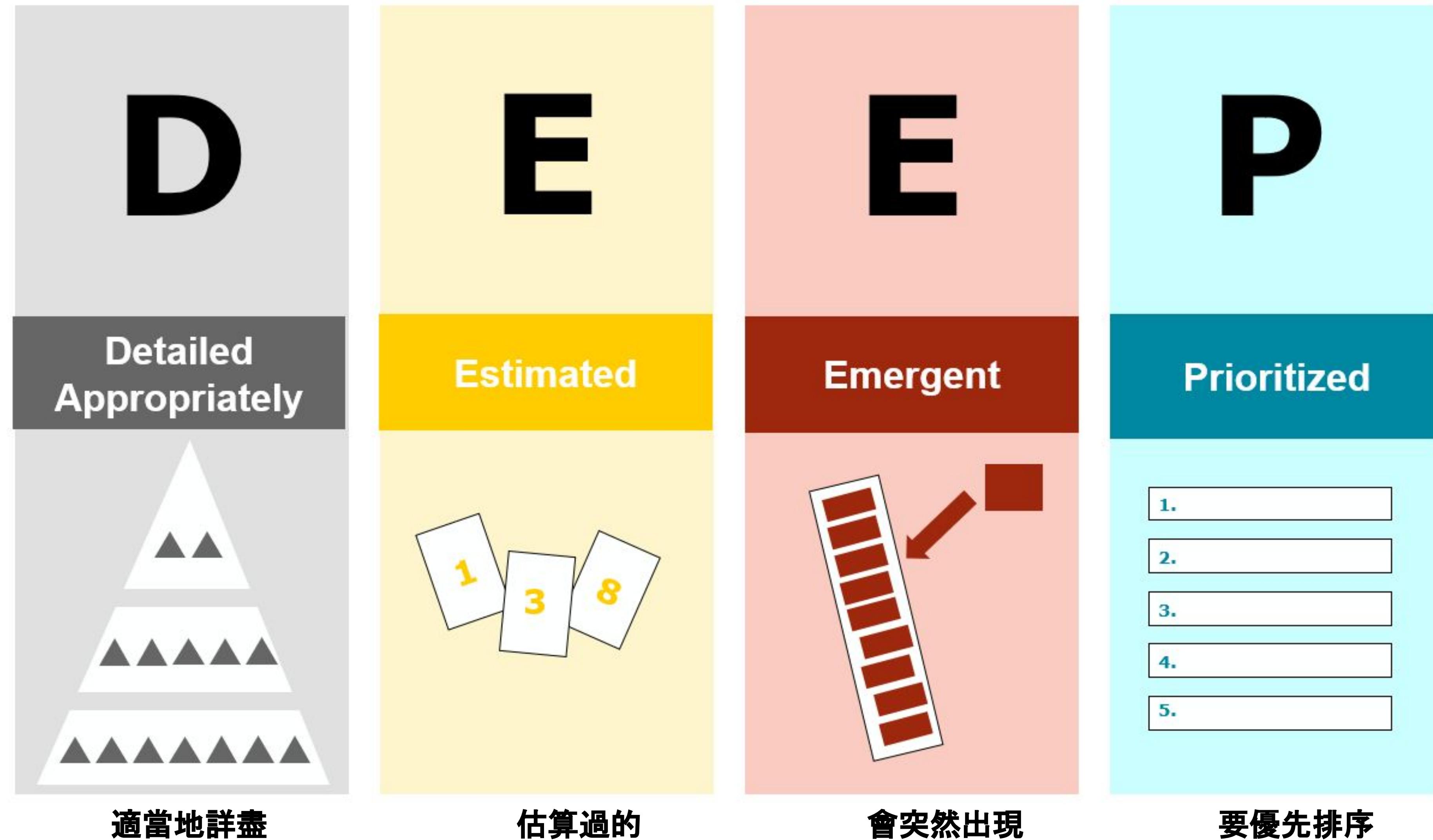
- **Customers 顧客**
- **Stakeholders 利益相關者**
- **Members of the Scrum Team 團隊成員**
- **Other Teams 其他團隊**
- No matter what tools or company procedures are established, the most effective method for introducing anything into the product backlog is by **conversation with the PO.**

在產品待辦上添加事項的最佳方式是  
直接與PO對話溝通



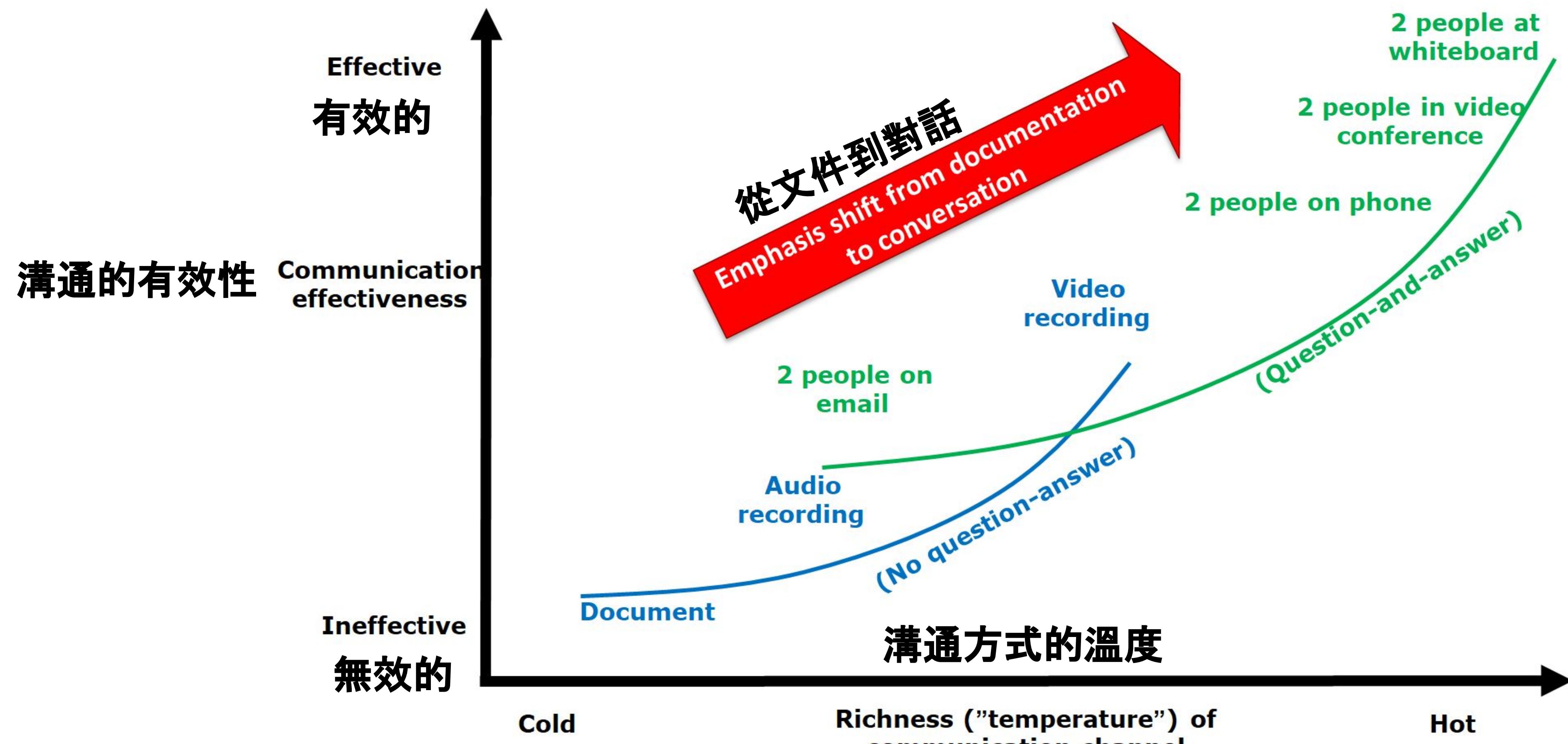
# Product Backlog Characteristics

## 產品待辦的特質



# Product Backlog Improves Communication Effectiveness

## 產品待辦能提高溝通效率

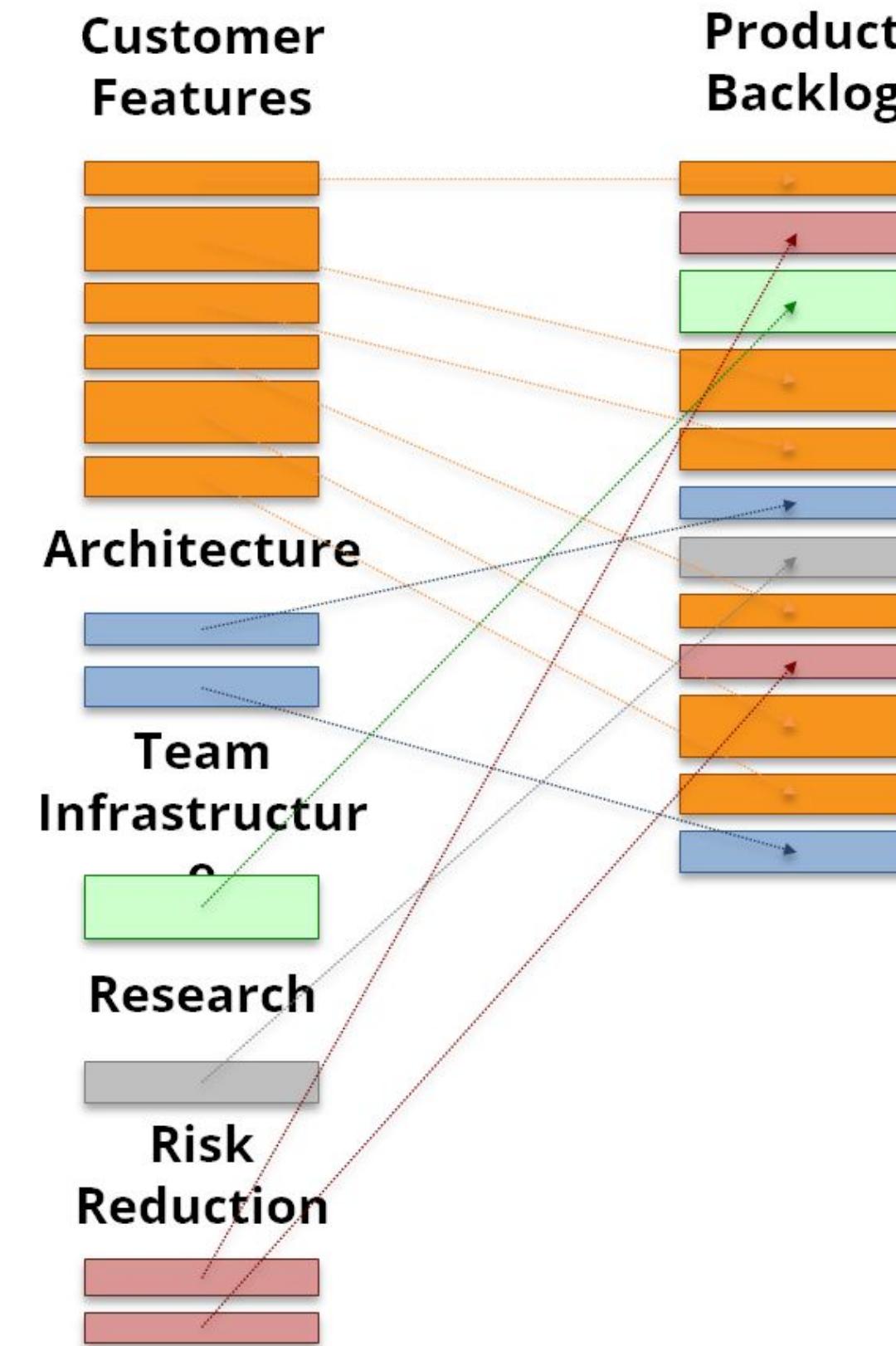


Source: research from McCarthy and Monk (1994) and Scrum Inc 2006-2015

# Product Backlog Composed of Different “Product Backlog Items”

## 產品待辦PB由不同的產品待辦事項PBI所組成

Backlog items include **everything** the team needs to do **in one ordered set of activities** 產品待辦事項包含團隊必須在一組排序的行動中所要做的一切事情



Wherever possible, backlog items should **deliver complete vertical slices of functionality** across work layers  
待辦事項的交付應該是完整垂直拆分的功能



It's how you slice a cake 像切蛋糕的方式

The best teams include **process improvements, bugs and technical debt fixes explicitly as PBIs**  
最佳團隊會明確地像做PBI一樣，算入流程改進、錯誤和技術債修復

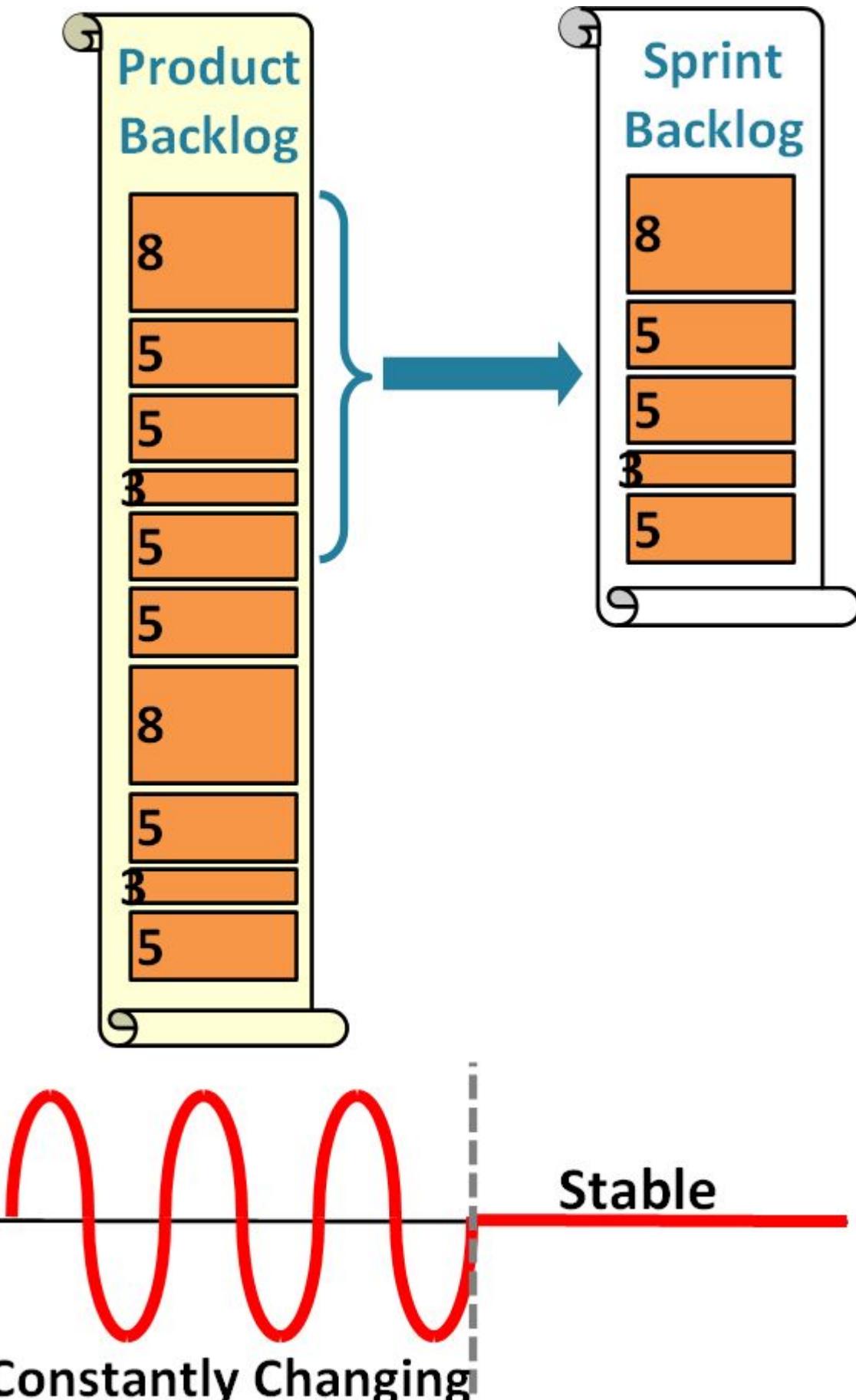
# Product Backlog is a Working Vision of the Product

產品待辦是產品的運作中的願景

Unlike the Sprint Backlog, Product Backlog can change at any time

產品待辦和Sprint待辦不同，產品待辦可以隨時更動

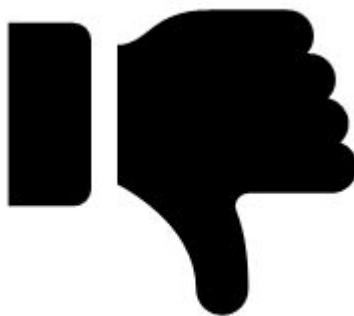
- **Any stakeholder can add anything** to the Product Backlog  
任何利益相關者都可以加任何內容
- Product Owner **orders** the Backlog 待辦是由PO排序
- Backlog Items that are **lower priority** can be defined **more roughly**  
優先級較低的可以比較粗略描述
- The PO and Developers should **spend time each Sprint to refine** the Backlog  
PO和團隊在每個Sprint都要花時間來優化待辦
  - Build in lessons learned from earlier work 包括前次所學到的
  - Refine feature definitions so they are "**READY**" **when the time comes**  
優化，以便在時間到時“準備就緒”



# Backlog Maintenance 待辦的排列維持

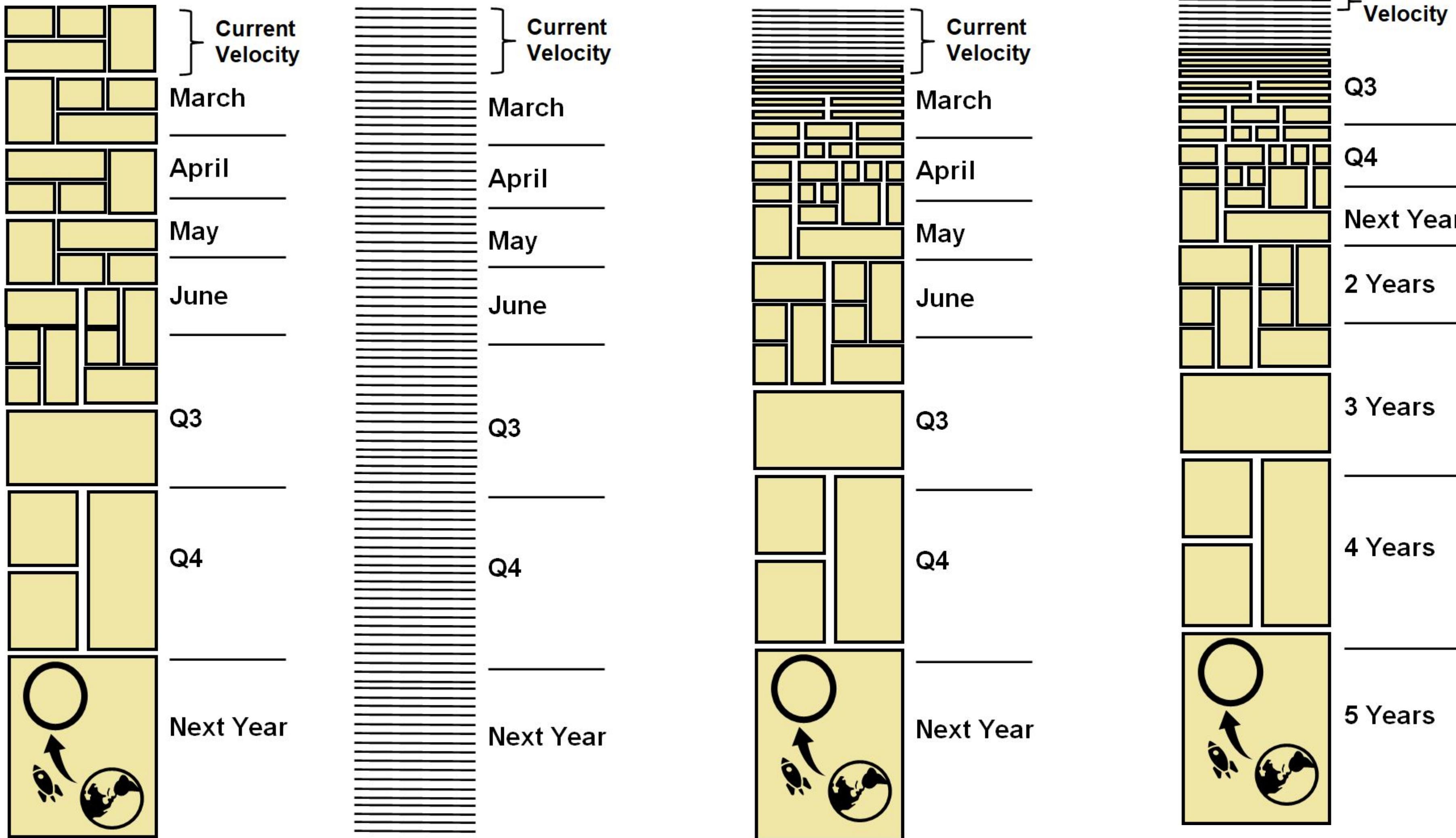


OR



Show me the  
finger (thumb)

