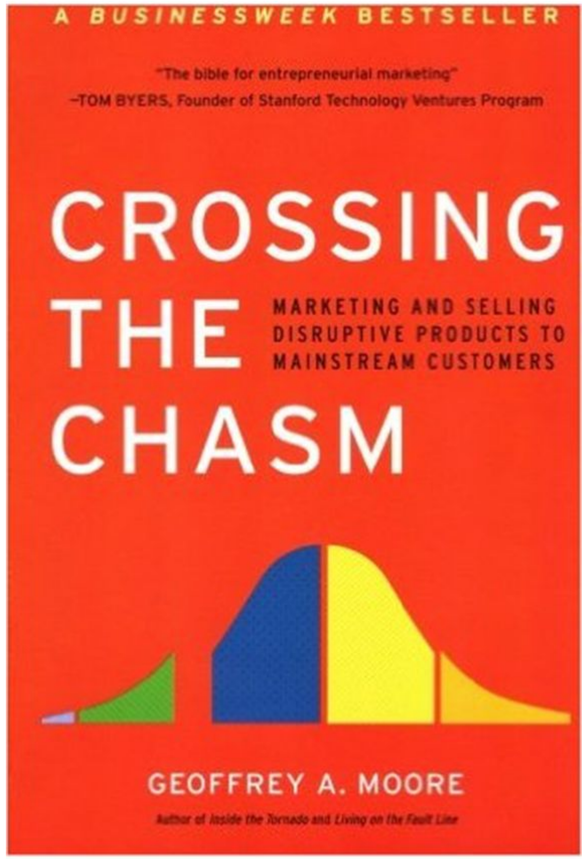
Vision/Mission Statement samples 願景/使命範例

以「光電節能、智慧科技最佳夥伴」為願景

Example: Creating a Vision Statement

FOR <target customers> 為<目標客戶>
WHO <have a need or desire> 誰有<什麼需求>
THE <product name> IS A <product description>
<產品名稱> 是 <產品描述>
THAT <key benefit, compelling reason to buy and use>
有<主要好處，有說服力的購買和使用理由>
UNLIKE <competition/alternative>
不像<競爭對手/替代產品>
OUR PRODUCT <differentiating statement>
我們的產品<差異的說明>



A successful vision statement is **compelling** enough to be broadly shared, yet **concise and easily remembered**
成功的願景聲明令人信服，足以廣泛分享，簡潔易記

scruminc.

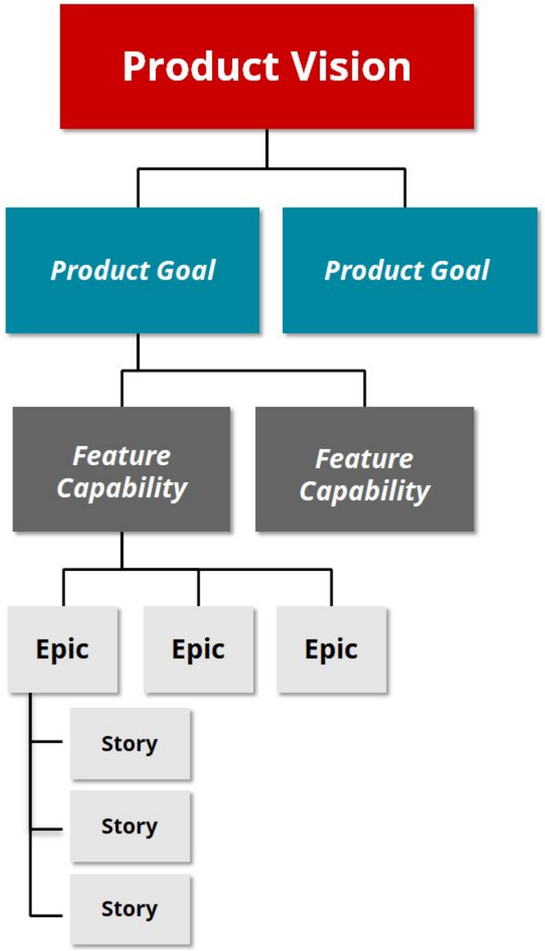
Example: A Vision Statement for Tesla Model 3

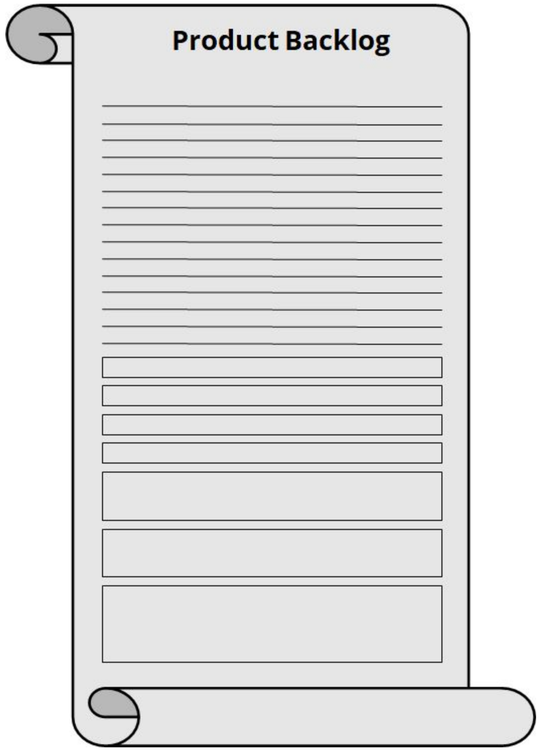
FOR 為<Everyday Drivers 每天開車的人>
WHO <Want a Better and Fun Way to Drive>
想要更好，更有趣的駕駛方式
THE <Tesla Model 3> IS A <Electric Vehicle 電動汽車>
THAT <is Affordable, zero emissions, and over 300 miles range>
價格合理、零排放、充電一次能開300英里以上
UNLIKE <Nissan Leaf>
不像我們競爭對手的車輛
OUR PRODUCT <is priced like a Prius and drives like a Porsche>
我們的產品，價格像Prius，開起來像保時捷

Product Goal 產品目標

- The **Product Goal** is the **concrete deliverable(s)** that **achieves the Product Vision**.
  產品目標是實現產品願景的具體交付
- The Product goal is the **long-term objective** for the Scrum Team.
  產品目標是Scrum團隊的長程目標
- The Scrum Team must fulfill (or abandon) a Product Goal before taking on another one.
  Scrum團隊必須完成(或放棄)一個產品目標，再接著完成下一個