## ● 目前的速率



# Backlog Refinement

## 產品待辦的優化 (\*)

---

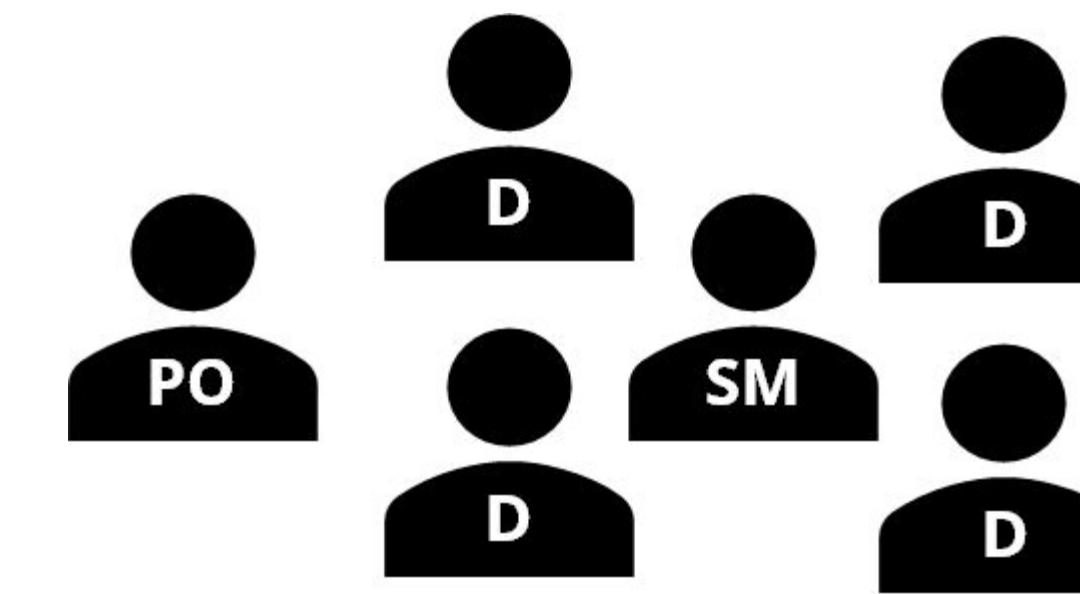
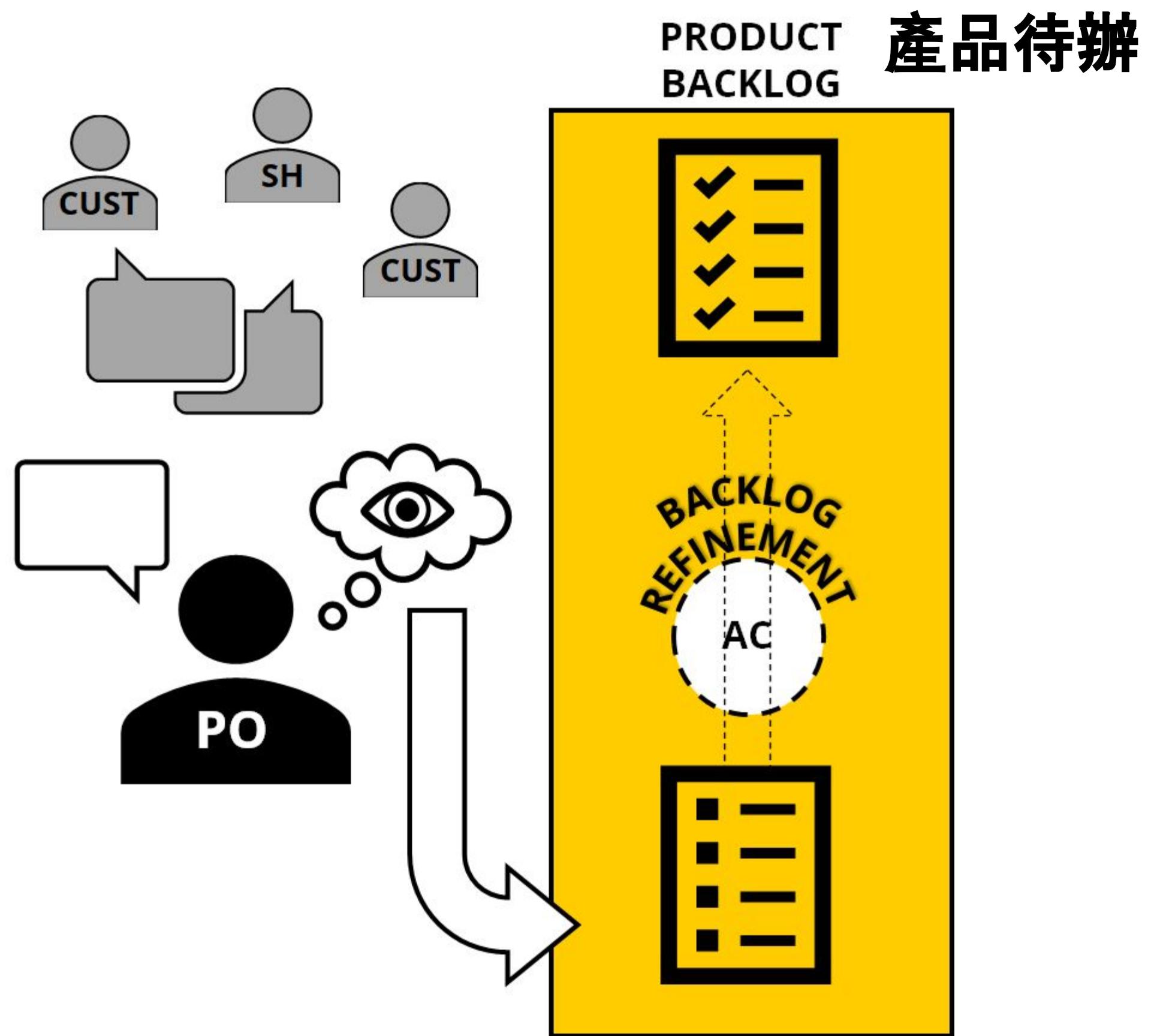
As a member of a Scrum Team, I need to dedicate up to 10% of my time to Product Backlog Refinement, so that backlog can be Ready for Sprint Planning and my team can double our performance.

身為Scrum團隊的一員，我要投入10%的工作時間去做產品待辦優化，  
所以待辦才能在Sprint規劃會議前準備好，並且讓團隊能夠加倍

# Learning Objectives:

- State that the Scrum team owes **up to 10% of their Sprint time** to the Product Owner for the purpose of refining the Product Backlog. Scrum團隊需要將Sprint時間的10%交給產品負責人，以完善產品待辦事項清單
- Recognize that Backlog **Refinement is a Product Owner meeting** 是一個產品負責人會議 and that the Scrum Master may help to facilitate it.
- Recognize that Product Backlog Items (PBIs) can be any size and become **more granular and detailed as they get closer to execution.** 產品待辦事項(PBI)可以是任意大小，並且隨著接近執行而變得更加細粒度和詳細。
- Create PBIs using a format that incorporates the "**who," "what," and "why**" (e.g., **user story format**). 用「誰」、「什麼」和「為什麼」的格式(例如使用者故事格式)建立 PBI。
- Explain why we want to get PBIs in a “ready” state and give **examples of “Definitions of Ready.”** 解釋為什麼我們希望PBI處於「準備好」狀態，並舉例說明「準備好的定義」。
- **Distinguish “Definition of Done” from “Acceptance Criteria.”** 區分「完成的定義」和「驗收標準」。
- Apply **story-slicing** techniques 使用者故事切割技術 that enable swarming on PBIs.

# Scrum Framework



# Backlog Refinement 優化待辦

## OBJECTIVE:

Develop **1-3 Sprints of "READY" Product Backlog Items** which can be pulled during Sprint Planning to create a Sprint Backlog. 要做好1-3個Sprint的"準備好的"產品待辦事項，可以拉到Sprint規劃會議上所做的Sprint待辦清單上

- **Clarify the context of upcoming PBIs through conversation** with the PO. 經由與PO的溝通來釐清接下來要做的PBI的來龍去脈
- **Break down PBI's** into Sprint-sized pieces and reorder as needed 將PBI再拆分成Sprint可以完成的事項，若需要就重新排列優先順序
- **Estimate the effort** required to complete each PBI. 估算完成每個PBI需要的工作量
- The goal is to create a mutual understanding of **WHAT (not the HOW)** needs to be done which includes not just development 目標是就需要做什麼(而不是如何做)達成共識，其中不僅包括開發

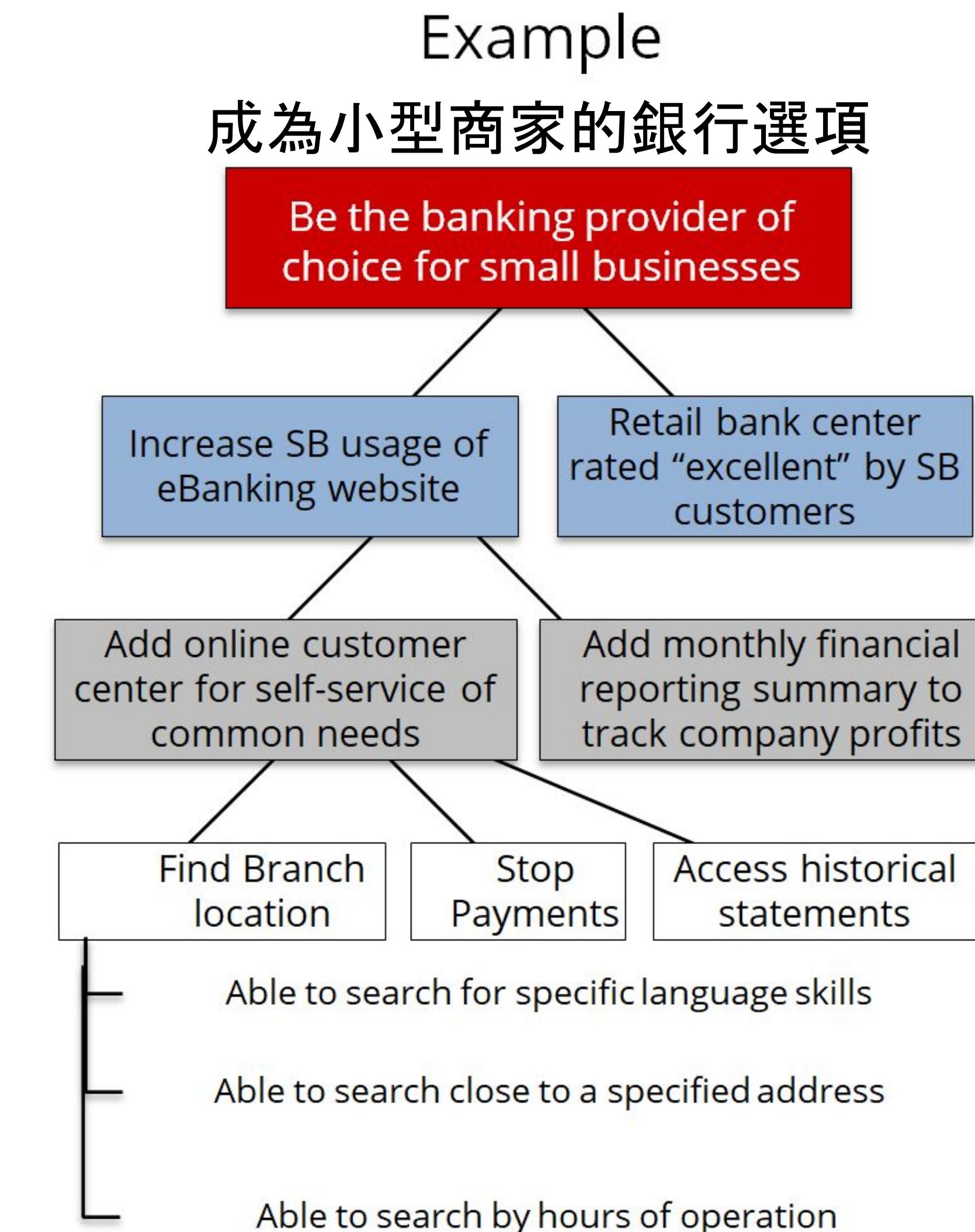
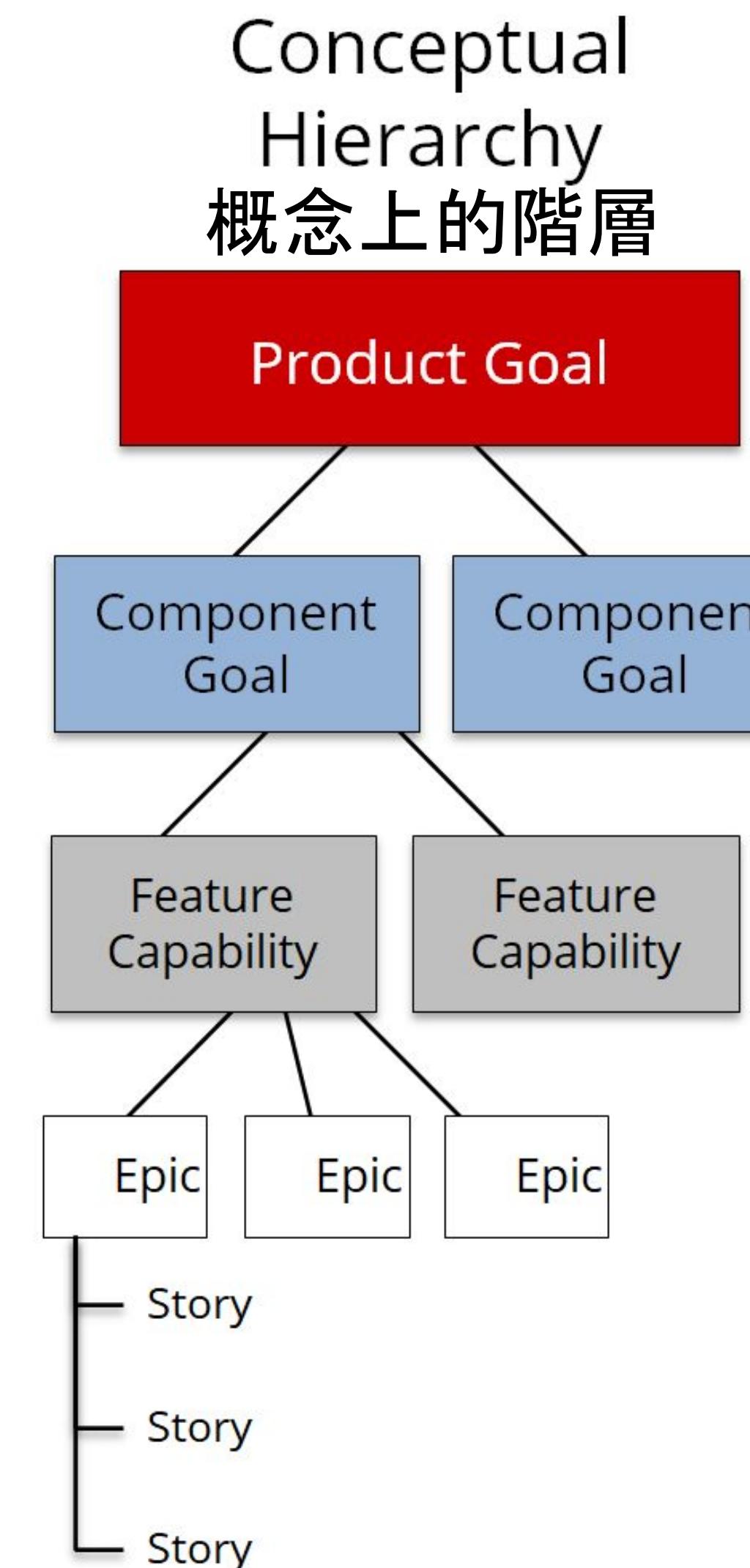
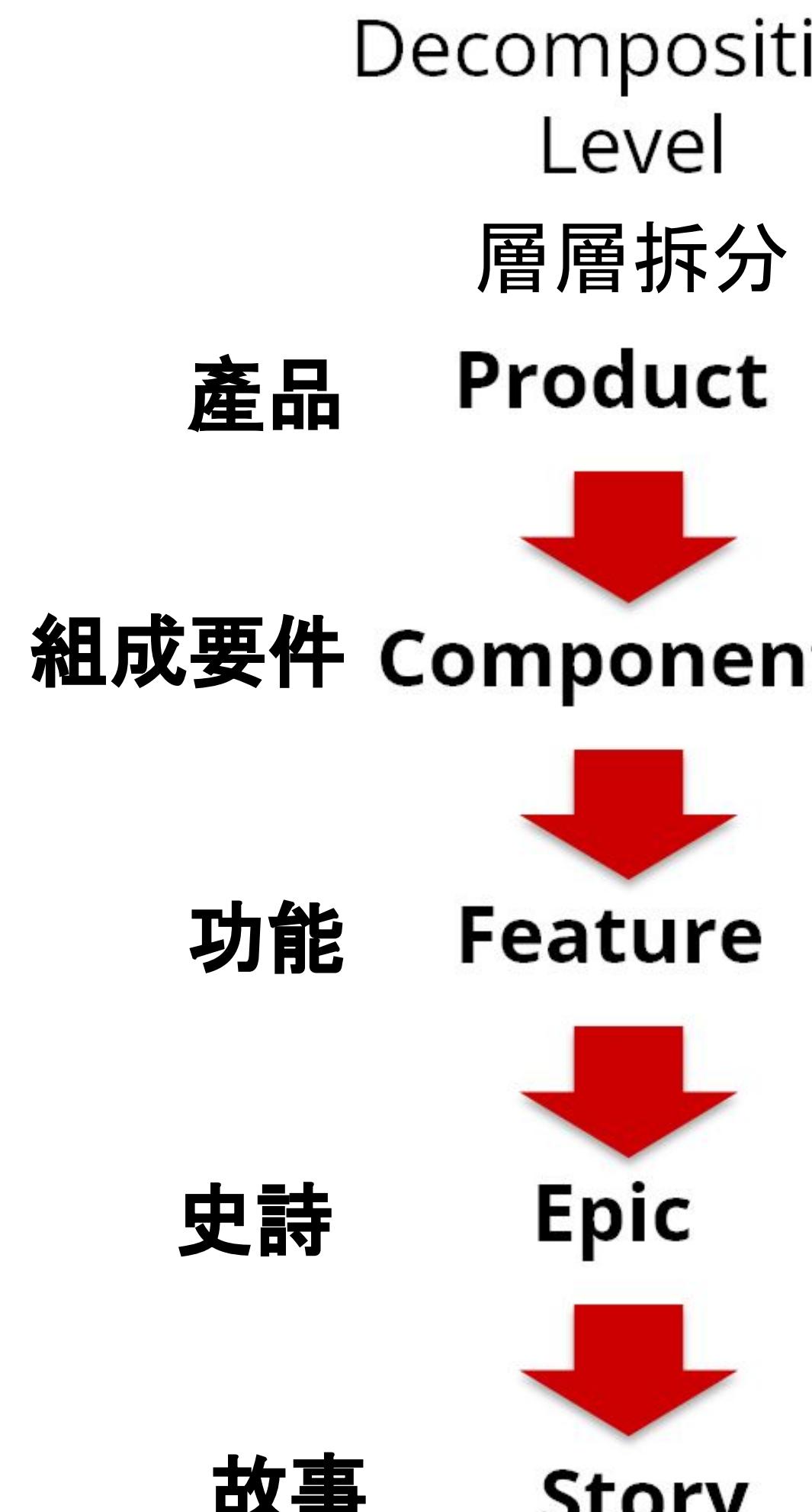


**No more than 10% of the capacity of the Scrum Team**

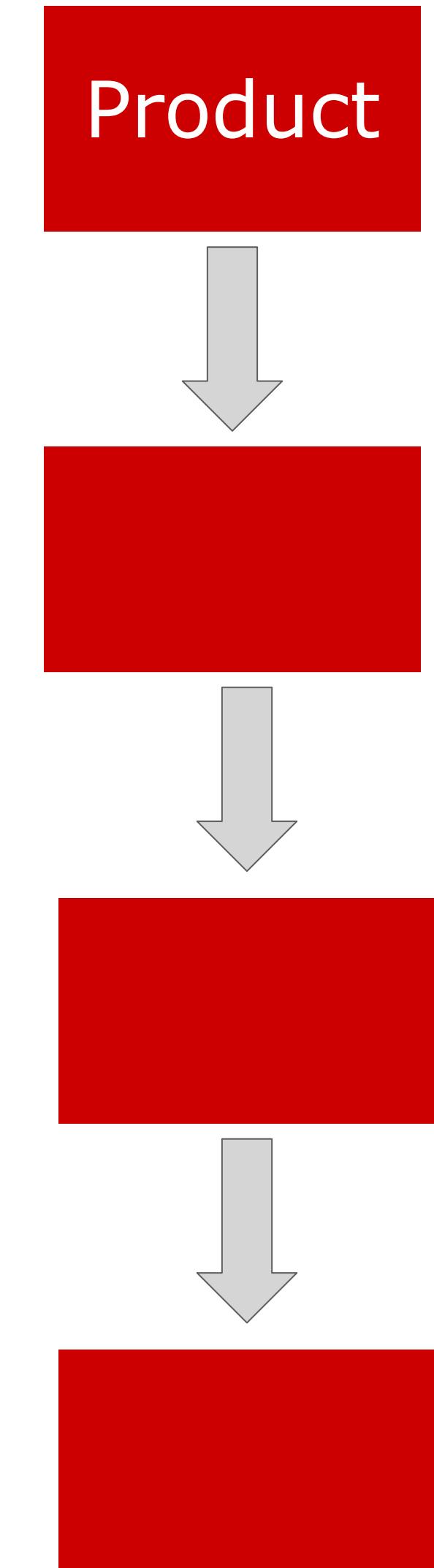
若以一週(工作40小時)為一個Sprint週期，團隊用於Backlog Refinement的時間不超過4小時(10%)

# Maintain a Clear Product Decomposition Hierarchy

## 維持清晰的產品分層結構



# Exercise: What is your Product Hierarchy? 您的產品層次結構是什麼？



# INVEST in good Product Backlog Items

## 好的產品待辦事項的INVEST

- How do we know when a **PBI is ready for the team to bring it in to a Sprint?** Teams use a **Definition of Ready (DoR)**:  
我們如何知道PBI已準備好，可以放進Sprint裏？團隊是用準備好的定義：
  - The INVEST acronym reminds the teams of **good PBIs**  
縮寫字INVEST用來提醒團隊好的待辦事項為何
  - It's like a pidgin language, Customers <- user story -> the Team
  - **I**ndependent 獨立 (**I**mmediately Actionable 可立即採取行動)
  - **N**egotiable 可協商 (not a contract, but open to discussion)
  - **V**aluable 有價值 (to the customer incrementally)
  - **E**stimable 可估算 (by the team)
  - **S**mall enough (for a day or two)
  - **T**estable 可測試 (or it's not clear)



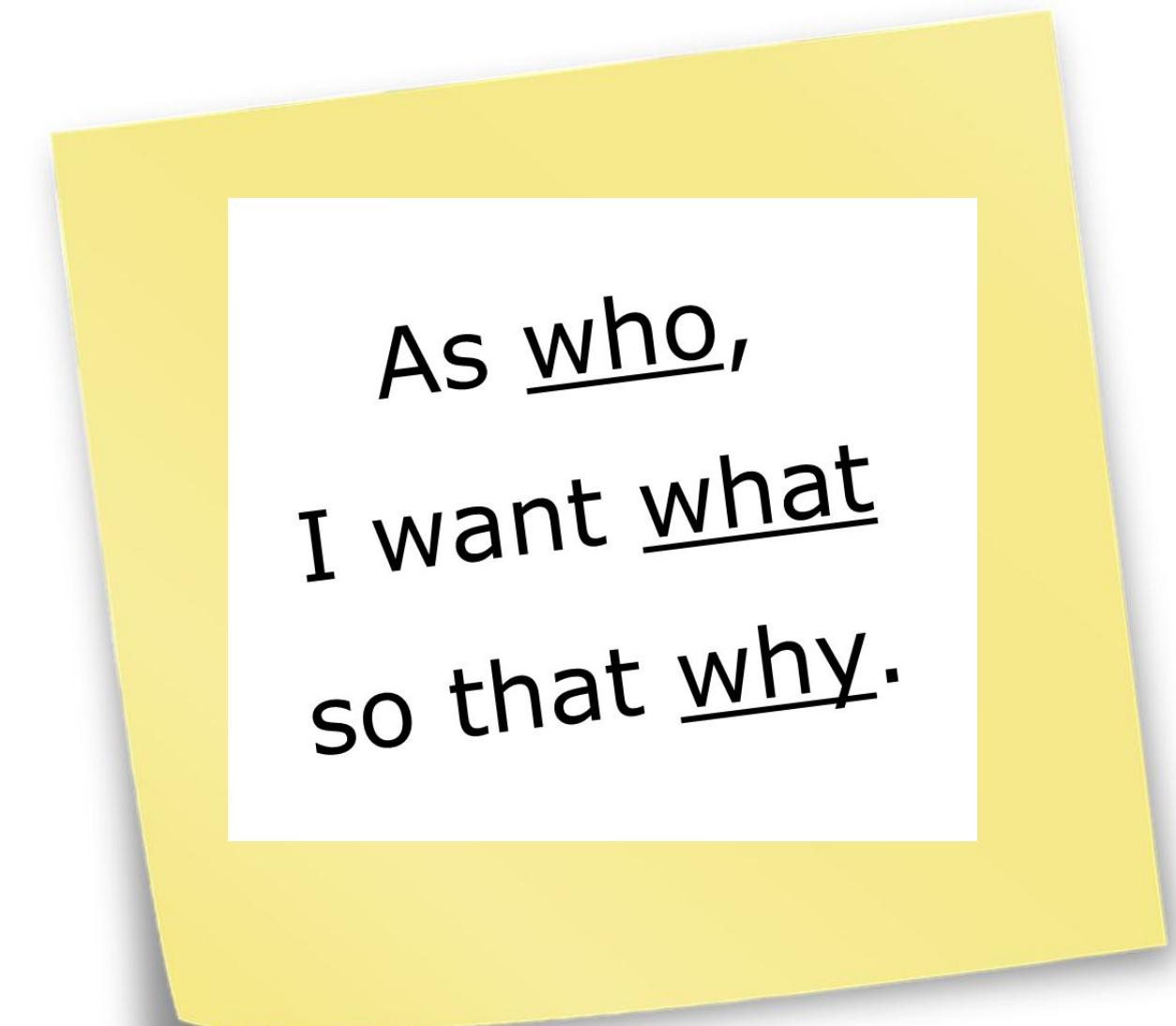
# Diamond has 4 Cs, User Story has 3Cs

- **Card卡片:** or Post-it or electronic tools 或用便利貼或app
  - Just enough information 剛好足夠的資訊
  - Represents the User Story of it 代表其用戶故事
- **Conversation對話溝通:**
  - Between the Customers, PO and Developers (**The “WHAT” part**) 客戶、PO 和開發人員之間(「什麼」的部分)
- **Confirmation確認:**
  - PO can confirm (with the Acceptance Criteria) what has been done is what customer want 由PO確認(藉由驗收標準)做的交付是顧客要的



# User Story Templates 用戶故事撰寫範本

- As a <role> I would like to be able to <action> to achieve <business value>  
身為<角色> 我想要能夠<做什麼>才能達到<商業價值>
- The point of writing user stories like this is to **create a conversation** about what is needed from the end user's perspective that **includes internal acceptance tests**  
要這麼寫用戶故事是要展開終端使用者需求為何的溝通，包含內部的接受測試
- User Story is a way to help customers and the team agree on how to grow a product.  
用戶故事是協助顧客和團隊就如何開發產品而達成共識的一種方法



# Exercise: User Story Writing 用戶故事撰寫練習

Can you write an user story as a **solution for this situation?**

你能寫一個使用者故事來解決這種情況嗎？(5 mins then 2 mins to share)



# Ready Backlog Example 準備好的例子

## Our Default Definition of Ready "DoR"

- Business **Value is clearly** articulated 清楚明確地說明了商業價值
- Details of the **WHAT** are sufficiently understood by the Team  
團隊充分了解要做的細節
- **Dependencies are identified** and no external dependencies exist that would block the story from being completed  
相依性被確認，而且沒有會阻止故事完成的外部相依性
- All **enabling items** are present; Specs, Wireframes etc.  
能有利完成工作的都準備好：規格、線框稿等等
- The team has, or will acquire the **skills to complete the work**  
團隊已經擁有或將獲得完成工作所需的技能
- The Story meets **INVEST** 故事符合INVEST準則, The PBI is sized 估算好
- **Acceptance criteria** are clear and testable 驗收標準明確且可測試
- Performance criteria, if any, are defined and testable如果有效能標準，要清晰且可測試
- Scrum team understands **how to demonstrate** the story **at the Sprint review**  
團隊了解如何在Sprint Review展示會議中展示該故事



# Acceptance Criteria 驗收標準

For a **PBI to be testable**, the **criteria** for being complete **must be available**.

為了使PBI可測試，必須有“完成的驗收標準”

**Acceptance Criteria provide two major benefits** when available as part of the PBI:

驗收標準定義好而成為PBI的一部分時，會帶來兩個主要的好處：

- Promotes test driven development by bringing test planning forward into the design phase, enabling **testing during the Sprint**

藉由將測試推到設計的階段，來促進測試導向的產品開發，以利在Sprint期間的進行測試

- Helps focus the effort on building **just enough functionality**

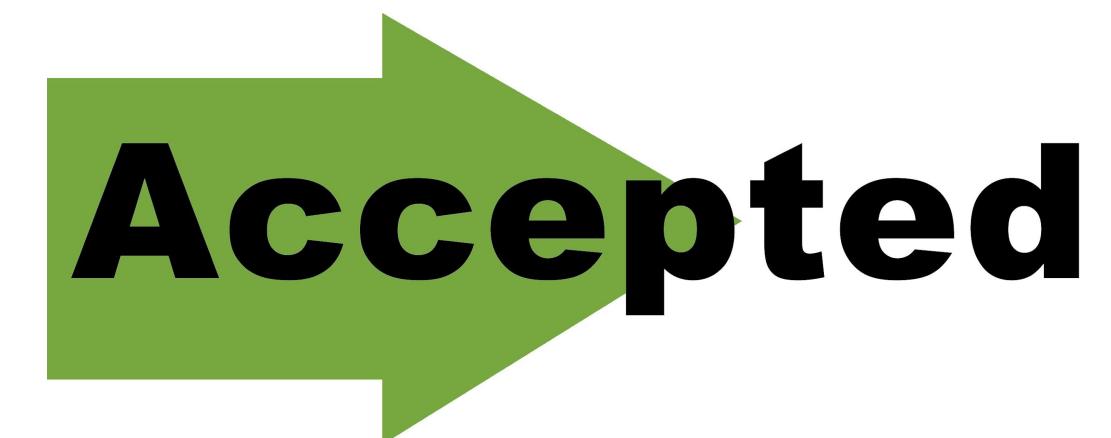
協助專注於打造剛好足夠的功能上



# Type of Acceptance Criteria 驗收標準類型

## 4 Types of Acceptance Criteria (AC): 4 種驗收標準 (AC)

- **Functional:** What is the function looks like for the customers? (Example: when you click on the button, the screen should do XYZ) 功能性: 對客戶來說, 功能是什麼樣的呢? (例如: 當您單擊按鈕時, 螢幕應該做 XYZ)
- **Test conditions:** What should be tested? Did it pass or fail? (Yes or No, there is no Maybe here) 測試條件: 我該測試什麼? 通過還是失敗? (是或否, 這裡沒有也許)
- **What if:** How the function should behave when something goes wrong? (Example: If the transaction failed, what should be displayed on the screen?) 如果假設: 出現問題時應該顯示什麼? (例如: 如果交易失敗, 螢幕上應該顯示什麼?)
- **Non-Functional Requirement:** for example: Security, Scalability, no personal information is stored (encryption), Code Review (Engineer Practice), meeting compliance... 非功能性需求: 例如: 安全性、可擴充性、不儲存個人資訊(加密)、程式碼審查(工程師實踐)、滿足法規...



# Writing Good Acceptance Criteria

**Writing good AC sometimes is more important than writing good User Story 寫好的AC有時比寫好的使用者故事更重要**

What are good acceptance criterias? 什麼是好的驗收標準？

- **Testable:** What do you need to test (usually a list) to make sure it is working the way you expected (the Quality you expected). Example: when you click the button, it will start processing the form 可測試：您需要測試什麼（通常是一個清單）以確保它按照您期望的方式工作（您期望的品質）。範例：當您按一下按鈕時，它將開始處理表單
- **Clear:** Everyone with domain knowledge knows what need to test 清晰：每個擁有領域知識的人都知道需要測試什麼
- **Scenarios:** Create a list of scenarios and the expected behavior. Example: when the form has any missing field, an error message will popup 根據狀況：狀況和預期行為清單。例子：當表單缺少任何欄位時，會彈出錯誤訊息
- **Use the format of your test automation tool:** You write in human readable, and machine interpretable language, and the test can be automated. Example using Gherkin 使用自動化測試工具的格式：用人類可讀且機器可解釋的語言編寫，測試可以自動化。Gherkin 例子

**Scenario:** User click on a button 場景：使用者點擊按鈕

**Given** I am on the Contact Us form page (one or more **Conditions**) 鑑於我位於「聯絡我們」表單頁面（一個或多個條件）

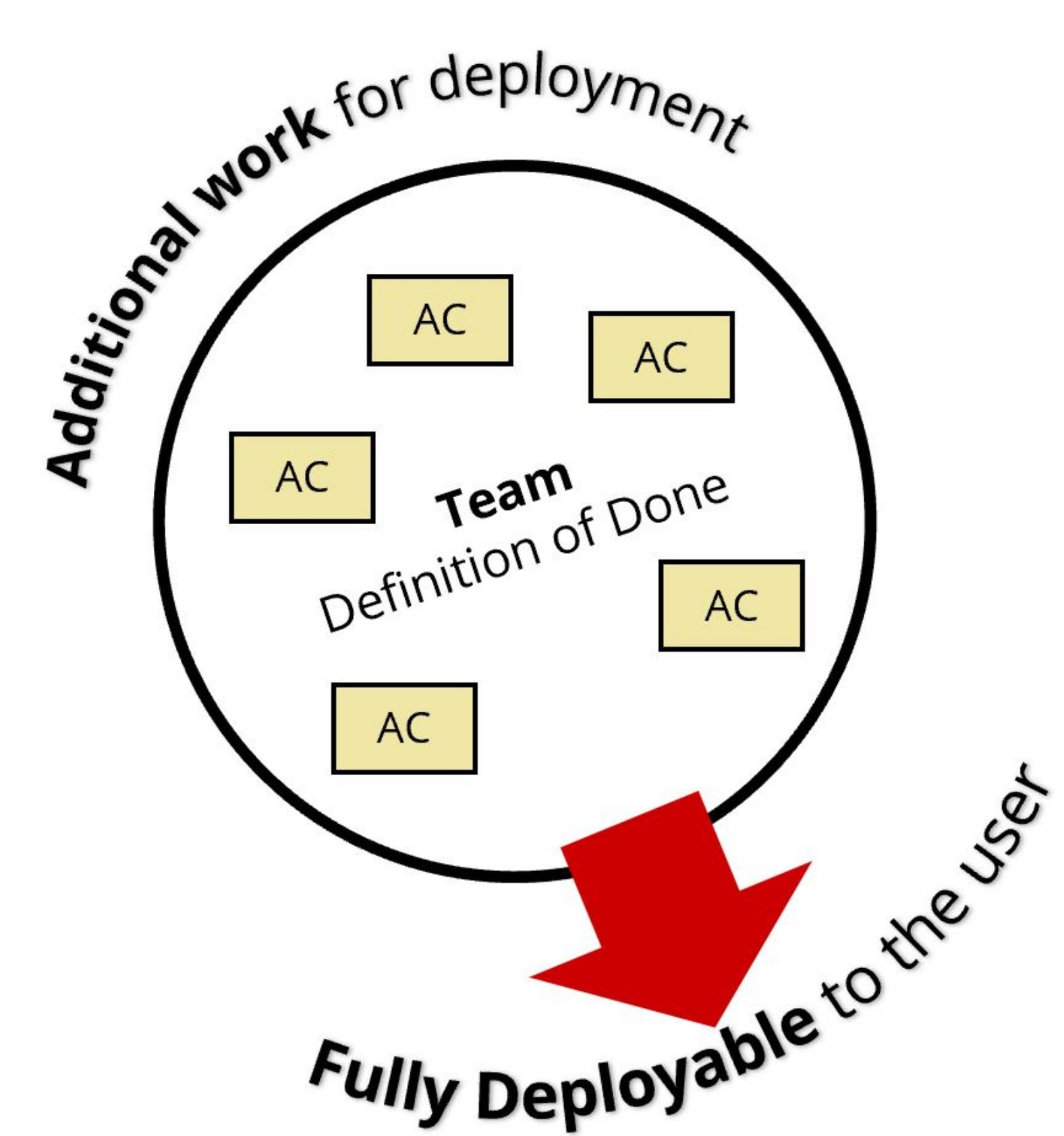
**When** I click on Submit (Actions) 當我單擊提交（操作）時

Then I see Form submitted message (Result) 然後我看到表單已提交訊息（結果）



# Definition of Done 完成的定義

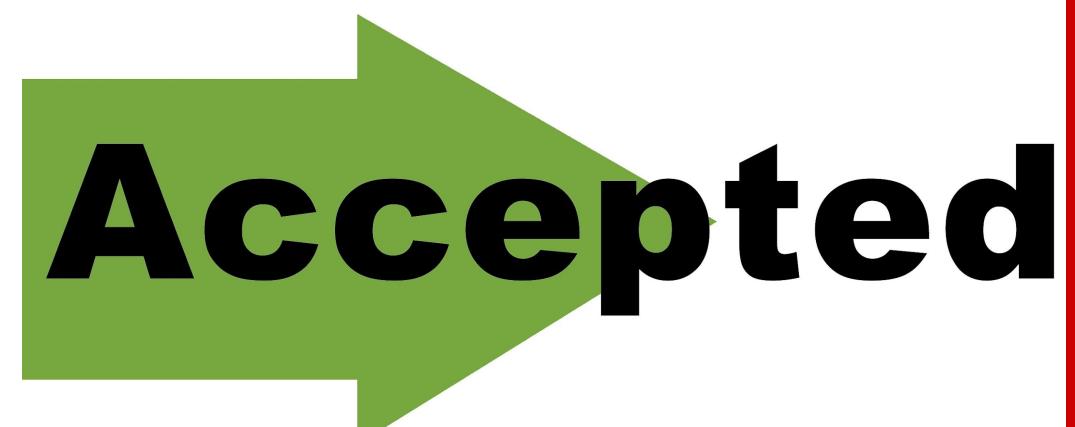
- Definition of Done is in **addition to the Acceptance Criteria (AC)** defined for a Product Backlog Item
- If the organization does not set a **standard Definition of Done (DoD)**, then the Scrum Team should set its own  
若是公司組織未有標準的“完成的定義”，Scrum團隊必須做自己團隊的“完成的定義”
- Often there is additional work before it is fully deployed to the user 在完成交付前，經常會有額外的工作
- You can think of **AC** as what a **customer needs** and the **DoD** as what the **organization needs** in order to deliver it to the end user. 可以這麼想 ~ 為了交付給終端使用者，驗收標準是顧客需要的、完成的定義是組織要的
- The Definition of Done is the **Commitment** for the **Increment**  
完成的定義是對增量的承諾



# Acceptance Criteria vs Definition of Done (DoD)

## 驗收標準 vs 完成的定義

- **Acceptance Criteria**, also known as “conditions of satisfaction” are **applied to an individual Product Backlog Item (PBI)**, and used to confirm that the desired outcome of the PBI is met.  
驗收標準也被視為“滿意的條件”，適用於一個產品待辦事項，也用來確認達成用戶故事的目的
- A clear description or list of outcomes that prove **the story will be acceptable** to the PO that this story is completed to their satisfaction.  
用來證明用戶故事會通過PO驗收的清楚描述或成果清單
- These are typically **written by the PO**, but the team can help, and often do if it is a Kaizen or Technical Debt story.  
驗收標準是由PO填寫，團隊可以協助，經常會這麼做當遇到改善或技術債的用戶故事
- **DoD applies to all Product Backlog Items in a product.** DoD適用於某一產品的所有產品待辦，it can be thought of as **a master set of standards** that can be applied to **division or company wide**. 可以想成是一組標準適用於各單位或全公司
- DoD is a formal description of the state of the Increment when it meets the quality measures required for the Product. (Think of it as a quality mark when shipping the completed stories) (可想成是完成交付的品質標誌)



# Small, but How? a few techniques you can use

- Draw a **workflow diagram** (can you break it by implement part of the workflow (that still deliver end-to -end)?
- Can you break it by **CRUD** (Create, Read, Update, Delete)? When the request involves CRUD, break it by each one)
- Can you break it by **Paths** (Different Channels)? (pay by cash or credit card)
- Can you break it by **Interface**? (Web or iOS Android)
- Can you break it by **Data**? (Look at your Data Model, can we do one type of data a time? Or if the ticket is about ETL, do you need to ETL all types of data on Day One?)
- Can you break it by **Business Rules**? (Relaxing the Rule, first implement no limit to buy tickets, then limit it to like 5 a day)
- Can you break it by **Users**? (start with group of with most users, example: Right-Handed, English Spoken, Business Users...)
- Can you break it by **Simplicity** (Agile Principle)? (build the MVP first, most common needs first, also remember: customer doesn't know what they want until you show them), think this question: **Is it needed on Day One?**
- Think this question: **What will the customer want to see first?**
- Totally new: **Do a Spike**