Creates a Backlog from the Product Vision & Product Goal
根據產品願景和產品目標來建立產品待辦

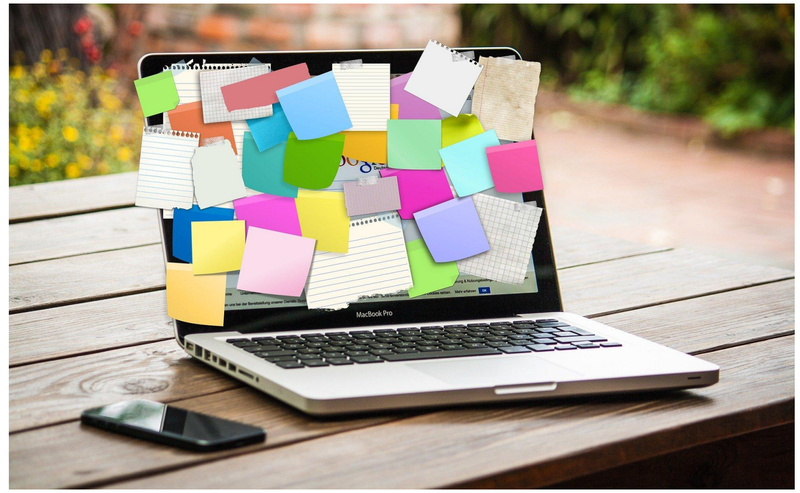




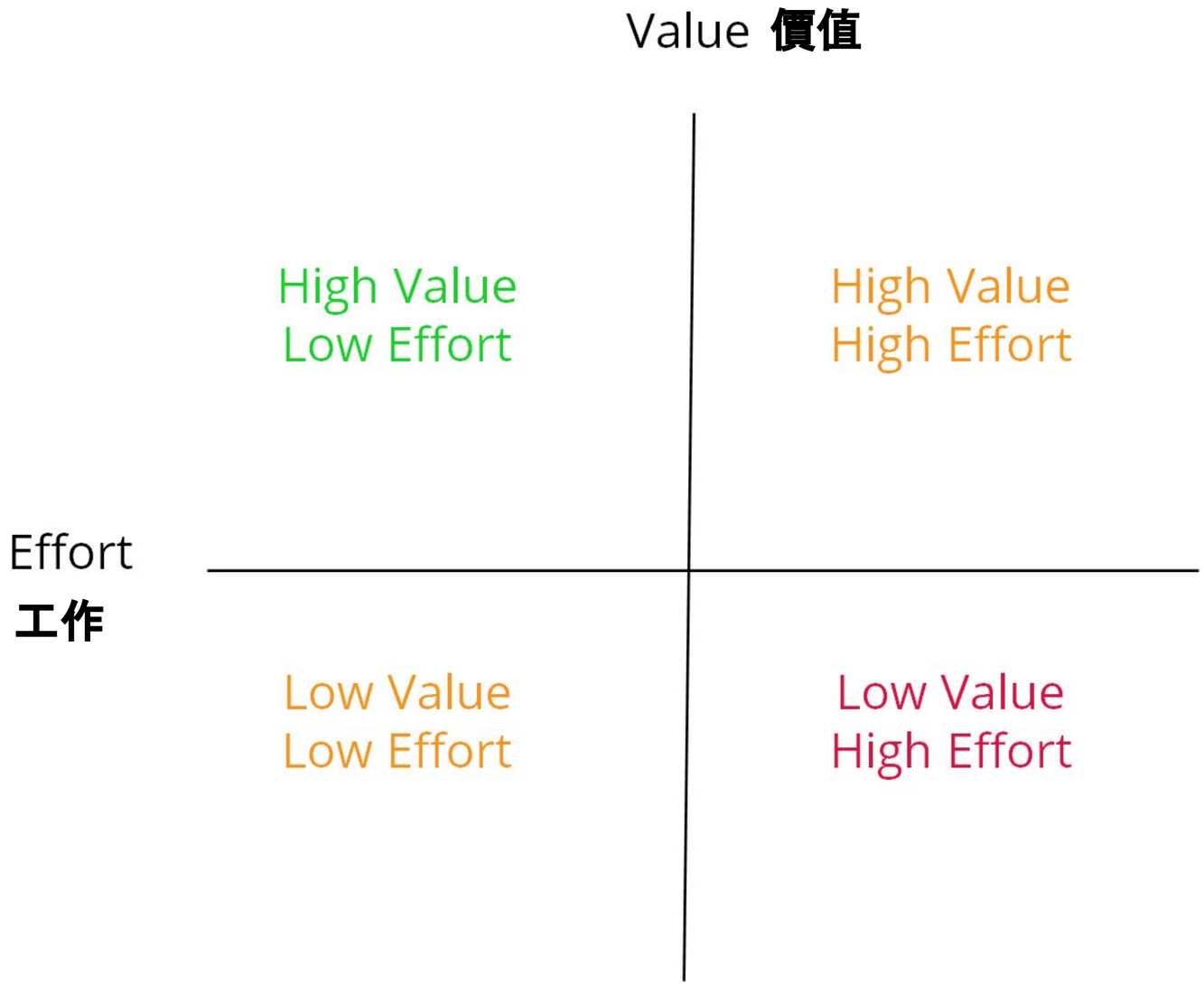
scruminc.

What to Do First? 首先要做什麼？

- The most important question for a Product Owner to answer is **what to do first?**
  PO要回答的最重要問題是首先要做什麼？
- All other questions depend on this answer
  所有其他問題都取決於此答案
- **ROI** calculations give you the earliest delivery of value
  ROI計算可為您帶來最早的價值交付
- **Estimation Cards** is easy and works well. More sophisticated approaches use financial metrics
  更複雜的方法使用財務指標
- One of the best financial methods is **Net Present Value**
  淨現值是最好的財務方法之一