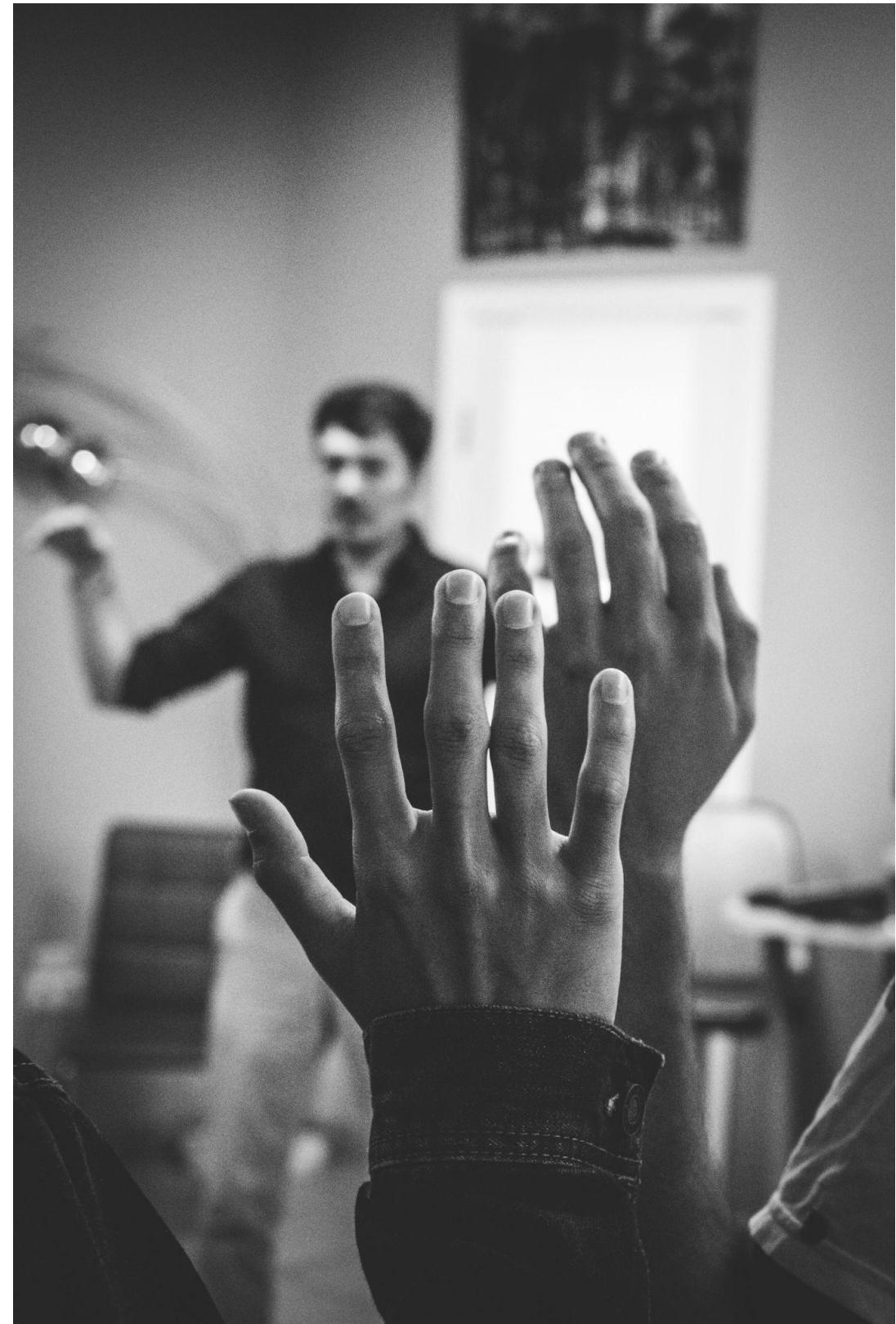
Notice:

- Build the least effort with most needs first
- “Can it scale?” is a great question to think about, break your story into “Make it Work” and “Make it Fast”
- There is a slightly difference between slicing a Feature vs slicing an Epic? Example: When slice a Feature, you work on most valuable feature first and leave “Nice to Have” for later. However, when slice an Epic, you break it into a few PBIs, each PBI is part of the original “function” and only when you are done with most of these small PBIs, the function will show its value.

# Key Points

- **Refinement** is a conversation between the Product Owner and the Scrum Team to ensure everyone understands **WHAT** is needed to complete the PBIs 優化是產品負責人和Scrum團隊之間的對話，以確保每個人都了解完成 PBI 需要什麼
- **Up to 10% of a team's time** each Sprint should be dedicated to Refinement 每個Sprint，團隊應將最多 10% 的時間用於優化
- You **do the Estimation during Refinement** (so that Planning is focused on the Sprint Goal and Commitment, and Planning will be a short event) 在優化活動進行估算(規劃活動重點關注在Sprint目標)
- A good PBI should be I.N.V.E.S.T. 一個好的 PBI 應該是 I.N.V.E.S.T。
- Refinement is about make PBIs “**READY**” 優化是為了讓 PBI “**準備就緒**”

# Questions?



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

# Relative Sizing 相對大小 (\*)

As a member of a Scrum Team, I need to know why and how to estimate Product Backlog Items with leading practices, so that I can help my team and organization predictability and pull the right amount of work into a Sprint and measure if we are getting faster  
身為Scrum團隊的一員，我需要知道為什麼以及如何透過領先作法來估算產品待辦事項清單項，以便我可以幫助我的團隊和組織實現可預測性，並將適量的工作放入Sprint 中，並衡量我們是否越來越快

# Learning Objectives:

- Discuss why **only those doing the work estimate it** 討論為什麼只有做的人才能對其進行估算
- Recall at least **two empirically supported reasons** for why estimating in **points is better than estimating in time** 回想一下至少兩個有經驗支持的理由來說明為什麼按點數進行估算比按時間進行估算更好
- Describe **relative size estimation**, which may include estimates of effort or value 描述相對估算, 其中包括工作量或價值的估算
- Identify at least two methods for estimating PBIs with points (e.g., **Estimation Cards and affinity estimation**) 至少兩種用點數估計PBIs的方法(例如估算卡片和近似估算)
- **Practice** estimating PBIs **using points** 練習使用點數估算PBIs

# Exercise: To Estimate or Not to Estimate

- Form two groups 分成兩組
- Group A: **Yes, we need to Estimate** A組:是的，我們需要估算
- Group B: **No, we do NOT need to estimate** B組:不，我們不需要估計
- 5 mins group discussion to **list your supportive viewpoints** 5分鐘小組討論，列出您的支持觀點
- 5 mins one person for each group to **present your viewpoints** 每組5分鐘 一人陳述你們的觀點

Notes for trainer:

- Put “Yes” and “No” on two whiteboards
- Each group writes their viewpoints on their whiteboards



# No Need to Estimate, because...

- It's **always wrong** 總是錯
  - Question: do we know **how to estimate?** 我們知道如何估算嗎？
  - Is the **PBI READY?** PBI 準備好了嗎？(are we **estimate on partial information?**) 我們是根據部分資訊進行估算嗎？
- It takes **a lot of time** 花很多時間
  - Is it **a lot of time discussing the details** of the PBI or a lot of time on estimation or both? 是花很多時間討論PBI的細節，還是花很多時間估算，還是兩者？
- What are the other benefits from the Group B? B組還有哪些好處？
- Because of the above, we believe there is no value in Estimation 由於上述原因，我們認為估算沒有價值

However, 然而

- How will we know **when we can deliver?** 我們如何知道何時可以交付？(if we don't estimate)(如果我們不估算的話)
- **How can we improve?** 我們該如何改善？(if we don't know our velocity)(如果我們不知道我們的速率)

## Conclusion: We need to Estimate 結論:我們需要估計

# Traditional Project Estimation Error

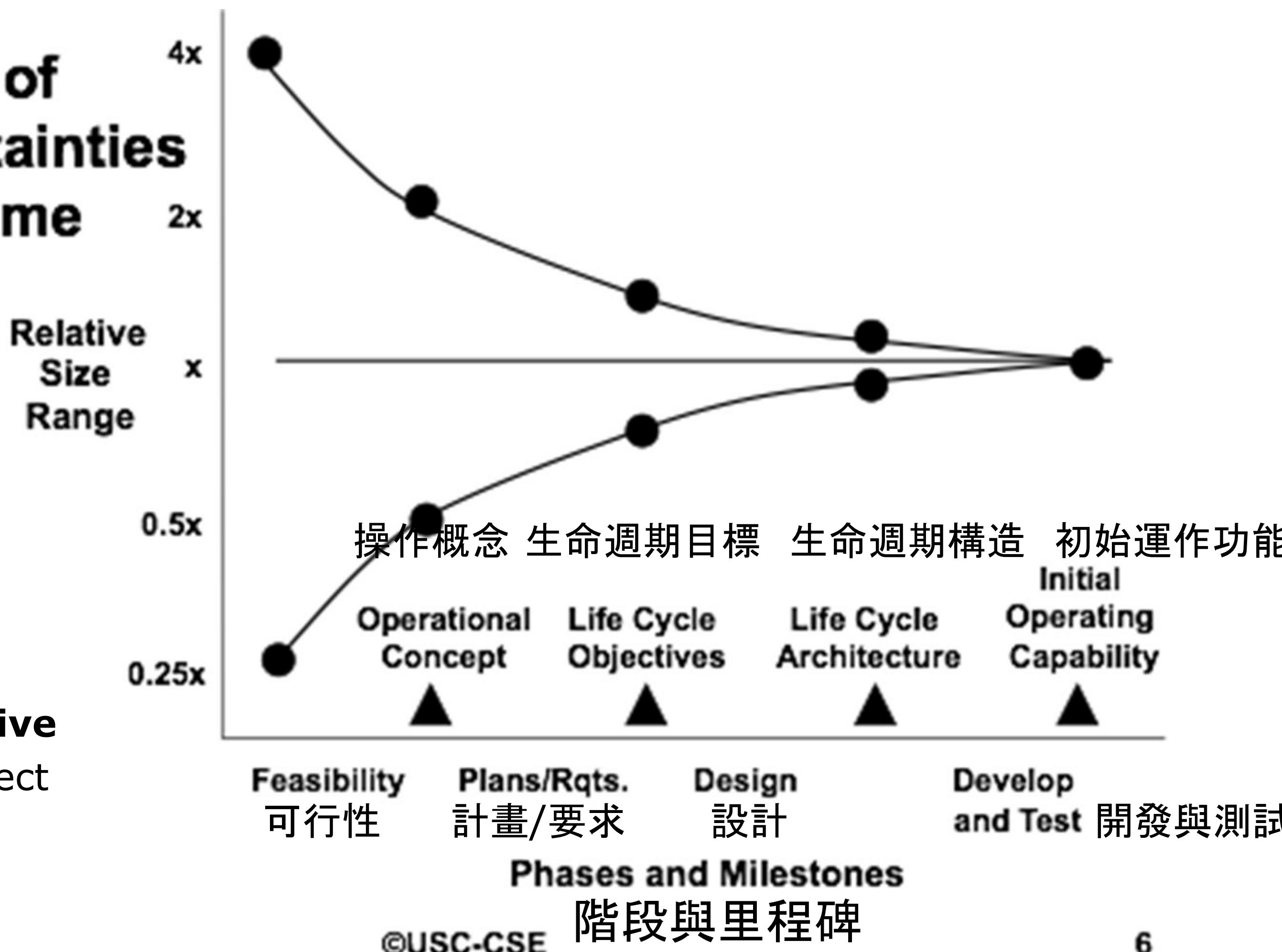
## 傳統專案估算誤差

不確定性效應  
(隨著時間)

- **Effect of uncertainties over time**

相對尺寸範圍

We know the **least about an initiative at its start** and the accuracy of project estimates tends to improve as more information becomes available

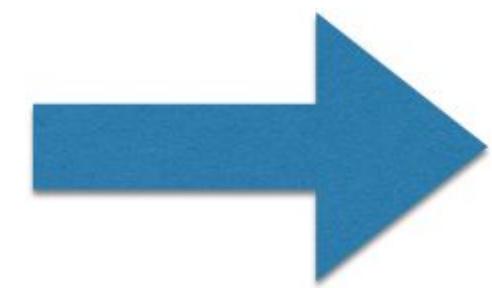


6

Barry Boehm. Cost Estimation with COCOMO II. CS 577a, University of Southern California, Center for Software Engineering. Fall 2006

# The Reason We Estimate 估算的原因

Why do we estimate?



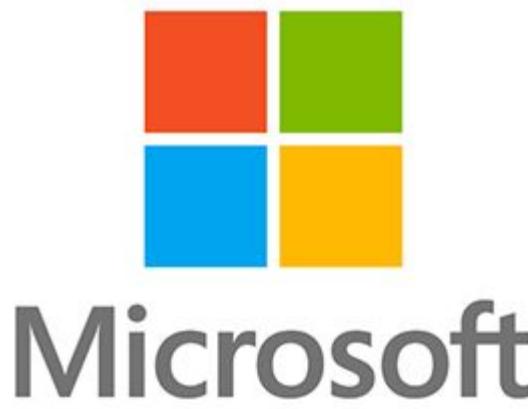
In order to determine our **Velocity**, so  
**that we know when it will be done**  
**(Predictability 可預測)**  
為了估算我們團隊的速率  
and how much it will cost

**Question: How can we run faster?**

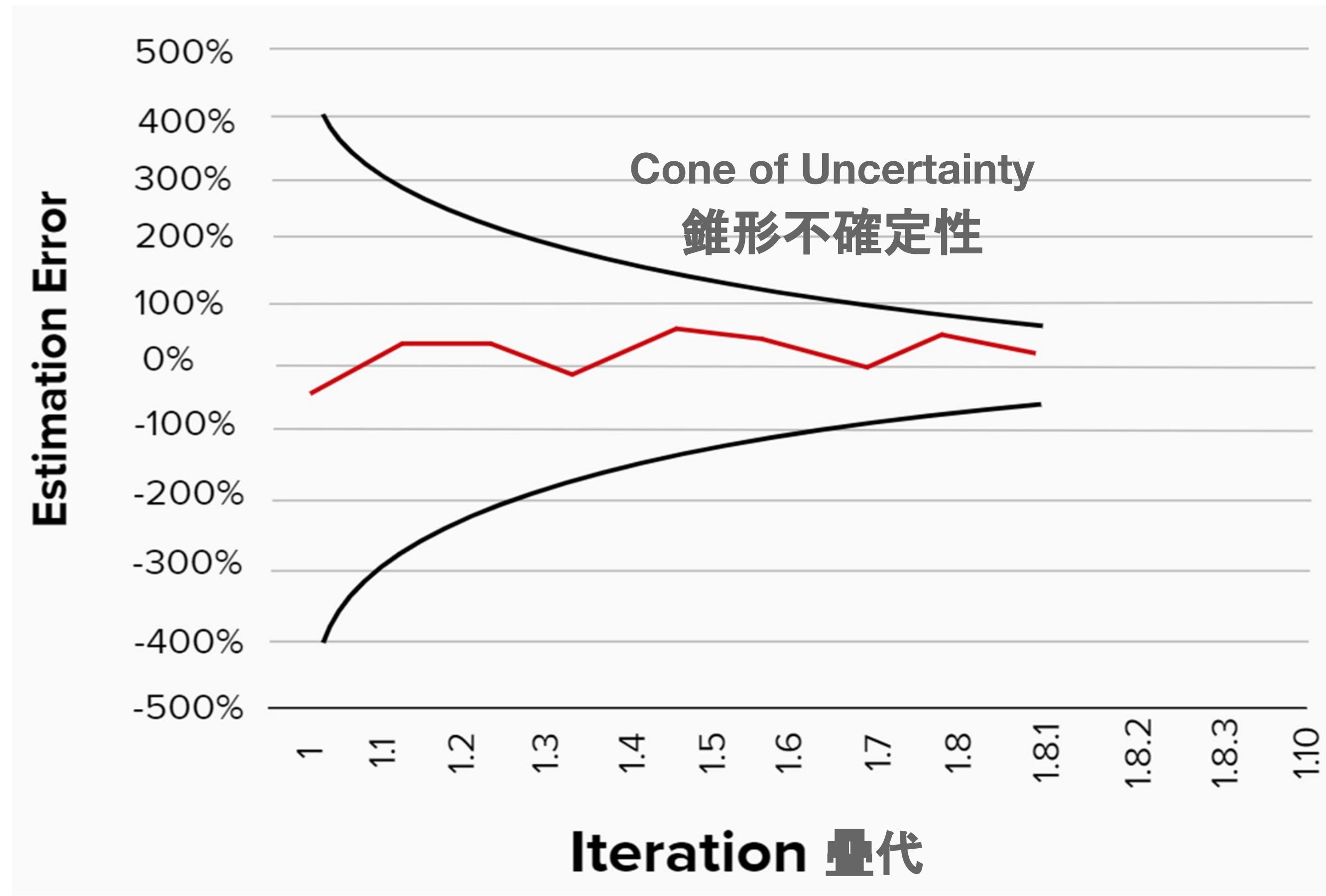
**Answer: Remove the Impediments**

# Case Study: Estimation Accuracy at Microsoft

Points are more accurate 採用點數估算更準確



估算錯誤



Black Line - Cone of Uncertainty (The old way)

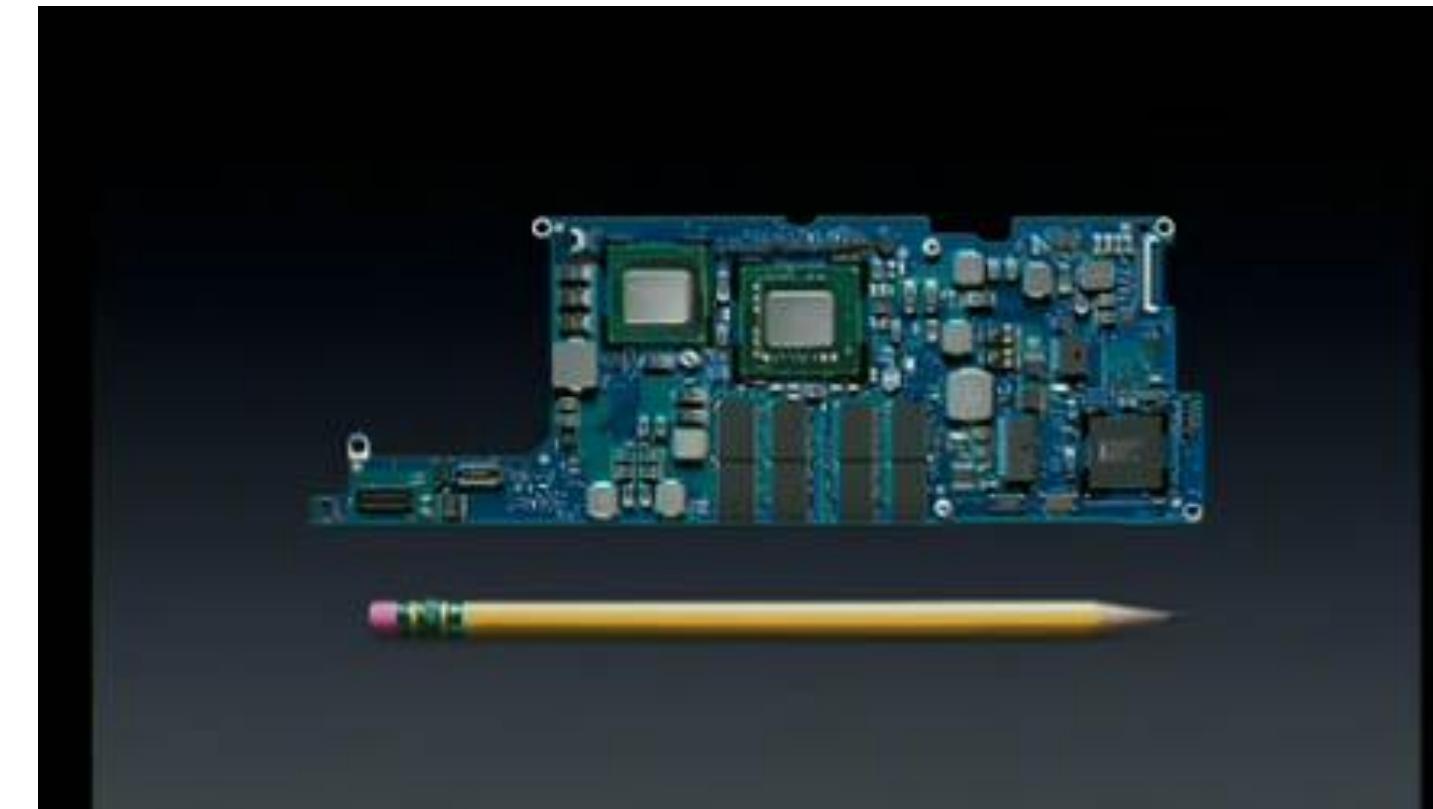
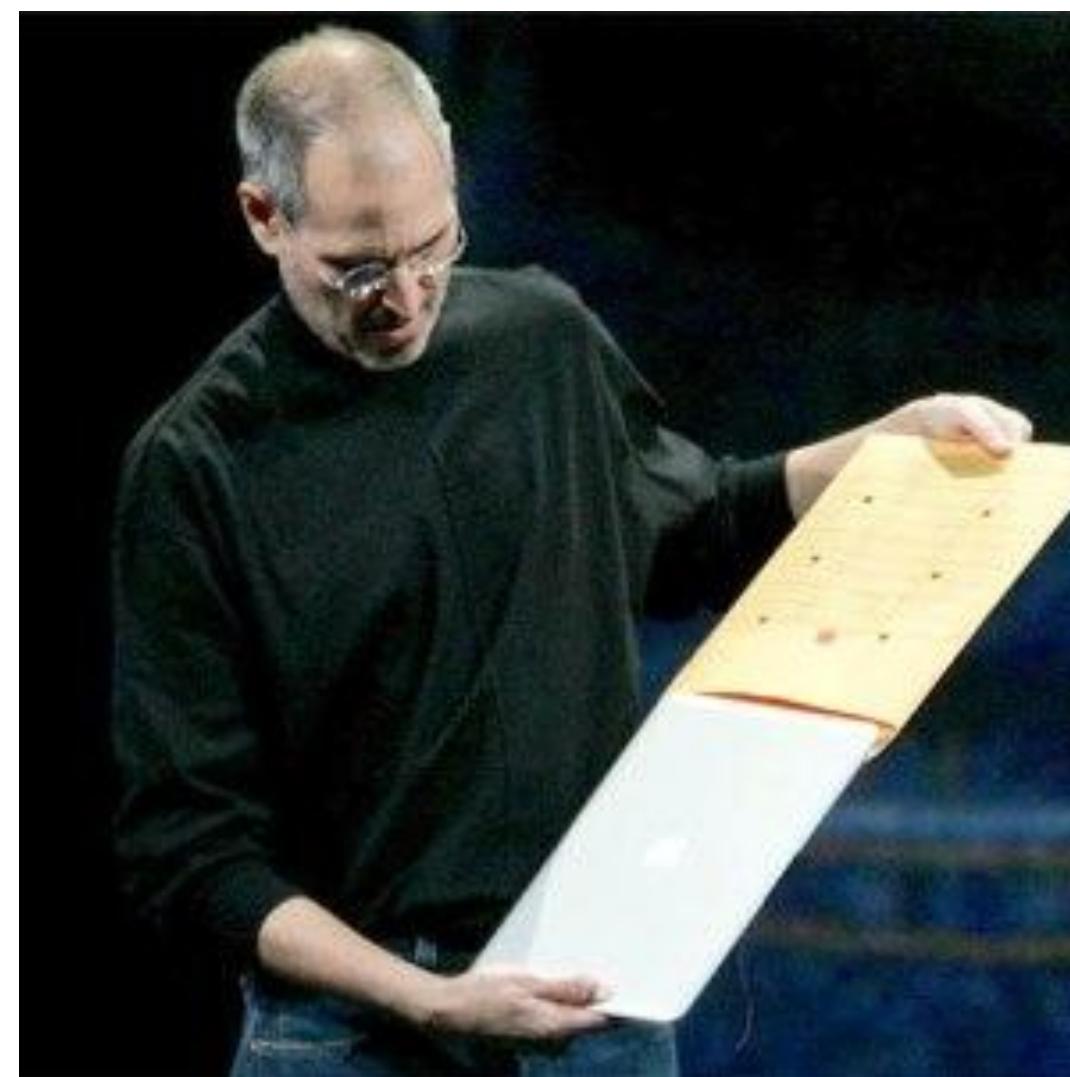
Red Line - Points

Laurie Williams, Gabe Brown, Adam Meltzer,  
Nachiappan Nagappan, 2011, Scrum +  
Engineering Practices: Experiences of Three  
Microsoft Teams  
IEEE Best Industry Paper Award, 2011  
International Symposium on Empirical Software  
Engineering and Measurement

# It's Relative 相對的

When you want to tell people how big something is, you **compare it with something we are all familiar with**

要告訴別人某個東西的大小，你就拿它和大家熟悉的東西做比較



**Can you guess how big is the 3rd item?**

# How big is it?



**Now, that is easy**

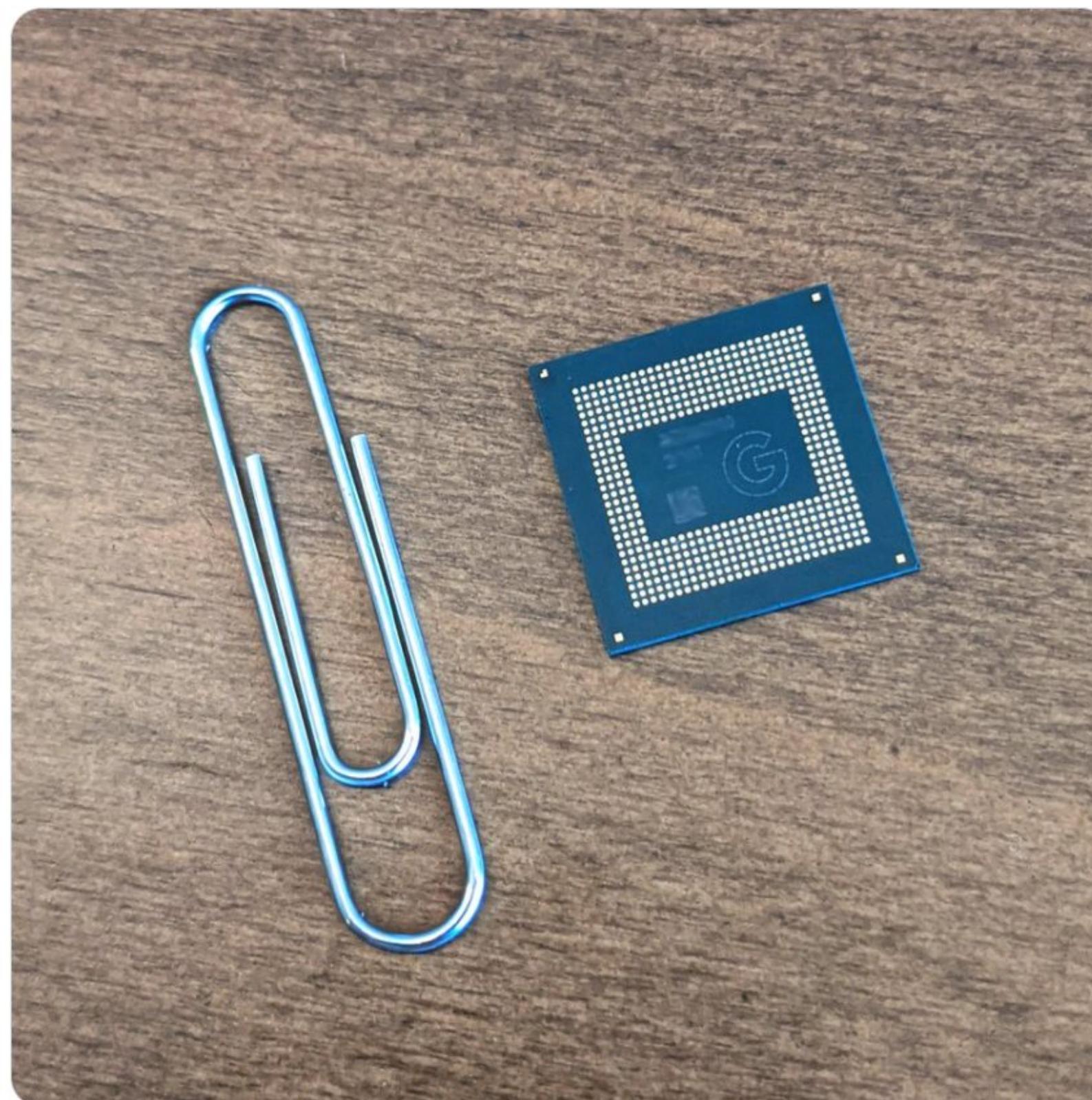
# Another example from Google



Sundar Pichai   
@sundarpichai

...

So excited to share our new custom Google Tensor chip, which has been 4 yrs in the making ( for scale)! Tensor builds off of our 2 decades of computing experience and it's our biggest innovation in Pixel to date. Will be on Pixel 6 + Pixel 6 Pro in fall.  
[blog.google/products/pixel...](http://blog.google/products/pixel...)



**Why he puts a paper clip there?  
為什麼在旁邊放迴紋針？**

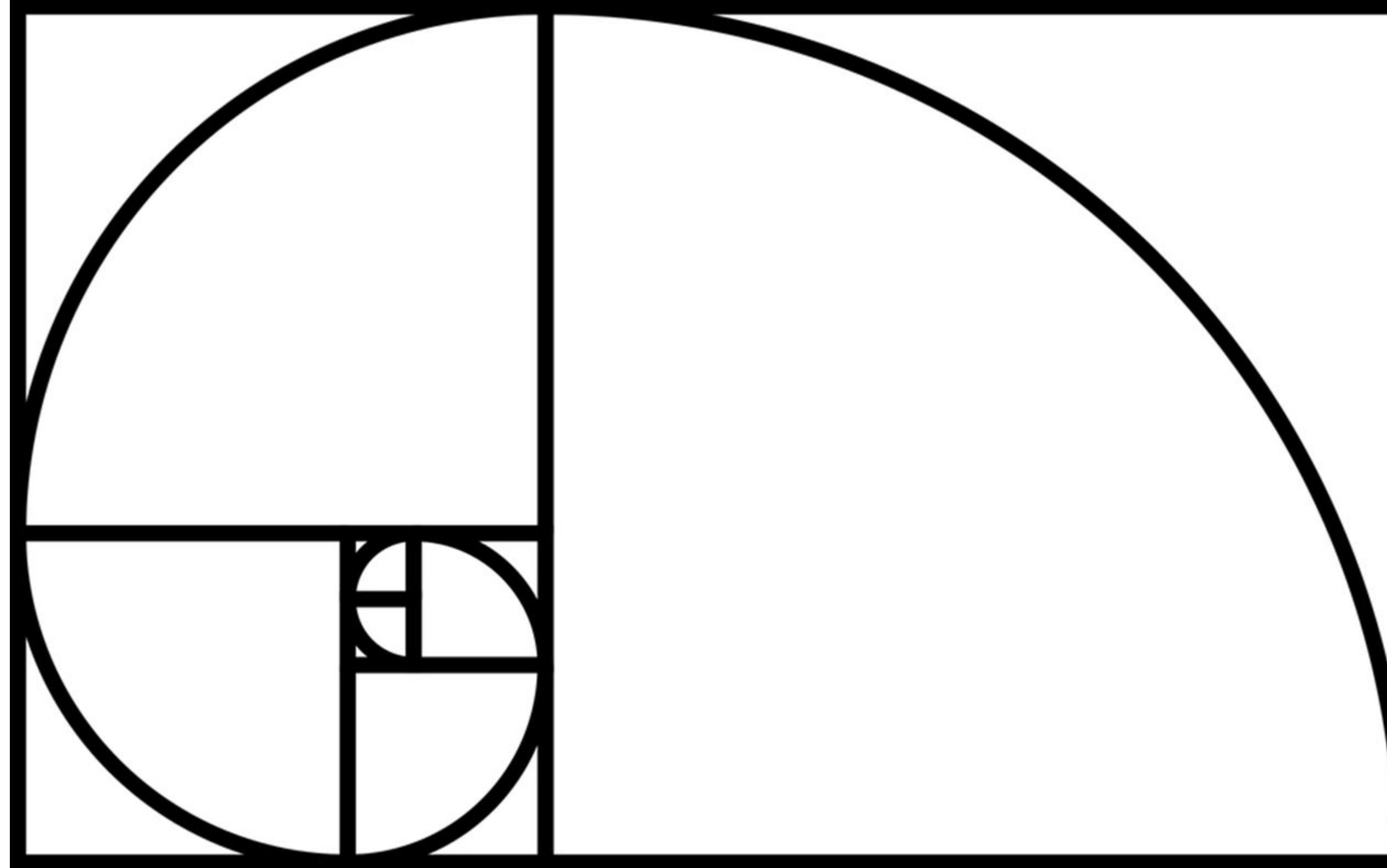
**You know Relative Estimation  
already, and it's even more intuitive  
您原本就知道相對估算, 而且它更加直觀**

# Fibonacci Sequence... the Golden Ratio

## 斐波那契數列 ... 黃金比例



# Fibonacci Numbers 斐波那契數



$$0+1=1$$

$$1+1=2$$

$$1+2=3$$

$$2+3=5$$

$$3+5=8$$

$$5+8=13$$

$$8+13=21$$

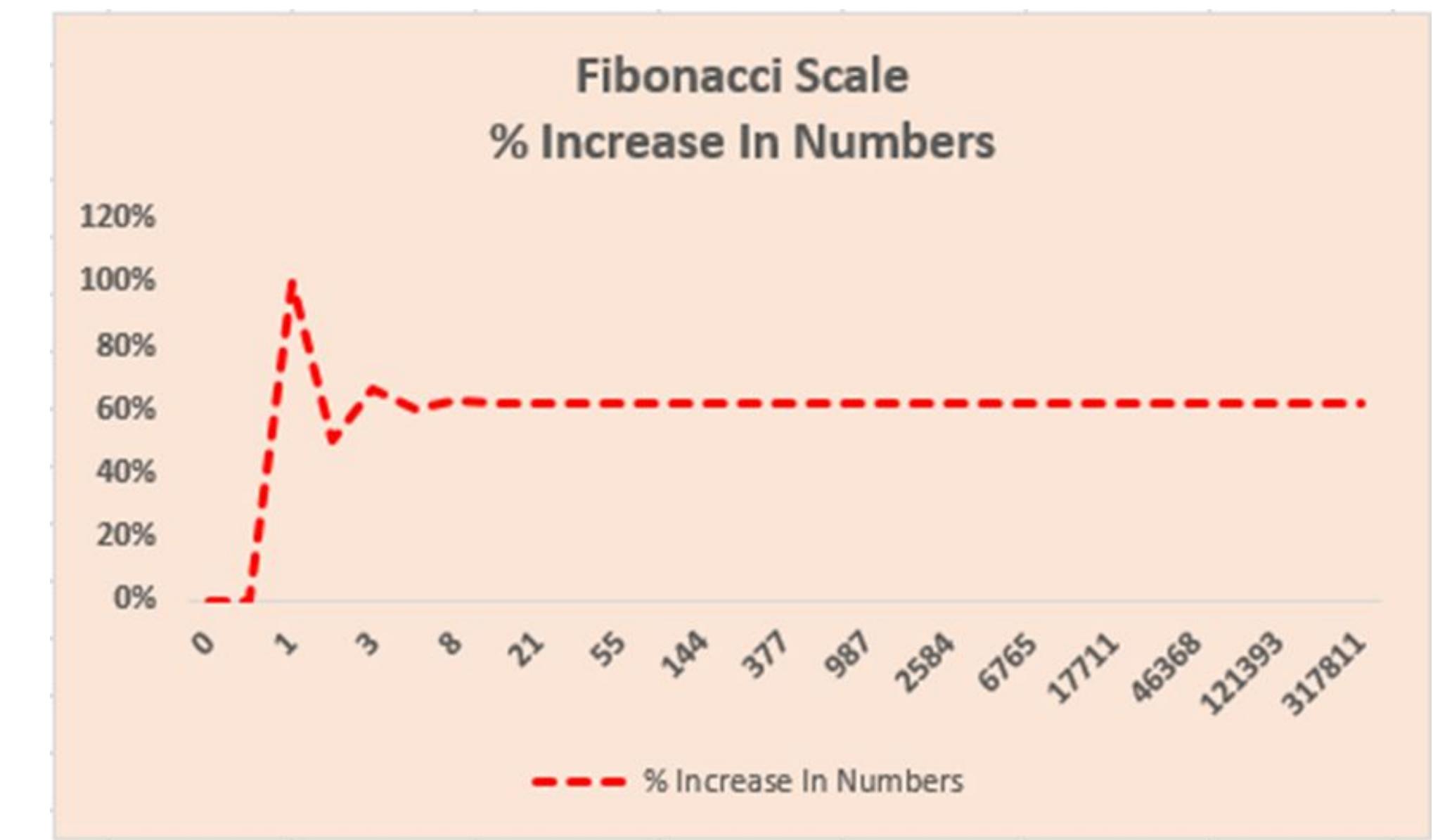
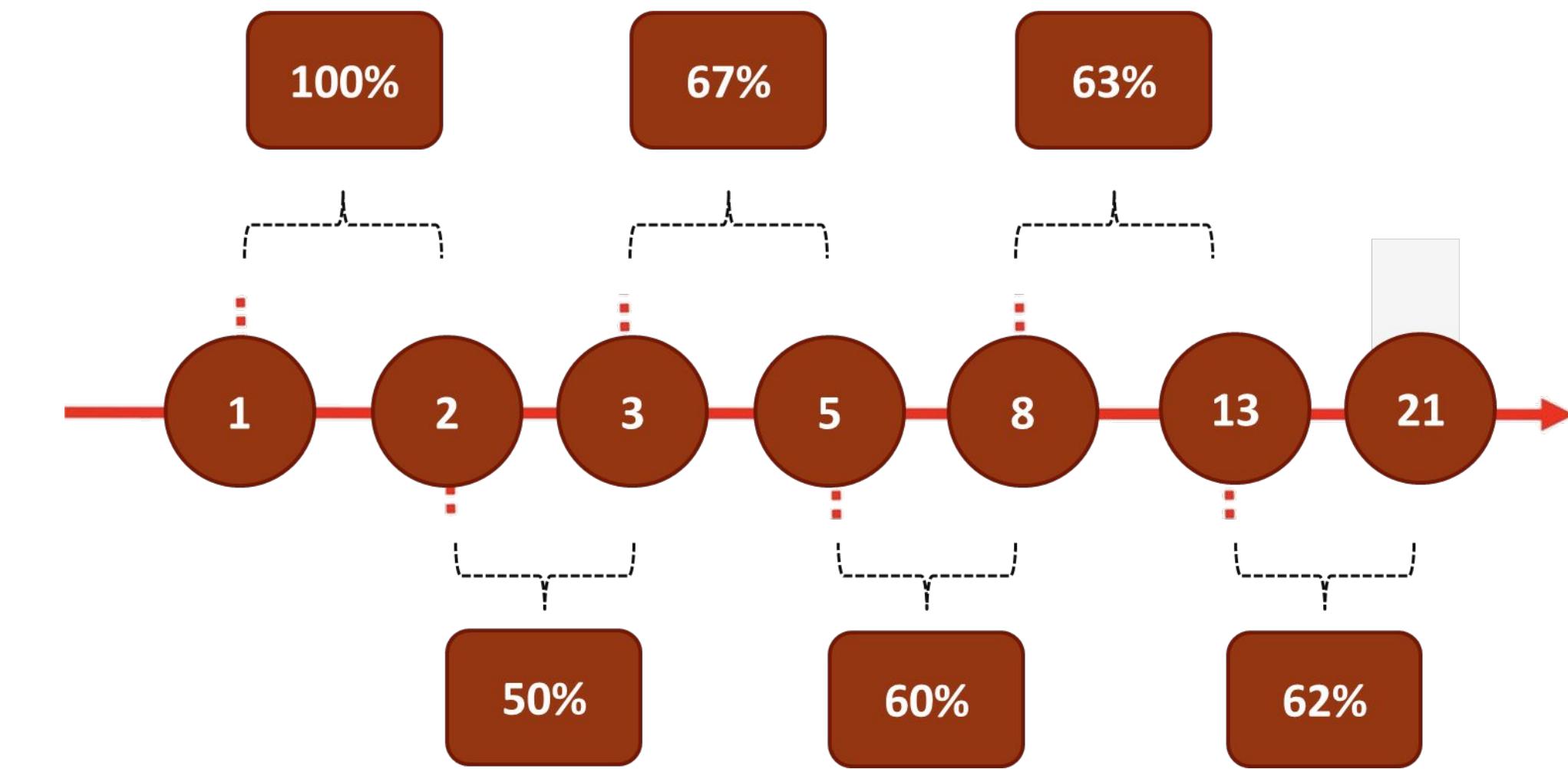
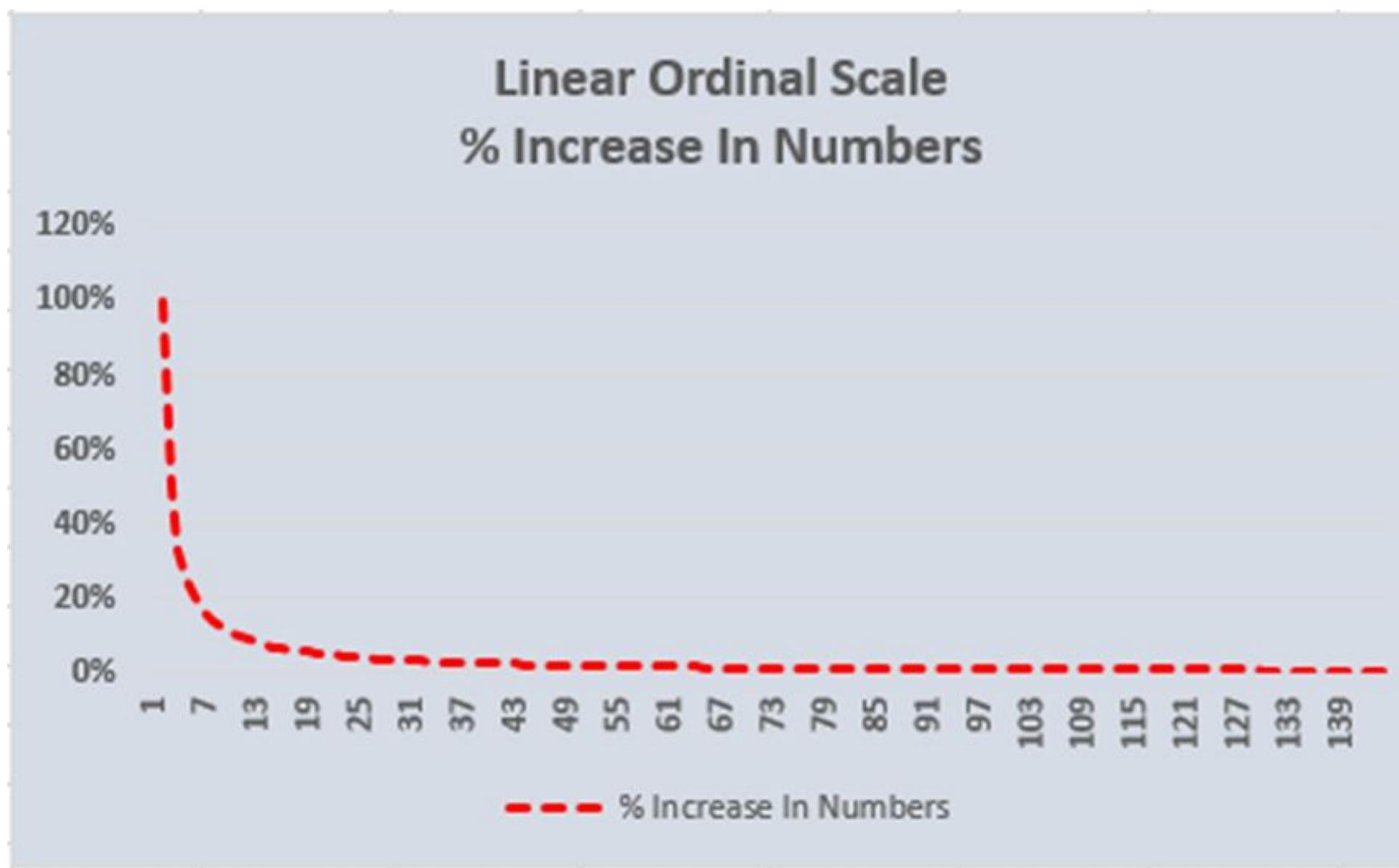
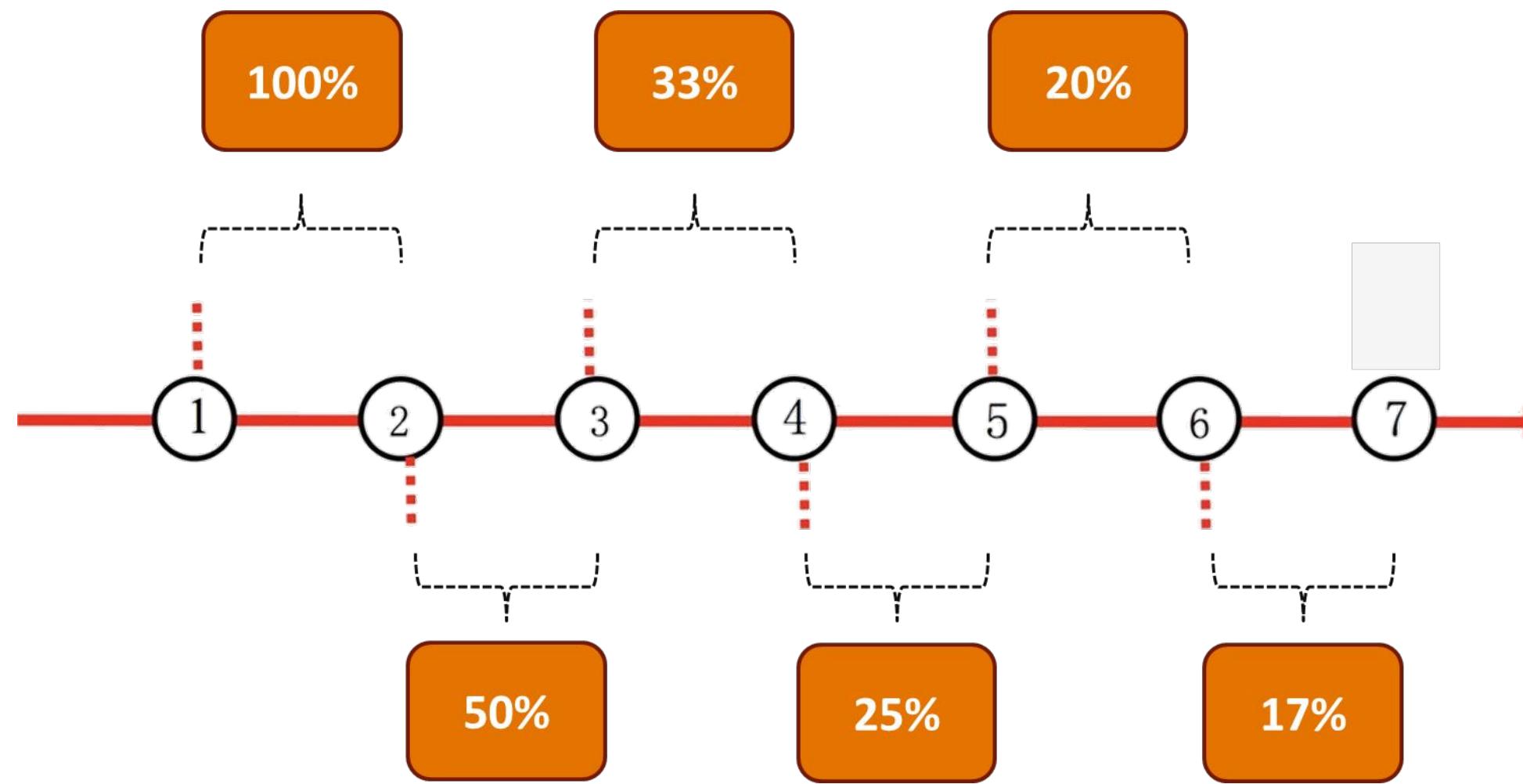
$$13+21=34$$