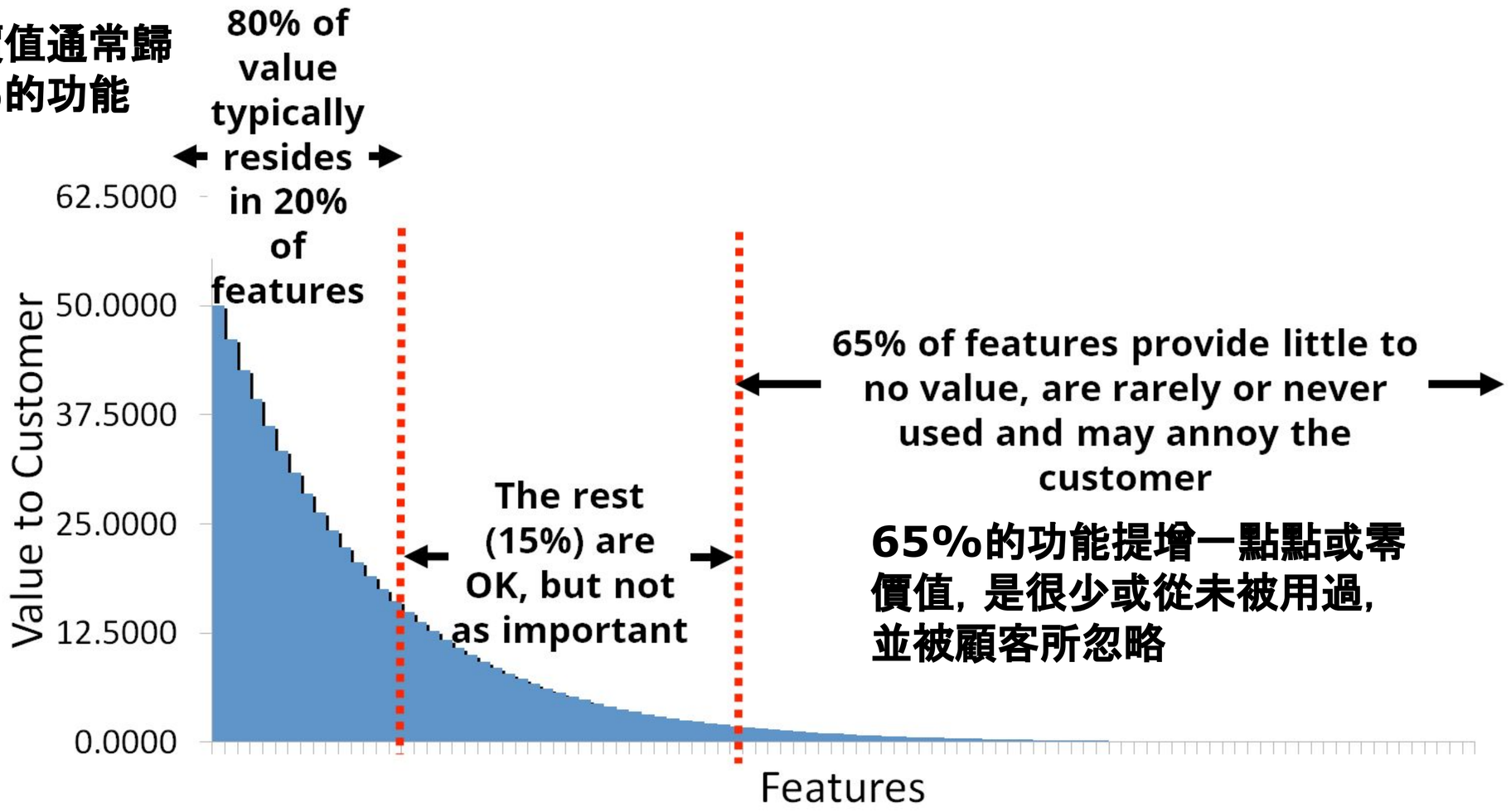
What Do We Build First? 我們要先做什麼？



Not All Features Are Created Equal!

並非所有功能都創造相對一樣的價值

80%的價值通常歸屬在20%的功能



How can you tell ahead of time which features add value and which don't?

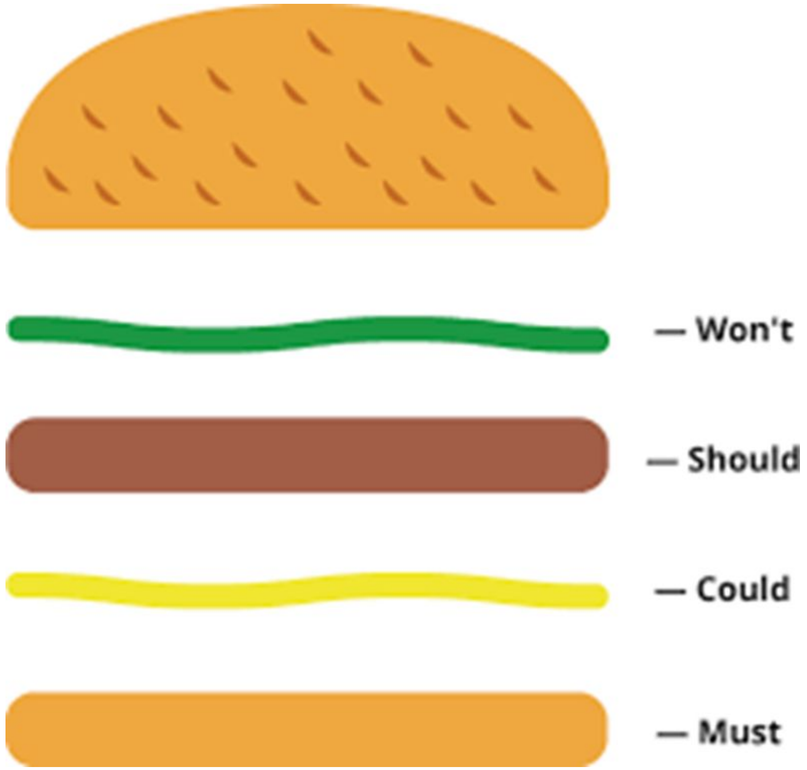
如何在事先知道哪些功能會提增價值，那些不會呢?

scruminc.

What We Build First Depends on Competitive Environment
我們首先建造的(產品)取決於競爭環境

MoSCoW (縮寫變成"莫斯科")

- **M**ust Have
- **S**hould Have
- **C**ould be nice to have
- **W**on’t need to have this - may be later



Kano Model 狩野模型/二維品質模型

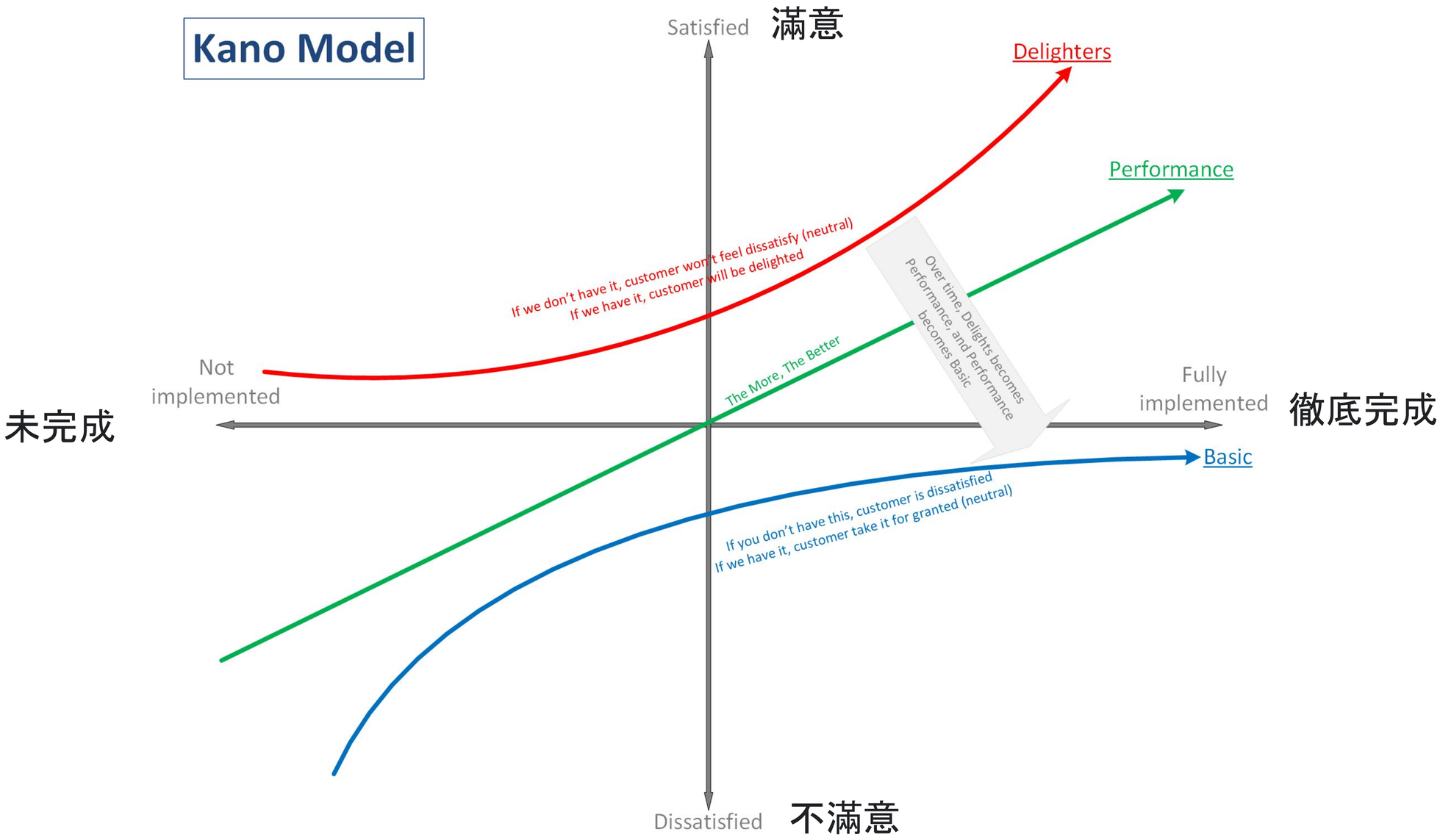
Noriaki Kano: Quality is subjective 品質是主觀感受

- Kano analysis is a quality measurement tool used to **prioritize customer requirements** based on their impact to customer satisfaction. [John Carter, isixsigma.com]
  這是品質衡量工具，根據客戶滿意度來確定客戶需求的優先

We can divide perceived quality into **four groups**
可以將感知的品質分列以下四類

- **Exciters 令人興奮:** positive, beyond expectation 正向，超出預期
- **Performers** (or Satisfiers) **完成或令人滿意**: linear qualities – the more the better 線性量化 - 越多越好
- **Basic needs 基本需求:** we expect them to be there, if not we are dissatisfied 期望得到，如果沒有會感到失望
- **Indifferent 無關緊要:** we don't expect them, and we don't care. Some might be annoying. 不期望得到，且不在乎，有些還會令人嫌煩

Establishing Business Value 建立商業價值



scruminc.

Ways to Order the Product Backlog 待辦排序方式

Bubble-sort Strategy 氣泡排序策略

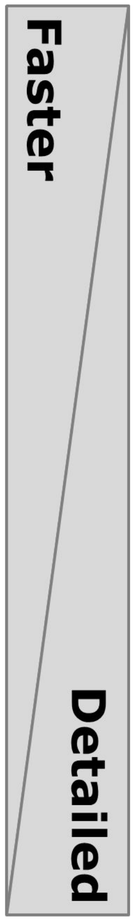
- Take first two items – which is more important?
- Take second and third – which is more important?
- Keep doing it until sort is complete

Low Priority First Strategy 低優先策略

- Assume project does not complete one item – which item is given up? 假設專案能有一項待辦沒完成 - 放棄哪一項待辦？
- Assume another is not complete – which one is given up? 假設另一個待辦沒有完成 - 放棄哪一項待辦？
- Keep doing this and back into a forced ranking

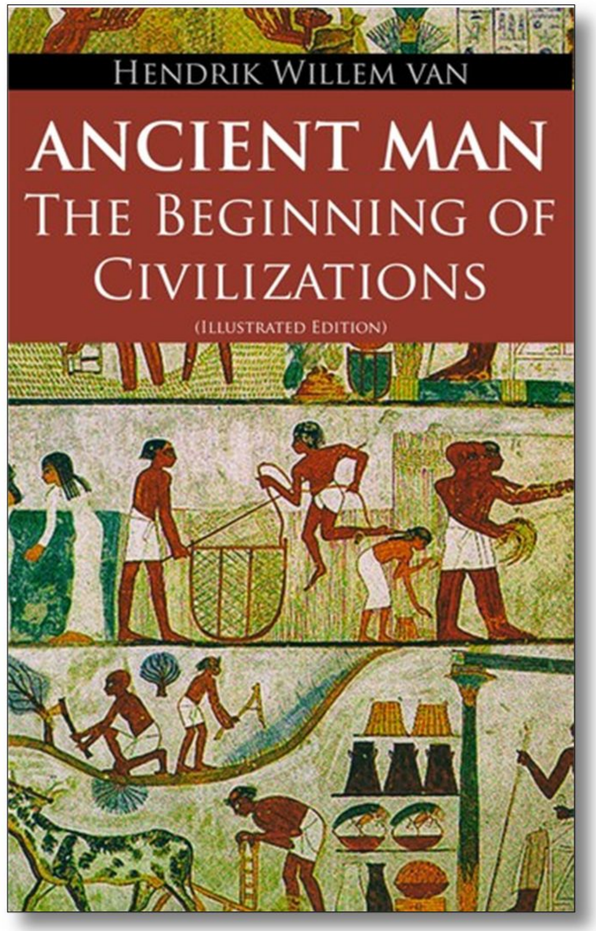
More Comprehensive Approaches 更能令人理解的方式

- Estimation Cards 估算撲克牌
- Financial metrics - NPV/point 財務指標



Definition of an Epic 的定義

- **An Epic is a Product Backlog Item or User Story** that is **too big to be completed in one Sprint**
  Epic是太大的產品待辦或用戶故事，無法在一個Sprint中完成
- Simple Epics need to **broken down** so that the Team can deliver value in a given Sprint - **Done at Backlog Refinement**
  在待辦優化會議中將Epic拆分，團隊才能在Sprint一一交付價值
- **Larger Epics** require the Product Owner to work with Leadership and the Scrum Team to **create a Road Map** so **the most valuable features are created first**
  較大的Epic須由PO與管理者及Scrum團隊一起合作，才能制定路線圖，以便首先開發最有價值的功能
- Epics are **components of the enterprise's vision**
  Epic是企業願景的構成要件
- **Business value can best be estimated at this level**
  在此級別上，能最佳估算(產品)商業價值



Example: Tesla in Kano Model

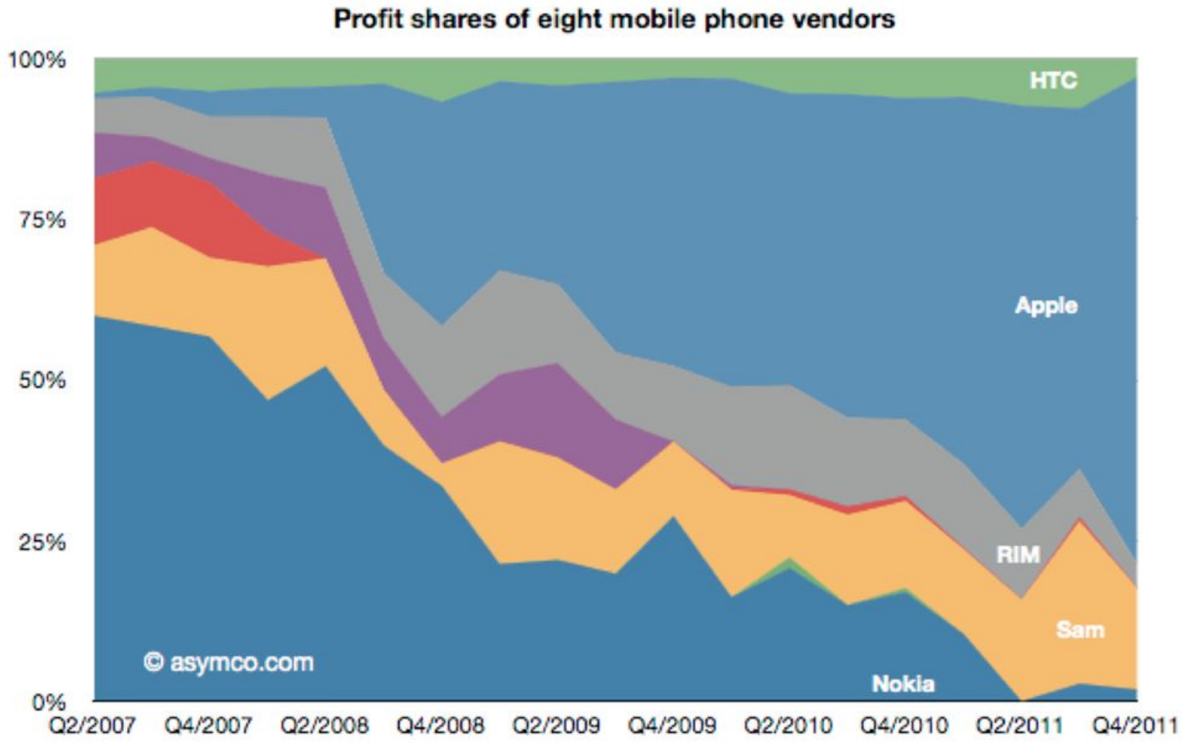
Visualizing the Customer 想像消費者的模樣 (*)