$$21+34=55$$

...

# Measuring Effort: Linear vs Fibonacci

計算成果：線性 vs 斐波那契



# Exercise: Hours vs Points

We agreed that we should Estimate, the next question is: Use Hours or Points? 我們同意應該估算, 下一個問題是:使用小時還是點數?

- Form two groups 分成兩組
- Group A: **Let's estimate in Hours** A組:讓我們以小時為單位進行估算
- Group B: **Let's estimate in Points** B組:讓我們以點數為單位進行估算
- 5 mins group discussion to **list your supportive viewpoints** 5分鐘小組討論, 列出您的支持觀點
- 5 mins one person for each group to **present** your viewpoints 每組5分鐘 一人陳述你們的觀點

Notes for trainer:

- Put "Hours" and "Points" on two whiteboards
- Each group writes their viewpoints on their whiteboards



# Estimate using Hours 小時為單位進行估算

- It's **intuitive and easy**, everyone knows how to do this 直觀且簡單, 每個人都會
- **Hours can be measured**; we know when we can deliver 可以測量小時數; 我們知道什麼時候可以交付
- What else are the benefits of estimate using hours? 使用時間估算還有什麼好處?

**However, What if (and Why) 3 hours turn into 3 days?** 但是, 如果3小時變成3天呢?

- The PBI is not READY, still missing details, so you are estimating on partial information PBI尚未準備好, 仍然缺少詳細信息, 因此您正在估計部分信息
- **Our work is complex**, it's never as simple as we expected 我們的工作很複雜, 從來沒有我們想像的那麼簡單
- Interrupts coming, we need to **switch between this ticket and another tickets** (so you think it's 3 hours of no interruption) 插件一到來, 我們需要在這PBI和另一PBI之間切換 ((所以你認為這是3小時不間斷))
- Now, no one is happy **because it has been 3 days...** 現在, 沒有人高興, **因為已經三天了...**
- Your hours vs my hours 你來做需要的時間 vs 我來做需要的時間
- Hours estimation can work if your work is Simple 如果您的工作很簡單, 小時估算就可以發揮作用
- It is always 40 hrs per week, **the only way to improve is: Increase the hours** 每週總是 40 小時, 唯一的改善方法是: **增加工作時間**

**Conclusion: Estimate using Points, NEVER use Hours**

**結論: 使用點數估算, 切勿使用小時數**

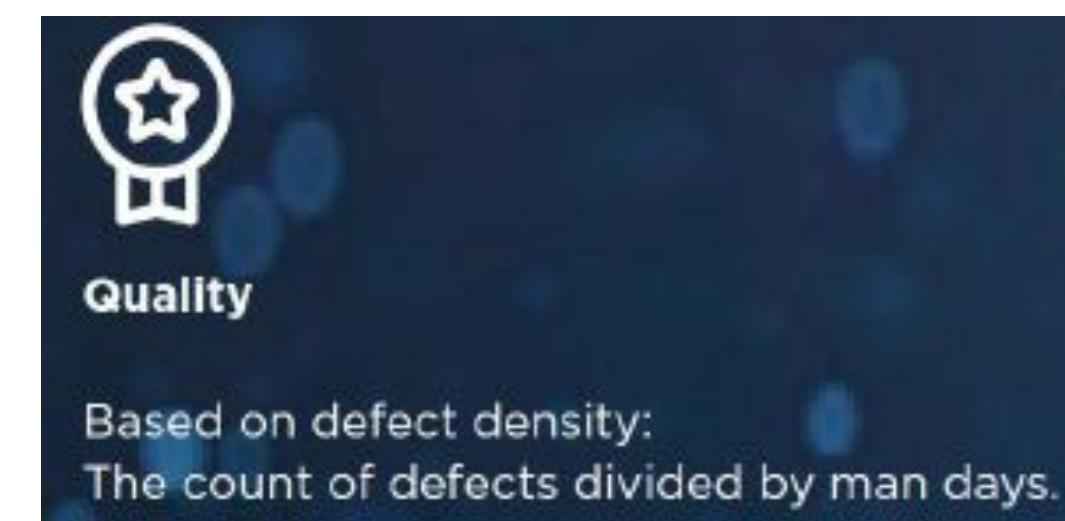
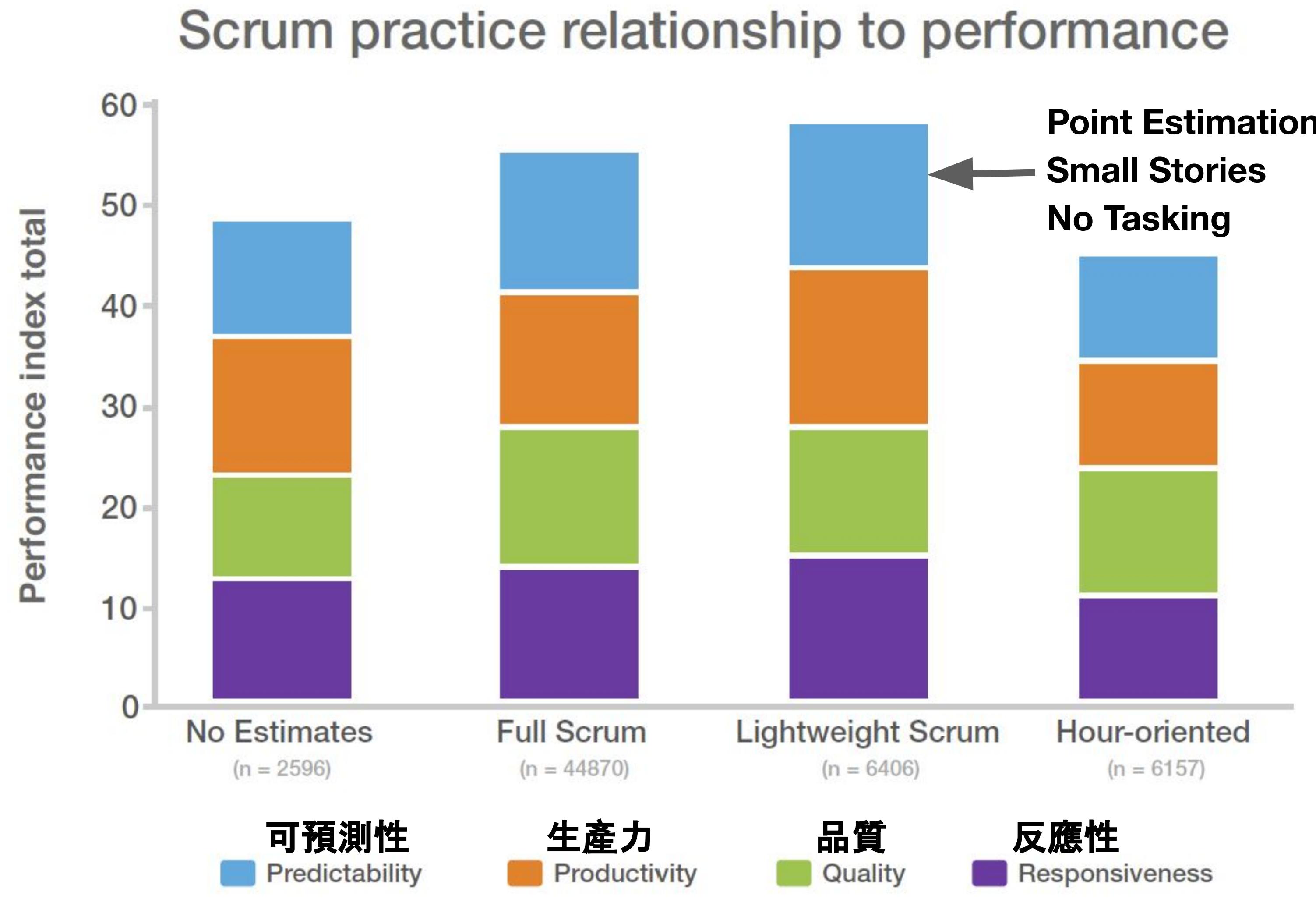
# The point is not about the (story) points

## 重點不在於(故事)點

- Relative Estimation helps us **to be real**, and more **confident knowing we can deliver what we committed** 相對估算幫助我們更真實、更有自信地知道我們能夠兌現我們的承諾
- Relative Estimation helps us to be **consistent** 相對估算有助於我們保持一致
- Relative Estimation helps us to **remain flexible** 相對估算幫助我們保持靈活性
- Relative Estimation enables us to **be able to improve** 相對估算使我們能夠改進
- If your work is simple (like shoving the snow on the driveway), you don't need story points, hours will work for you 如果您的工作很簡單(例如在車道上鏟雪), 則不需要故事點, 小時為單位進行估算很有效
- the point of estimation in points is **the effect of the process**, because estimation is at the end of the process, the team will need to have the discussion to make sure we capture the information needed, we are clear about what we need to do and what we need to test, then the estimation will be more real, **these processes all together helps the quality of what we deliver** 點數估算的重點在於過程的效果, 因為估算是過程的最後, 團隊需要進行討論以確保我們捕獲所需的信息, 我們清楚我們需要做什麼, 並且我們需要測試什麼, 那麼估算就會更加真實, **這些過程一起有助於我們交付的品質**
- I think that is the **main reason that No Estimation and Hours Estimation have the worse quality** and **Estimation in Points give best quality** (250% difference) 我認為這是「不需要估計」和「小時估算」品質較差, 而「點數估算」品質最好的主要原因(250% 的差異)

# Case Study: Rally Data Mining 資料探勘 of 60,000 Teams

## Small Stories vs. Tasking vs. Hours 小故事 vs. 任務 vs. 小時



©2013 Rally Software Development Corp.

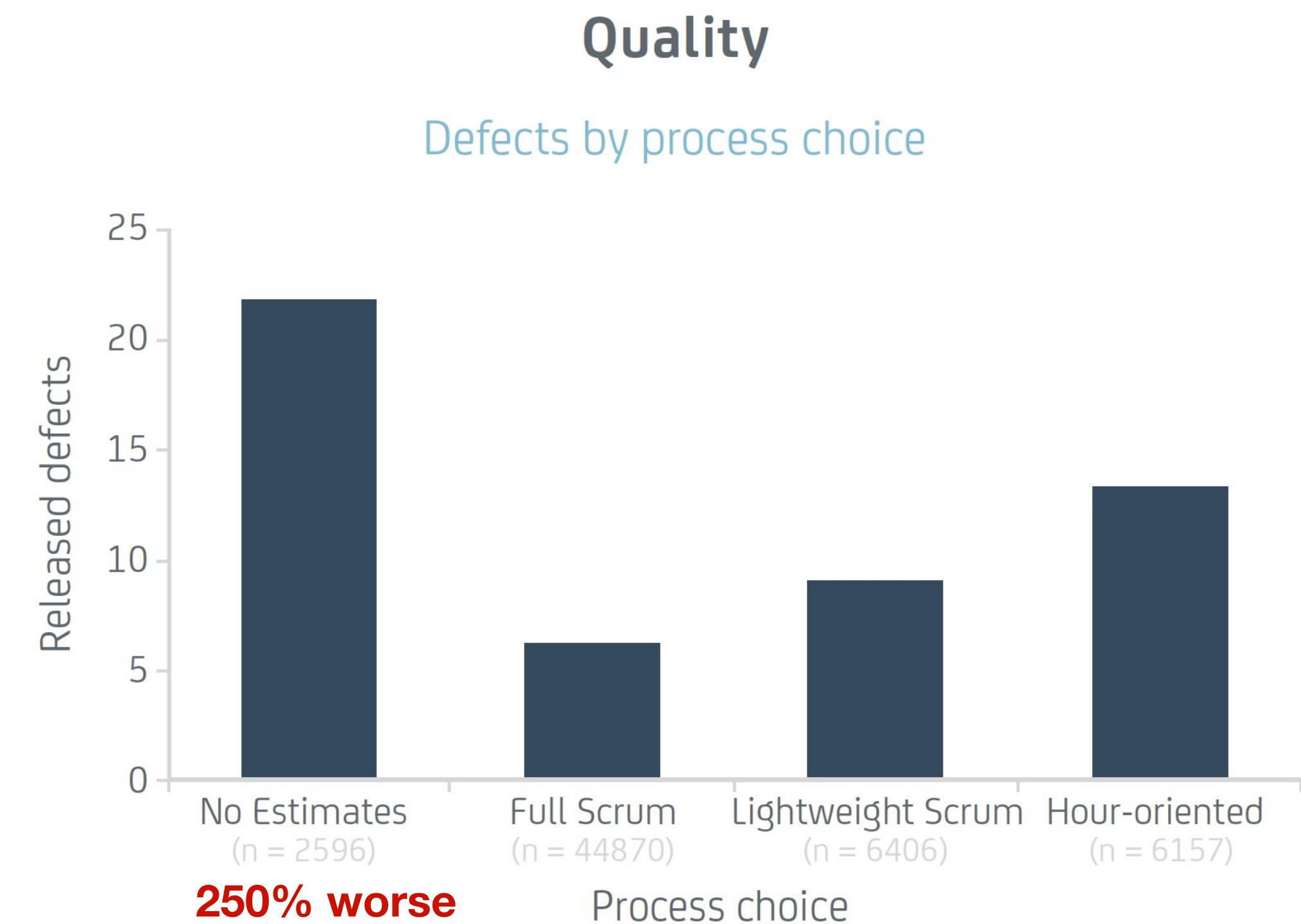
# Case Study: Rally Data Mining 資料探勘 of 60,000 Teams

## Small Stories vs. Tasking vs. Hours 小故事 vs. 任務 vs. 小時

What we found when we compared these various process choices is that teams that follow the Full Scrum process perform better than most alternatives, but Lightweight Scrum is actually better overall. This chart shows a score for each of the four dimensions added together.<sup>†</sup>

It's interesting to note that the group that we believe has received the least coaching (task-hour estimates only) performs the worst, and the coaching recommendation for mature teams (Lightweight Scrum) performs best.

There is one dimension where Full Scrum outperforms Lightweight Scrum, and that is the dimension of Quality. There is a 250-percent difference in defect density between the best and worst process choices, so that's pretty dramatic.\*



# Agile Estimating Strategy 敏捷估算策略

## **Don't estimate time 不要採用時間估算**

- Estimate **relative size** of stories (PBIs) 估算相對大小
- **Measure velocity** per Sprint 測量每個Sprint的速度
- Come up release plan 做出發布計劃

## **Estimates are done by the people who are going to do the work**

估算是由要去做的人來估

- **Not by the people who want** the work done 不是由想要工作完成的人來估
- Don't assume that only one specific team member (Developer) will do the work  
不要設定只由某一位團隊成員(developer)來估

Estimate **continuously during the project**, not only up front 過程中不斷進行估算

**Prefer verbal (or face) communication** over detailed, written specifications

用電話、視訊或面對面溝通，而不要傳送細節規格的書面來溝通

# Estimation 估算 (\*)

---

As a member of a Scrum Team, I need to know how to estimate Product Backlog Items using story points, so that I can help my team pull the right amount of work into a Sprint and be more predictable with our agile way of working

身為Scrum團隊的成員，我需要知道如何使用故事點來估算產品待辦事項，以便我可以幫助我的團隊將適量的工作納入Sprint，並透過我們敏捷的工作方式變得更加可預測

# Estimation is from ...

- Rand Corporation received a grant from U.S. DOD 國防部 in the 1950's to determine best way to estimate tough projects  
決定估算高難度專案的最佳方法
  - Discovered **estimation in hours has high error rate and wide variance**  
發現小時估算會有較高的錯誤率和較大的變動
  - Found people could **put things in relative size piles best**  
注意到將事情用相對大小來堆放最好
  - Experts need to estimate independently - **avoid anchoring** 避免錨定效應
- Delphi estimation technique 德爾菲估計法 has massive amount of research
- See Rand Corp. papers

Dalkey, Norman C. (1968) The Delphi Method: An Experimental Study of Group Opinion. Rand RM-5888-PR  
[http://www.rand.org/content/dam/rand/pubs/research\\_memoranda/2005/RM5888.pdf](http://www.rand.org/content/dam/rand/pubs/research_memoranda/2005/RM5888.pdf)

# Estimation Cards 估算撲克牌

- Choose a **Reference Point**: Pick an item for which the team has the **best shared understanding** that is **not too big and not too small**, and arbitrarily assign it **3 story points** to create a stable starting point.