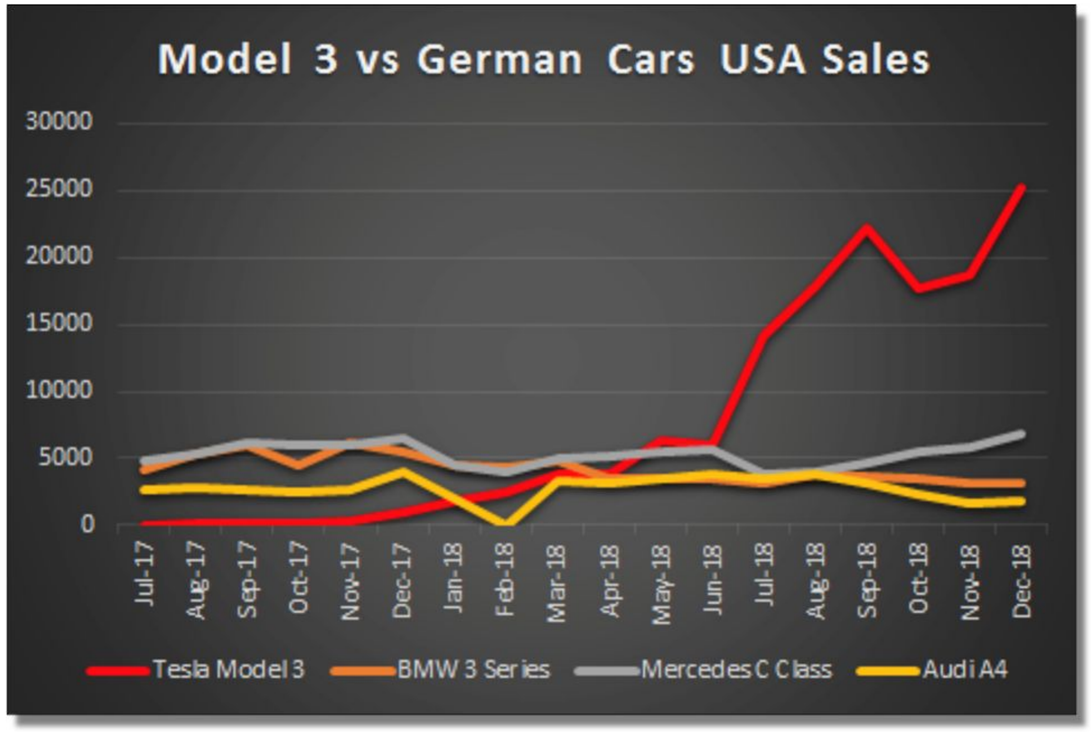
As a Great Product Owner, I need to Visualize the Customer, so that we are delivering the products that our customers want

身為很棒的Product Owner，我要溝通清楚我們要交付什麼及為何要做該交付，所以大家才會贊成/支持我們的產品目標

Customers Don’t Know What They Want Until They See It!

Humphrey’s Law (軟體品質之父)





[BMW CEO Steps Down After a Too-Cautious Strategy Loses Market Share]

scruminc.

A Successful Product Must Delight **All "Customers"**
成功的產品使所有"顧客"滿意

Users

- **Interact directly with the product**
  直接與產品互動
- Knowledge of **current usage patterns helps to design better**, more usable products
  了解當前使用模式有助於設計產品
- Unsatisfied users work around the product, nullifying its benefits and eventually eliminating it
- **Example?**

Purchasers

- Make **buying or adoption decisions**
  做出購買或採用的決定
- **Have their own wish lists** that may have **little to do with the users' needs**
  可能與用戶需求無關
- **Make purchasing decision**, so if they aren't happy, you won't get in the door
  做出購買決定
- **Example?**

Influencers

- Interface with the product or its users
- **Support, install, deploy** or benefit from use of the product
  支援，安裝，部署產品
- Can also wield **significant influence on decision to purchase**, retain or change products
  對購買決定產生重大影響
- **Example?**

Internal Stakeholders

- **Influence scope, priorities, budget and schedule**
  影響範圍，優先事項，預算和進度
- Assist with or may be dependent on **product releases**
- Can constrain or evaluate architecture or product development processes
- **Example?**

Who else could be your Customers?

How do you normally gather requirements?
您通常如何收集需求?

[Link: The Leading Way to Gather Requirements]

scruminc.

Let's Practice: Conducting a Product Interview 進行產品訪談

A sample way to gather more than requirements 取得多過產品需求單的資訊

We need two volunteers! 兩位自願者
(one as app developing firm, one as user)

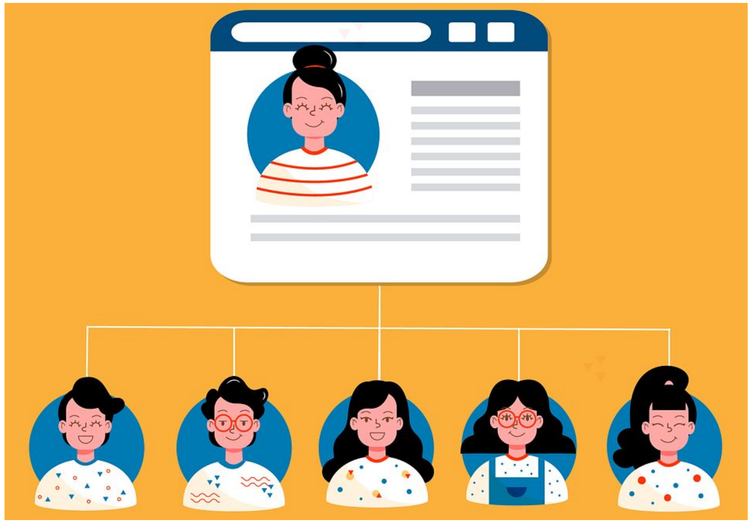
- What is your favorite phone app? 您最喜歡的手機app是什麼?
- Can you show me **how you use it?** 請示範您如何使用它?
- **How long** do you use it everyday? 您一天會花多少時間使用它?
- If the app disappear, **what feature will you miss?**
  如果應用程式消失, 您**會想念**什麼功能？
  (and what feature will you **not** miss? 以及**不會想念**什麼功能？)
- What would you change to **make it better**? 您覺得做什麼改變會讓那個app更好?
- What feature you **like to add**? 您會想增加什麼功能嗎?
- ...

What did we learn? 我們學到什麼呢?
What request should we put at the top of the backlog? 有什麼需求必須加在待辦的優先項目?

Personas 角色描述

A “Persona” is a **fictional** representation of an actual customer/consumer and is applied **throughout the stages** of product development or redesign
“Persona”是一位真實顧客/消費者的虛構代表，適用在整個產品開發或重新設計的各階段

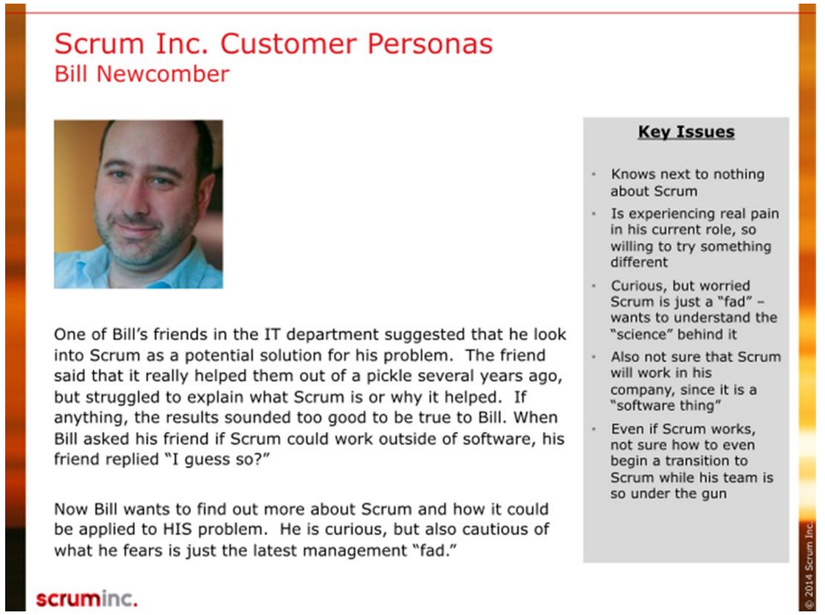
- Becomes **the face of the backlog item** to **create more empathy** about their needs 會成為待辦事項的面貌，以利產生更多對需求的同感
- **Provides context** to the development team to make informed decisions 提供開發團隊能根據資訊/情報做決定的來龍去脈
- Creates **a shared understanding** of the customer/ consumer’s perspective 建立對顧客與顧客觀點/看法的共同理解
- You can use multiple personas to create products that **promote diversity & inclusion**
  可以用多個Persona來創造具多樣化與包容性的產品



Buyer Personas Make Data Real

消費者的**角色描述**讓數據資料真實呈現

- **Personas** are archetypes…**not real people**
  角色描述是原型 … 不是真實的人
  - Describes the "centroid" of a customer segment
    描述客戶群的"重心座標"
- Provides context for a user and **what he/she wishes to accomplish** 提出使用者的情境，以及她/他希望成就的事
  - Team can design for just one person 團隊可以只為一個人設計
  - Personas often **end feature debates**
    角色描述經常可以終結功能上的爭議
- Personas can include: 角色描述可以包括：
  - **Profile:** Name, Picture, Age, Degree, Gender, Income, Location, Experience 個人資料：姓名、照片、年齡、學歷、性別、收入、所在地、經歷
  - Quote, Bio, Goals, Pain Points… 引述、傳記、目標、痛點…



Avoid Next Bench Syndrome

scruminc.

Fleshing out Personas for Greater Context
為得知更好的來龍去脈，顧客描述要夠充分(賦予血肉)

Determine the information relevant to understanding customers (particularly the buyer decision maker): 判定哪些資訊是相關的，來了解顧客（尤其是買方決策者）

- **Who** are they? 他們是誰？
- What is their typical **role**? 他們典型的角色是什麼？
- How do (might) they **use the product**? 他們(可能)如何使用產品？
- How would you **recognize them** if you saw them on the street? (demographics)
  您將如何認出他們？(人口統計資料)
- What are their **"pain points"**? 他們的"痛點"是什麼？
- **How do buyers make decisions about purchases?** 消費者如何決定要購買？

Avoid common persona pitfalls: 避免常見的角色描述陷阱:

- Beware of extraneous detail! 當心多餘的細節！
- Base personas on data **from customers** don't just rely on "creative writing"
  角色描述是以來自消費者的數據作為基礎，並不只是靠創意文案
- Use the persona **throughout the design process**, not just at the beginning
  在整個設計的過程中使用角色描述，不是只在流程的初期
- Continue to **evolve personas based on what you learn**
  根據您所發現及學到的繼續進化您的角色描述

How to Use Persona's in the Team Room

Keep the persona visible

將角色描述放在看得到的地方

- Often posted next to information radiator in team space

  通常是張貼在團隊工作空間之訊息佈告欄旁邊

Refer to personas in team discussions

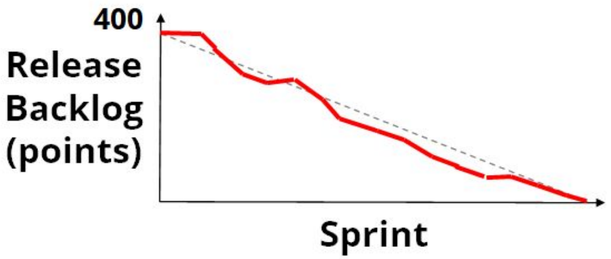
在團隊討論中提及與角色描述的關聯

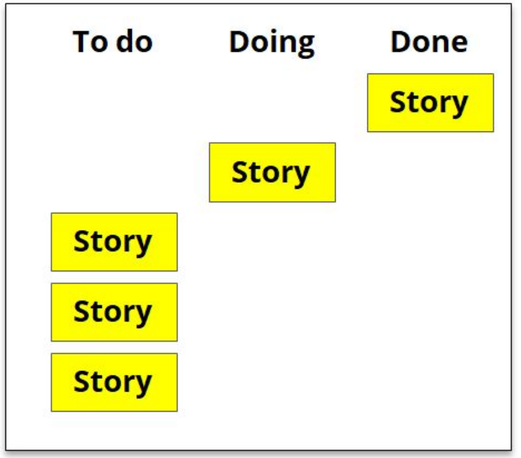
- "I think David would appreciate…"

  "我認為David會喜歡 …" (為人物設定取名字)

Use them to settle disagreements about what to build or how to design the interface