選出團隊有最佳共識認為不會太大也不會太小的工作項目，就設定它為三個故事點，作為穩定的參考點

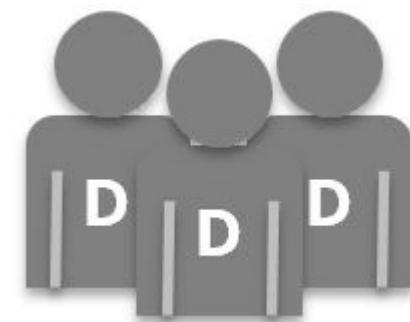
**Tips:** Avoid select too technical PBI as Reference Point 提示：避免選擇過於技術性的 PBI 作為參考點

- **Keeping your thoughts to yourself**, and using only **Fibonacci numbers**, estimate the other items by **comparing the level of effort to your reference item (point)**

自己想自己的，比對參考點所需的工作來估算另一個工作項目，並用斐氏數列來預估故事點數



# Estimating the Product Backlog



## Developers

- **Estimate** backlog
- Estimates are **forecasts**, not commitments  
估算是預測, 而不是承諾



## Scrum Master

- **Facilitates** process  
引導流程



## Product Owner

- Available to **clarify** intent of PBIs **to help estimate**
- But **should not estimate**  
但不應該參加估算

## Two options for estimating process:

### Estimate Stories Individually: 分別逐個地估算故事點數

- Lay out all stories 列出所有故事
- Agree on one - best shared understanding that is not too big and not too small, call this “3-points” 同意選一個大家最了解且不會過大或過小的故事, 將這故事設為3點(參考故事)
- Estimate other stories relative to the reference story 以和參考故事的相對大小, 去估算其他故事

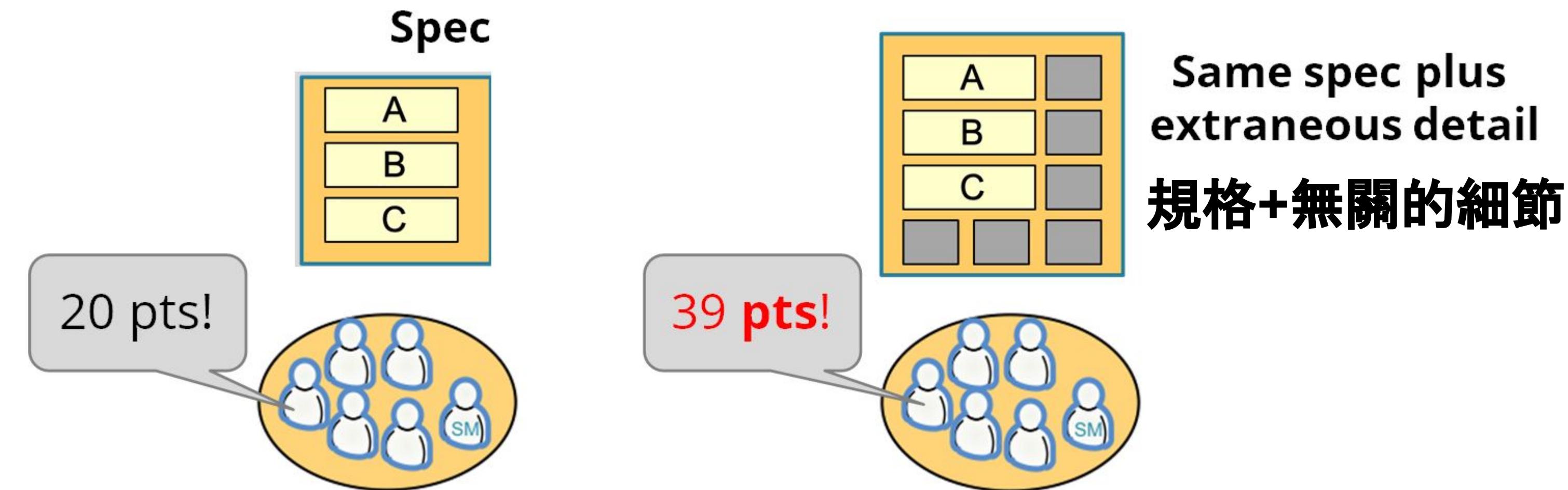
### Estimate Groups of Stories 估算成組的故事

- First, group stories into similarly sized piles of activity 首先, 將故事分組成大小相近的一堆一堆
- Then estimate number of points for each pile 然後估算每一堆的點數
- Fast way to estimate a large number of stories 是快速估算大量故事的方法

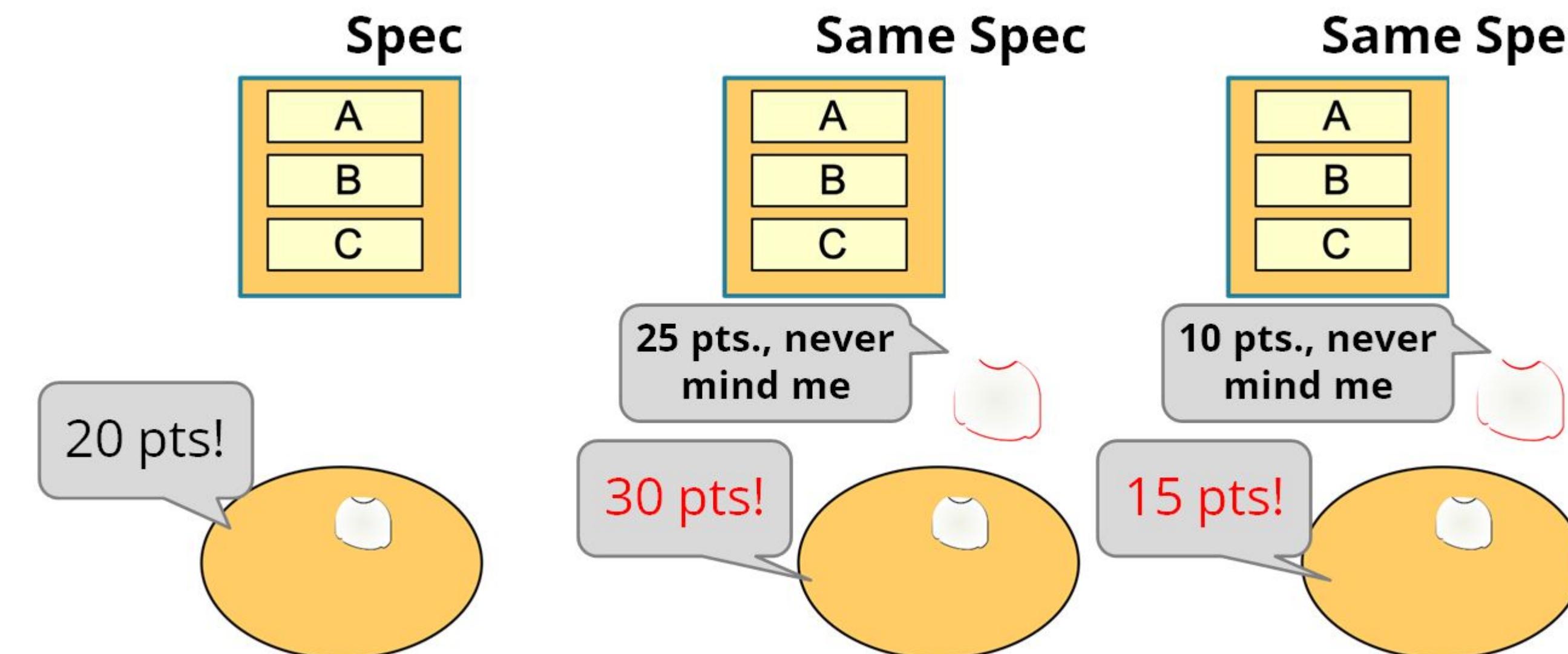
# Beware Common Estimating Pitfalls

## 小心估算時常見易犯的錯誤

### Irrelevant Information 不相關的信息

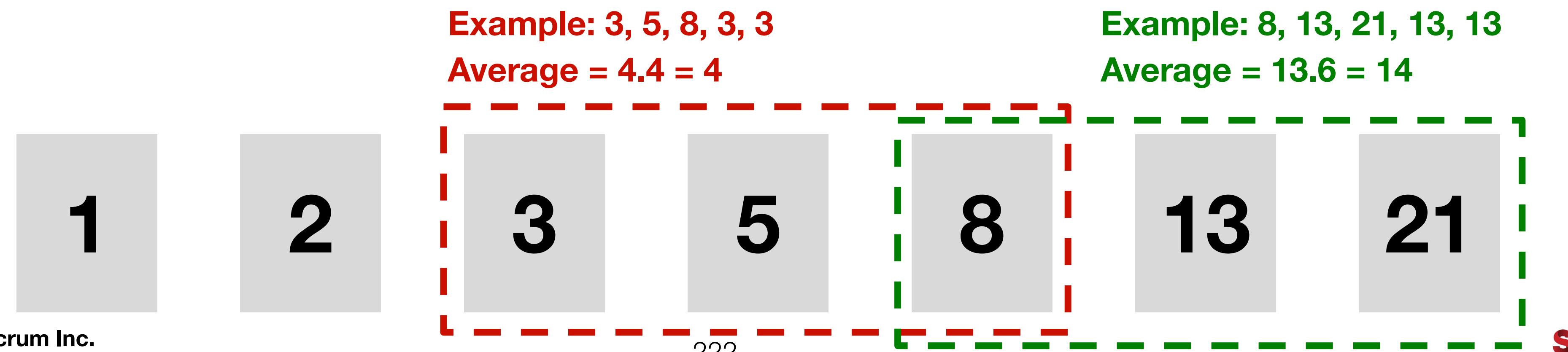


### Anchoring 錨定效應



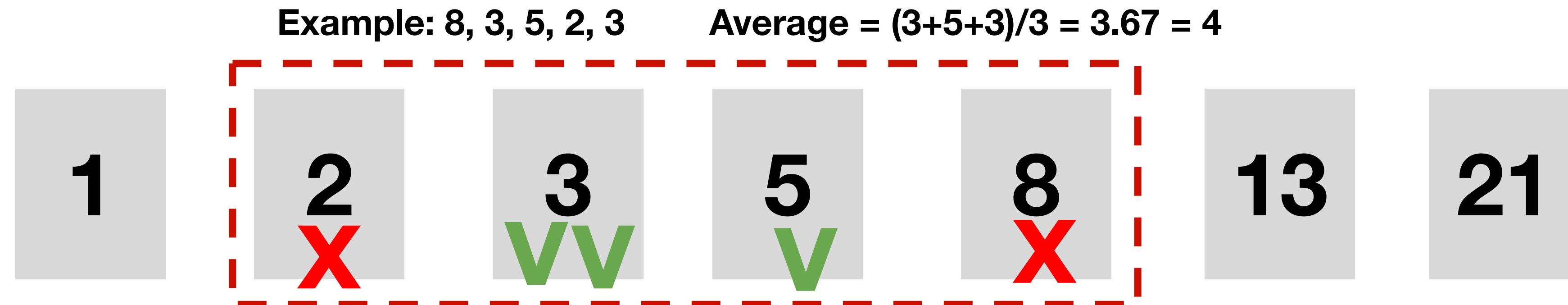
# Leading Estimation Technique 最佳估算方法

- Each Team Member holds their card (or phone if use the App) hidden from view
- When everyone indicates they are ready, the Scrum Master calls “Ready, Set, Show!” or something like that
- Everyone **shows their card at the same time**
  - This prevents “Anchoring” (influencing others) 防止影響他人
- If all of the estimates are **WITHIN three sequential (Fibonacci) numbers, take the average (of all estimates) and move on (to the next story)**  
當所有人的估算值的大小落在三個連續的費氏數列以內，則取所有人的估算值的平均，就是該故事的點數，然後繼續下一個故事



# Leading Estimation Technique

## 重要的估算方法



- If the estimates are **MORE than three (Fibonacci) numbers apart in sequence**, the "high" (8) and the "low" (2) states **why** (like: the assumptions, what are they seeing...)  
當所有人的估算值的大小超過連續的三個費氏數列，就請“最高”(8) “最低”(2) ..說明其估算理由
- And **everyone votes again** (by considering any new information provided by the "why")  
每個人再次決定其估算值（必須考慮其他人提供的任何新的信息）
  - NOTE: This is **NOT A DEBATE**, just a statement of reason 注意:這不是辯論，只是理由的陳述
  - Vote again (**up to 2 more times**) until all estimates are within three sequential numbers, then average  
再次亮出個人的估算值(最多再2次)，直到所有估算值都在三個連續數字之內，然後取其平均值
  - **If after 3 tries** (no need to vote on the 4th time), outliers still exist, **drop the high & low and average the rest!** (Then move on to the next Story)  
如果經過3次，離群值仍然存在，則刪除最高和最低並取其平均值！（然後繼續估算下一個故事）

**Do not try to converge. Consensus is UNNECESSARY!** 共識是不必要的  
**The best estimate may not be a Fibonacci number!** 最後估算值可能不是費氏數列

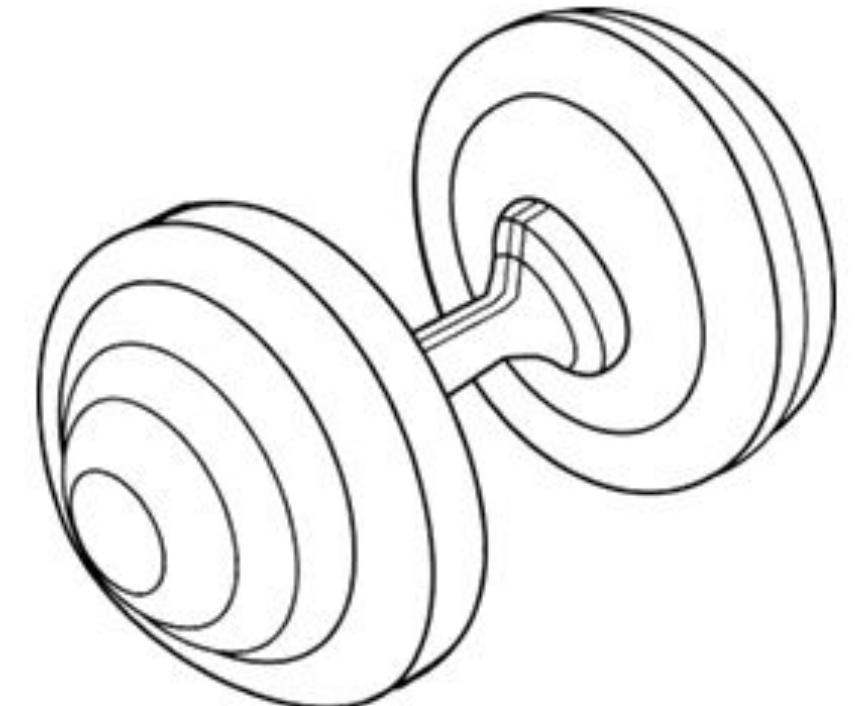
# Weber's Law: Effort Awareness

## 韋伯定理：感覺定律



*Could you  
Detect A  
Difference?*

您能察覺差異嗎？





3





3

MURAL

227



M U R A L

228



MURAL



MURAL

230



MURAL



3



Your conversation will be like: 你們的對話將是這樣的:

If the helicopter is a 3, then the car is a ... 如果直升機是 3, 那麼汽車就是...

If the helicopter is a 3, then the beach house is a ... 如果直升機是 3, 那麼海濱別墅就是...

# Record Your Results

| Model        | Team Estimations in Points | Average |
|--------------|----------------------------|---------|
| Helicopter   |                            | 3       |
| Car          | 3    5    8    3           | ?       |
| Big House    |                            | ?       |
| Front Loader |                            | ?       |
| Beach House  |                            | ?       |
|              |                            | ?       |

**Method:**  
Number of Pieces  
積木片數

106



132



286

706



1636

1

2

3

5

8

13

21

34

55

**Age**  
**6-12**



**Age**  
**7-12**



M U R A L

**Age**  
**8-14**



**Age**  
**11-16**

**Method:**  
Time to Assemble  
組合所需的時間



**Age**  
**6-12**

# What were you thinking about when you chose your estimates? 當您估算時，您在想什麼？



Factors affecting effort estimates 影響估算的因素

- Time it might take to build it 需要花費的時間
- Complexity 複雜度
- Number of pieces 件數
- Sizes of pieces 件數的大小

Other factors affect **real life effort estimates** 現實生活中的估算

- Unknowns / Uncertainty 未知數
- Skill Level 技能等級
- Hand-offs 轉手次數
- Dependencies 相依性
- Complexity of the work 工作的複雜性
- Risk 風險
- The amount of the work 工作量

If there is something not available in a PBI (example: we need to order a new router and it needs another 2 weeks),  
then the PBI is NOT Ready, move on to next PBI

如果 PBI 中有些內容無法使用(例如:我們需要訂購一個新路由器, 並且還需要 2 週), 則該 PBI 尚未準備好, 請繼續下一個 PBI

# Team Throughput 團隊生產力 = Velocity 速率



**True or False:  
Is Velocity a  
Performance  
Metric?**

**Velocity = Total of Points for all Product Backlog Items (PBIs) Completed in a Sprint**  
**速率=在一個Sprint所有完成的產品待辦事項(PBIs)的總點數**

# Velocity 速率

**Velocity** is a measure of **the amount of work a Scrum team can accomplish in one Sprint**

速率是Scrum團隊可以在一個Sprint中完成的工作量

- Most commonly measured by the **number of points** which can be completed by the team.  
通常用團隊可以完成的點數來衡量
- Should be **based on history** (empirical data)  
“**Yesterday's weather**” 基於歷史記錄("昨天的天氣")
- Primarily a **Planning Metric**, not a **Performance Metric** 是計劃指標, 而不是績效指標 (**Question: Why or Why Not?**)



# Important Notes

- In Scrum, **ONLY** the **people who do the work estimate it** (not those who want the work done) 只有做開發工作的人才能做估算(不是要得到交付的人做)
- We **NEVER** try and **normalize estimating between teams** (i.e. how they estimate items or the points they assign to items)  
絕對不在各團隊之間針對估算做標準化
- We **NEVER compare velocity between teams** as there is no way to compare teams that way 絕對不在各團隊之間針對速率作比較
- A team doing 50 points may actually be slower than a team doing 30. **Points are relative** to estimating approaches used and people doing the work. There is no way to normalize this. **DO NOT TRY** 做50點的團隊有可能比做30點的團隊慢, 對估算方式及做開發的人而言, 點數是相對的。不能加以規範, 不要企圖做做看
- We **CAN** compare **velocity trends across teams.** i.e. who are accelerating and who are decelerating or flatlining  
可以比較各團隊的速率趨勢, 誰在加速、誰在減速或持平
- Comparing trends highlights areas **coaching should be focused** 比較走向才是需要關注的地方

# Key Points

- **Estimate is Time is the best way to destroy the morale** (it's been 2 hours, why are you not done?) 使用時間估算是摧毀士氣最好的方法(已經2小時了，你怎麼還沒完成？)
- **Relative estimation gives you the best** of accuracy, predictability, quality, and productivity 相對估算為您提供最佳的準確性、可預測性、品質和生產力
- **Estimate on the effort** that each PBIs will take to complete 估計完成每個 PBI 任務所需的工作量
- Be consistent, so that you will have predictability (Say - Done ratio) 保持一致，這樣你就有可預測性(說和完成的比率)
- Use the leading practice we taught in this chapter when your team is doing the story point estimation 當您的團隊進行點數估算時，請使用我們在本章中教授的方法

# The Sprint

---

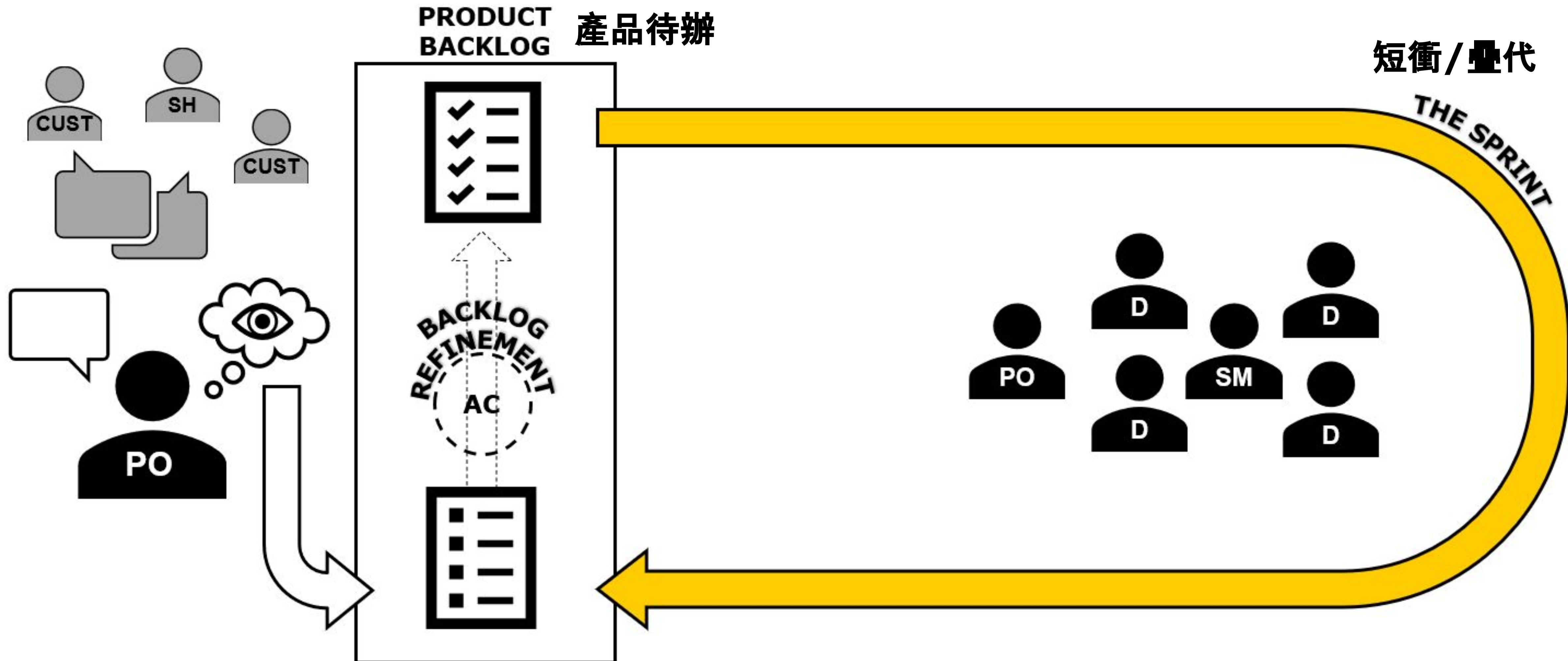
As a member of a Scrum Team, I need to understand what a Sprint is, so that we can deliver a potentially releasable product increment in a fixed period of time

身為 Scrum 團隊的一員，我需要了解什麼是 Sprint，以便我們能夠在固定的時間內交付可發布的產品增量

# Learning Objectives:

- Recognize the **Sprint as a fixed time-box** of one month or less, in which the team **produces a “done” Product Increment** 將Sprint視為一個月或更短的固定時間範圍，團隊在其中生產「完成」的產品增量
- Explain the importance of having a **stable Sprint cadence**, especially with regard to Velocity 解釋穩定的Sprint節奏的重要性，尤其是速率方面
- Describe why **shorter Sprint cycles are preferred** to longer Sprint cycles 描述為什麼較短的Sprint週期優於較長的 Sprint 週期
- How each of **these Sprint Events connect to each others** and how they truly derive value from these Events all together 這些Sprint活動如何相互連結以及如何真正從這些活動中獲得價值

# Scrum Framework



# The Sprint

## What is it?

- A cycle of work of **consistent length** 是前後一致的工作週期
- A container for all other Scrum Events 包含所有Scrum會議
- An Inspect (what you are doing/did and how you are doing/did) and Adapt cycle
- An **agreed upon length of time** in which **the Scrum Team commits** to producing **a meaningful increment of work**  
是Scrum團隊承諾產出有意義的增量所需且由大家同意的時間



**1 to 4 weeks**

**How do you determine the Sprint length?  
如何決定Sprint的週期長度？**

# The Sprint: Why do we timebox our work? 為何要設定固定的工作時間?

## Why?

- A fixed anchor, a predictable cadence, a rhythm 固定的工作節奏, it becomes how you live your life
- Allows a team to **calculate velocity** to derive lessons for the future 協助團隊計算速率, 以獲取學習並用於未來
- A fixed planning horizon 是一個固定/確定的未來規劃
- While we welcome (embrace and leading) change, but During the Sprint: No changes are made that would endanger the Sprint Goal



**1 to 4 weeks**

**Is One week Sprint better than 2 weeks Sprint? (Shorter is Better), Why?**

**Do you start from Mondays and end on Fridays?**

# Key Points

- The Sprint is a **timebox of 1 - 4 weeks**. Sprint 的時間範圍為 1 - 4 週
- The Sprint is an Inspect and Adapt cycle (shorter is better). Sprint 是一個檢視和調適的週期(越短越好)
- At the end of each Sprint, the team **delivers** a potential releasable product increment 在每個 Sprint 結束時, 團隊都會交付潛在可發布的產品增量
- Remember, **Velocity** = Total of Points for all Product Backlog Items (PBIs) Completed in a Sprint 速率=所有在一個Sprint完成的產品待辦事項(PBIs)的總點數

# Sprint Planning

---

As a member of a Scrum Team, I need to understand WHY, WHAT, and HOW is Sprint Planning, so that it helps my team finish early to accelerate faster

身為Scrum團隊的一員，我需要了解 Sprint 規劃的原因、內容和方式，以便幫助我的團隊提早交付並加速

# Learning Objectives:

- Describe the importance of **having a Sprint Goal that is motivating and achievable** 擁有激勵性且可實現的Sprint目標的重要性
- Identify **two methods for establishing the Sprint Goal** 確定建立Sprint目標的兩種方法
- State that **Sprint Planning is time-boxed** to two hours or less per week of Sprint 說明Sprint規劃的時間限制為每週Sprint為兩小時或更少
- Recognize that the **Scrum Master is responsible for facilitating this event** Scrum Master負責引導該活動
- Recognize that the Scrum Master should **help the team confirm their capacity** so they know how much work to pull into the Sprint. Scrum Master幫助團隊確認他們的產能,以便他們知道需要在Sprint中投入多少工作
- Discuss **why only “ready” PBIs should be pulled into the Sprint Backlog** 討論為什麼只有「準備好」的PBI才應該被拉入Sprint Backlog中
- State that the Sprint Backlog is one of the three Scrum artifacts. Sprint Backlog 是三個Scrum產出物之一
- Explain that a **Kaizen item**, or process improvement experiment, should be **at the top of the Sprint Backlog** 解釋Kaizen流程改善實驗應該位於Sprint Backlog的頂部

# Plans are worthless, but planning is everything

## 計劃毫無價值，但規劃才是一切

...“I tell this story to illustrate the truth of the statement I heard long ago in the Army: **Plans are worthless, but planning is everything.** There is a very great distinction because when you are planning for an emergency you must start with this one thing: the very definition of "emergency" is that it is unexpected, therefore it is not going to happen the way you are planning.

So, the first thing you do is to take all the plans off the top shelf and throw them out the window and start once more. **But if you haven't been planning you can't start to work, intelligently at least.**”



DWIGHT D. EISENHOWER

34th President of the United States: 1953 - 1961

Remarks at the National Defense Executive Reserve Conference

November 14, 1957

Mr. Secretary, My Fellow Citizens:

There are a number of reasons why I wanted to come and greet this group, and only a few minutes ago I found another reason which made it so that no one could have kept me away. They said this is one meeting, one conference, that costs the government not one single cent. I can't tell you what a real relief that is and what a reason it is for me attempting to offer you not only my congratulations but my grateful thanks.

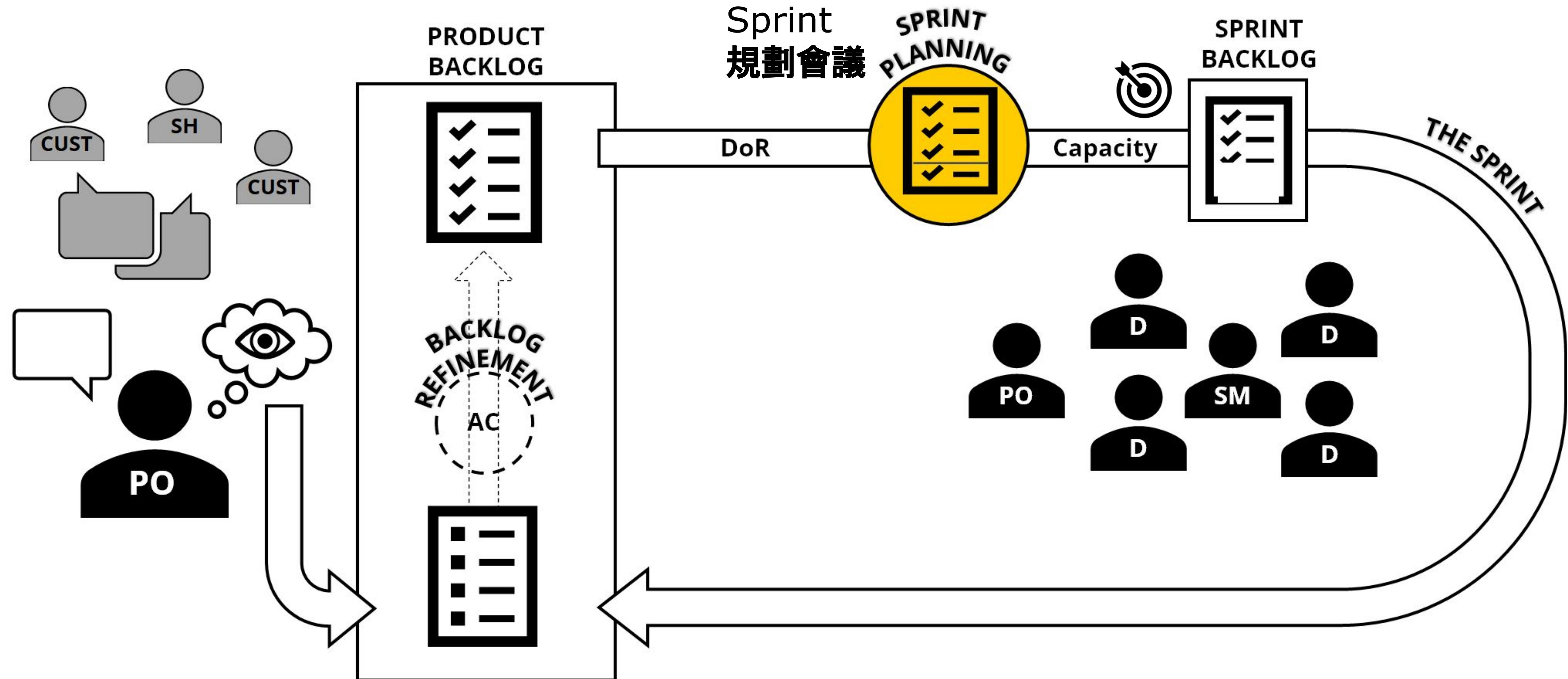
Some years ago, there was a group in the staff college of which some of you may have heard, Leavenworth Staff College. This was before our entry into World War One, and in that course it was necessary to use a number of maps and the maps available to the course were of the Alsace-Lorraine area and the Champagne in France. But a group of "young Turks" came along who wanted to reform Leavenworth. They pointed out it was perfectly silly for the American Army to be using such maps which could after all be duplicated in other areas without too much cost--they would get some area maps where the American Army just might fight a battle. So they got, among other things, maps of the area of Leavenworth and of Gettysburg, Pennsylvania, and in succeeding years all the problems have been worked out on those maps. The point is, only about two years after that happened, we were fighting in Alsace-Lorraine and in the Champagne.

I tell this story to illustrate the truth of the statement I heard long ago in the Army: **Plans are worthless, but planning is everything.** There is a very great distinction because when you are planning for an emergency you must start with this one thing: the very definition of "emergency" is that it is unexpected, therefore it is not going to happen the way you are planning.

So, the first thing you do is to take all the plans off the top shelf and throw them out the window and start once more. **But if you haven't been planning you can't start to work, intelligently at least.**”

That is the reason it is so important to plan, to keep yourselves steeped in the character of the problem that you may one day be called upon to solve--or to help to solve.

# Scrum Framework



# Sprint Planning

EVENT OBJECTIVE (**Why is this Sprint valuable?**): 活動目標

- Create a **Sprint Goal (The Why)** and **Sprint Backlog** which includes selected Product Backlog Items (PBIs) (**The What**) required to achieve the Sprint Goal **agreed upon by the Scrum Team.** 設定Sprint目標及Sprint Backlog, 包括完成團隊所同意的Sprint目標所需完成的所有PBIs
- All PBIs in the Sprint Backlog should be **refined** and “**ready to execute**”. 所有放進Sprint待辦的事項應該已經被優化並準備好被進行開發
- The **Sprint Goal** should be **realistically achievable** during the Sprint. Sprint目標應該是在該Sprint內切合實際可以被完成的
- Sprint Planning is the start of the Sprint

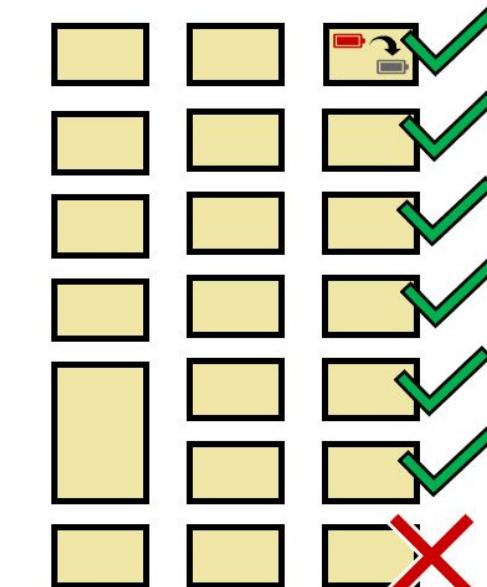


Time boxed to **Two Hours** or Less per **week of Sprint**

以一週為週期的Sprint, 其 Sprint Planning不超過兩小時

# Sprint Planning

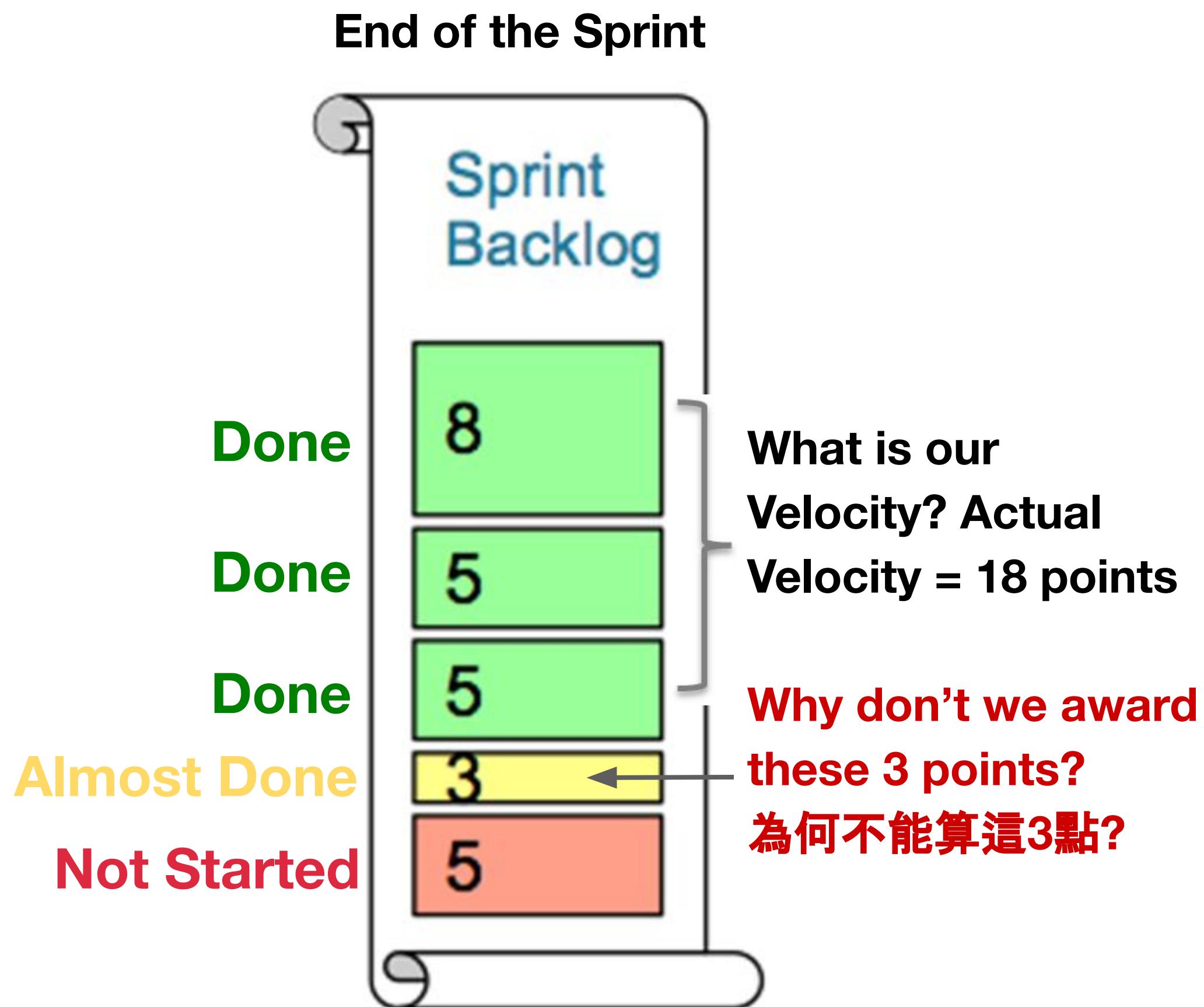
- Find work **capacity limit** for the Scrum team this Sprint. (By Review Velocity on last Sprints)  
找出團隊在該Sprint的工作量上限(點數)(藉由查看最近(3個)Sprint的速率)
- Decide on **a Sprint Goal as a team.**  
團隊一起設定Sprint目標
- Populate the **Sprint backlog with “ready” PBIs** until the work capacity limit is reached.  
當已經做到工作量能(點數), 再去加已經準備好的產品待辦事項
- This event is also the last opportunity to ask: is our estimation still looks right? Can we still break that big story to smaller?
- **Commit to the Sprint** and begin working.  
承諾並開始工作



**How do you know the team is committed?**  
**Hint: the 5 fingers**

# Scrum Pattern: Yesterday's Weather

Use the past Sprints to predict the next Sprint 利用之前的Sprints預測下一個Sprint



If the **team is stable**, how much should we assume can be accomplished in the next Sprint?

若團隊穩定運轉，下一個Sprint可以完成多少點數？

**Answer = 18 points !!!**

If we try to **pull more and fail**, it will reduce morale, further reducing productivity.

如果嘗試做更多卻失敗，將會打擊士氣，且進一步降低生產效率

But if we **keep our expectations in line** and implement the other strategies well, we will **accelerate faster** and be able to **pull more story points in!**

若能將期待保持一致，我們將能加速並拿進更多點數來做！

NOTE: To compute Yesterday's Weather, **take a rolling average of the velocity for the last 3 Sprints**.

要計算昨天的天氣，用最近3個Sprint速率的滾動平均值

Keep recalculating throughout the release.

持續在整個發布的過程中重新計算

# Setting a Sprint Goal (Why) 設定Sprint目標

- **Capturing the Sprint Goal**

- Maximum 1-2 sentence description of what the Scrum Team plans to accomplish 最多用1-2個句子描述團隊計劃完成的事
- Answers the question:
  - **Why is this Sprint valuable?**
  - **What do you want to show at the Sprint Review?** 您想在Sprint Review中展示什麼？

- **2 Methods for setting the Sprint Goal**

Both done collaboratively among the Scrum Team 設定Sprint目標的2種方法，由團隊協作完成

- **Demo Driven** 展示導向: The PO sets a goal and Developers chooses the Backlog Items to complete it. PO設定目標，團隊選擇要做那些事項來達成目標
- **Priority Driven** 優先事項導向: The Developers select the **top-priority Backlog Items** from the Product Backlog which they can deliver and decide what goal the work accomplishes  
團隊從產品待辦事項列表中選擇能完成的最優先事項，並決定它們能達成什麼目標
- The Sprint Goal is the **COMMITMENT** for the **Sprint Backlog**. The Sprint Goal provides **FOCUS** for the Scrum Team on what the PO's intent is to deliver for that Sprint. Sprint目標是對Sprint待辦的承諾



"Would you tell me, please,  
which way I ought to go from  
here?" Alice asks the Cat.



"That depends a good deal on  
where you want to get to," said  
the Cat.

"I don't much care where —"  
said Alice.

"Then it doesn't matter which  
way you go," said the Cat.

# Don't forget to Kaizen 別忘了改善

To pursue continuous improvement, the team should develop **at least one (just one) Kaizen story** (improvement experiment) **every Sprint**

為確保持續的改善，團隊必須在每一個Sprint做到至少一個改善故事(只要一個)

- The **Kaizen is the top priority story** in the next Sprint backlog.

改善故事是下一個Sprint待辦事項中的最優先事項

- While several process improvement opportunities may be identified, **only one experiment** should be conducted in each Sprint

有許多過程改善的機會，但每一個Sprint只進行其中一個試驗



**Kaizen 改善**  
“Change for Better”