利用角色描述來解決關於構建什麼或如何設計界面的意見分歧

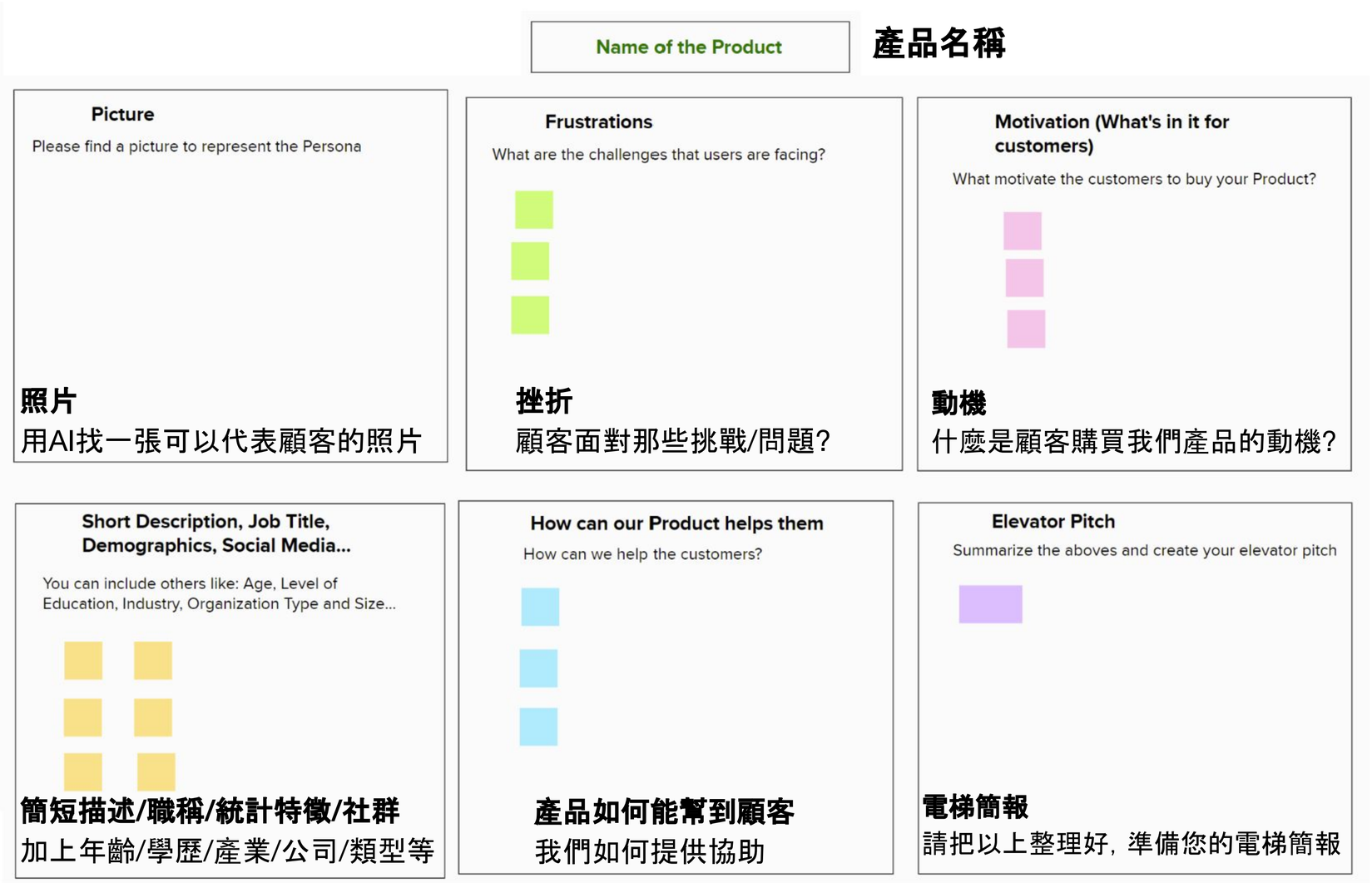




Build My Persona

scruminc.

Let’s build a Persona (for your Product)



Business Value and ROI
商業價值及投資報酬

As a Product Owner, I need to know how to estimate Business Value and ROI so that I can effectively order the Product Backlog

身為Product Owner，我要知道如何估算商業價值及投資報酬，
所以我可以有效地排序產品待辦

Sources of Business Value 商業價值的來源

Will this feature allow us to: 完成這功能將可以:

Market Value 市場價值

- **Sell more** units? 賣更多？
- Charge a **higher price**? 收取更高的價格？
- **Reduce the cost** of providing the product/service? 降低提供產品/服務的成本？
- Improve quality? 改善品質?

How will completing this PBI allow us to: 完成這待辦事項將可以:

Risk Reduction 降低風險

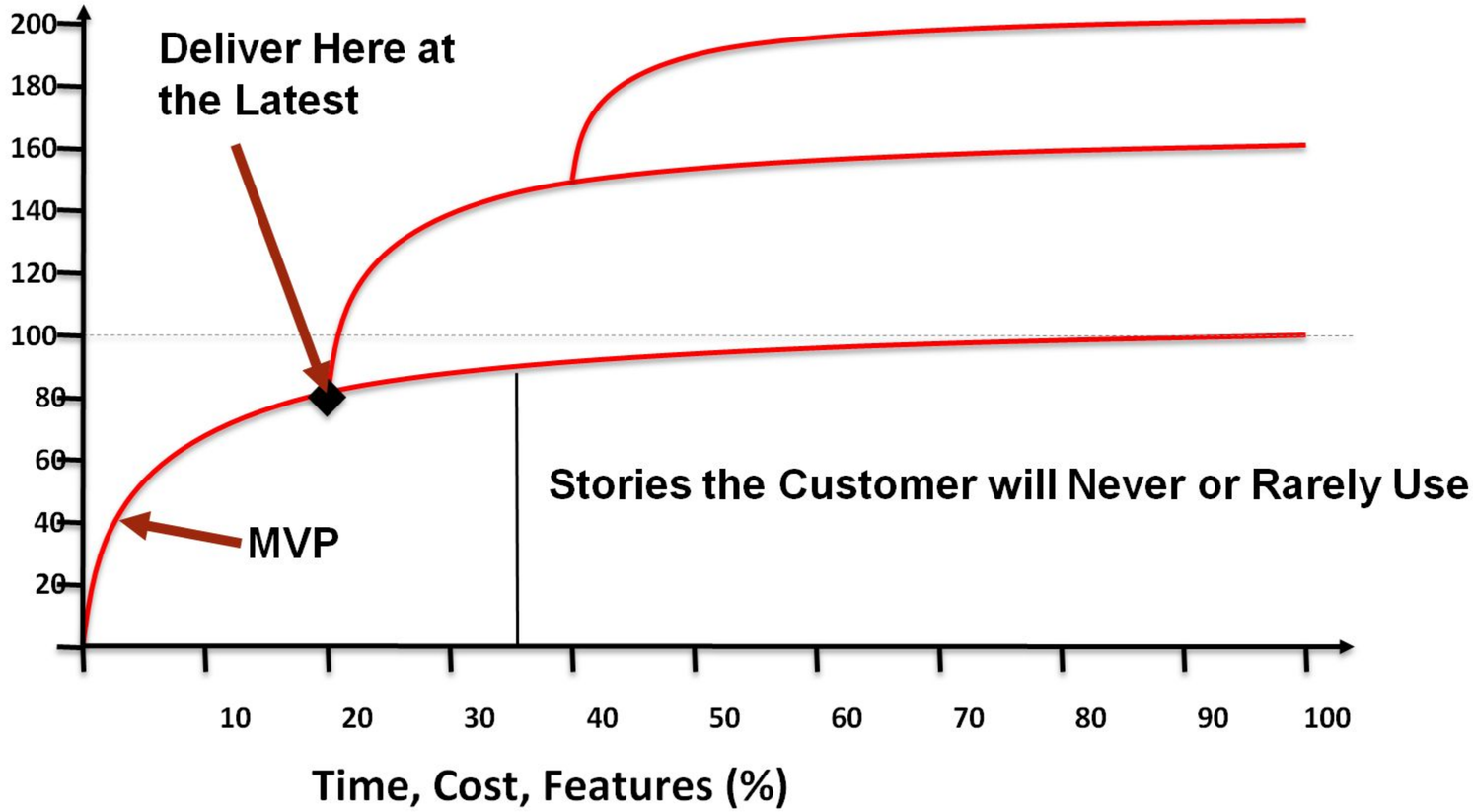
- Develop or refine hypotheses about the market? 進展或琢磨推敲對市場的假設？
- Prove technical assumptions? 證明技術上的假設？
- Improve compliance or reduce security fraud 改善政策法規或降低安全舞弊

Will completing this story: 交付這個故事將可以:

Capability Building 能力/生產力的增長

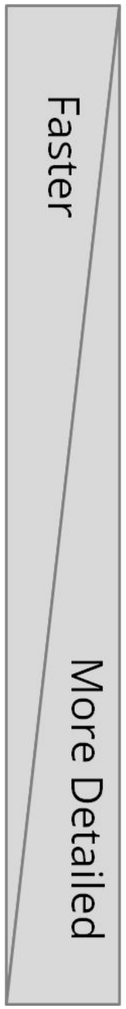
- Enable our team to do something we couldn't before? 使我們的團隊能夠做我們以前無法做到的事情？
- Reduce or eliminate the need for low-value activity? 減少或消除對低價值活動的需求？

Product Owner Delivering Customer Features Incrementally 逐步交付功能 can Drive Radically Better Value Delivery 更有影響的價值交付



scruminc.

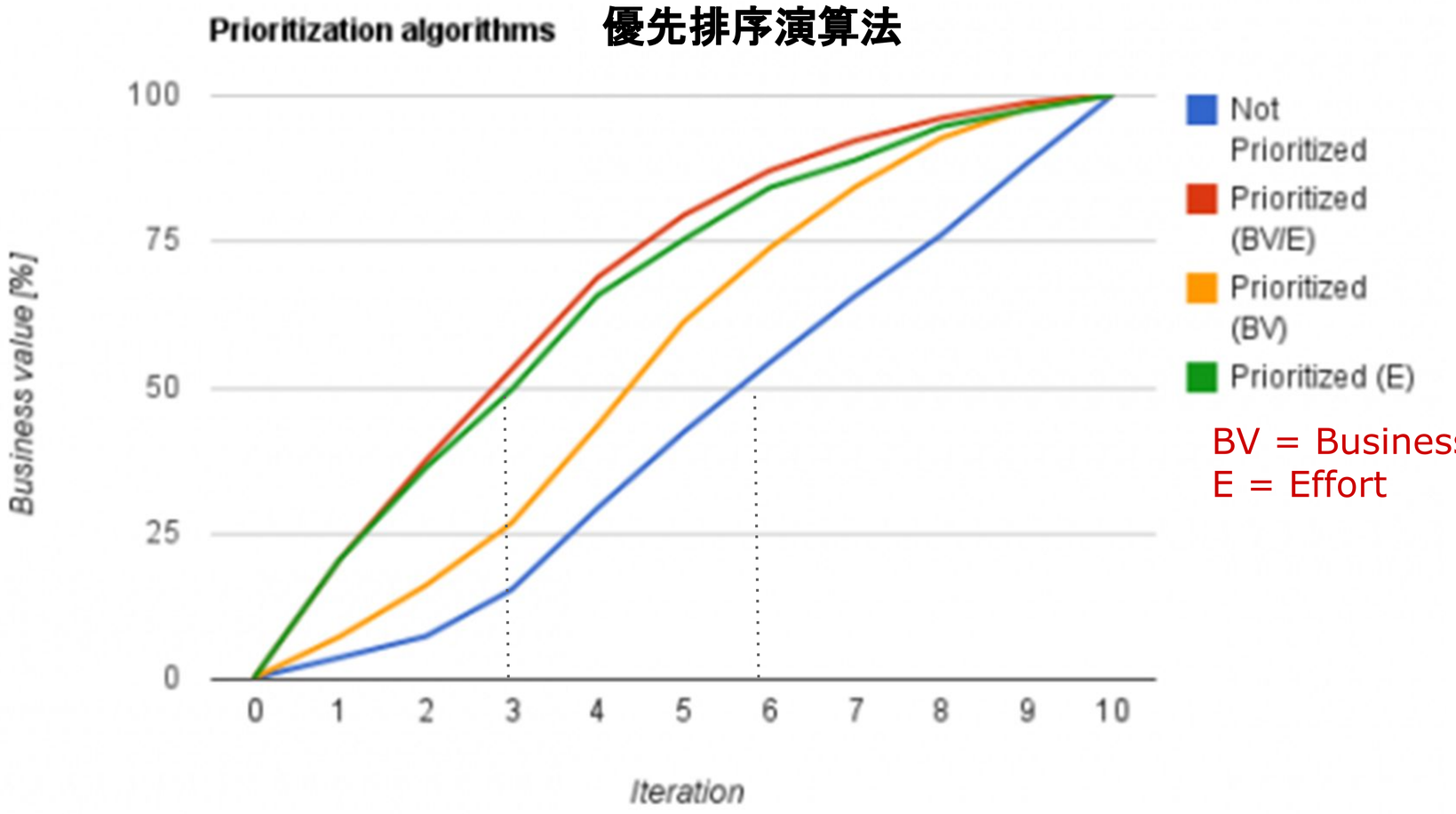
Methods for Determining Value 決定價值的方法



- Bubble Sort
  氣泡排序法
  - Start at the top of a list of stories
  - Compare value of stories one at a time
  - Move the lower value story down one place in list
  - Repeat until all stories have been compared
- Value Poker
  價值撲克
  - Pick a low value item and assign it 3 points
  - Use estimation cards to independently estimate a story
  - Show estimates, if everyone is within three numbers, average the estimates
  - If not within three, discuss highs and lows, estimate again (max 2 more times)
- Cost of Delay
  延遲成本
  - Estimate in a lightweight way the opportunity cost of NOT completing a feature
  - Often divided by feature size to get a "proxy" for ROI
- Break-even analysis
  收支平衡分析
  - Compare cost of feature creation with expected incremental revenue of feature
  - How many incremental units would we need to sell to equal the development cost?
- Return On Investment
  投資報酬
  - = [Total expected revenue from new feature]/total cost to develop feature] – 1
  - Expressed as a percent
- Cash Flow Analysis
  現金流分析
  - Over a reasonable planning horizon, what are the revenues and expenses associated for a feature in each month?
  - What is the net effect on cash flow over that horizon?
- Net Present Value
  淨現值法
  - Building on the cash flow analysis, what is the effect of including the "time value of money" in the calculation? (i.e. a dollar today is worth more than a dollar tomorrow)

Remember $100 Game? 100美元遊戲?

Prioritization of Value/Effort Can Cut Cost & Time to Market by 50%
以價值/努力去做優先排序 可以將成本和上市時間減少50%



BV = Business Value
E = Effort

CPO of Hewlett Packard how to get 20B more revenue per year **by prioritizing by value**
Look at Blue on Sprint 6 vs Red on Sprint 3, same 50% BV in 3 Sprints (instead of 6)

scruminc.

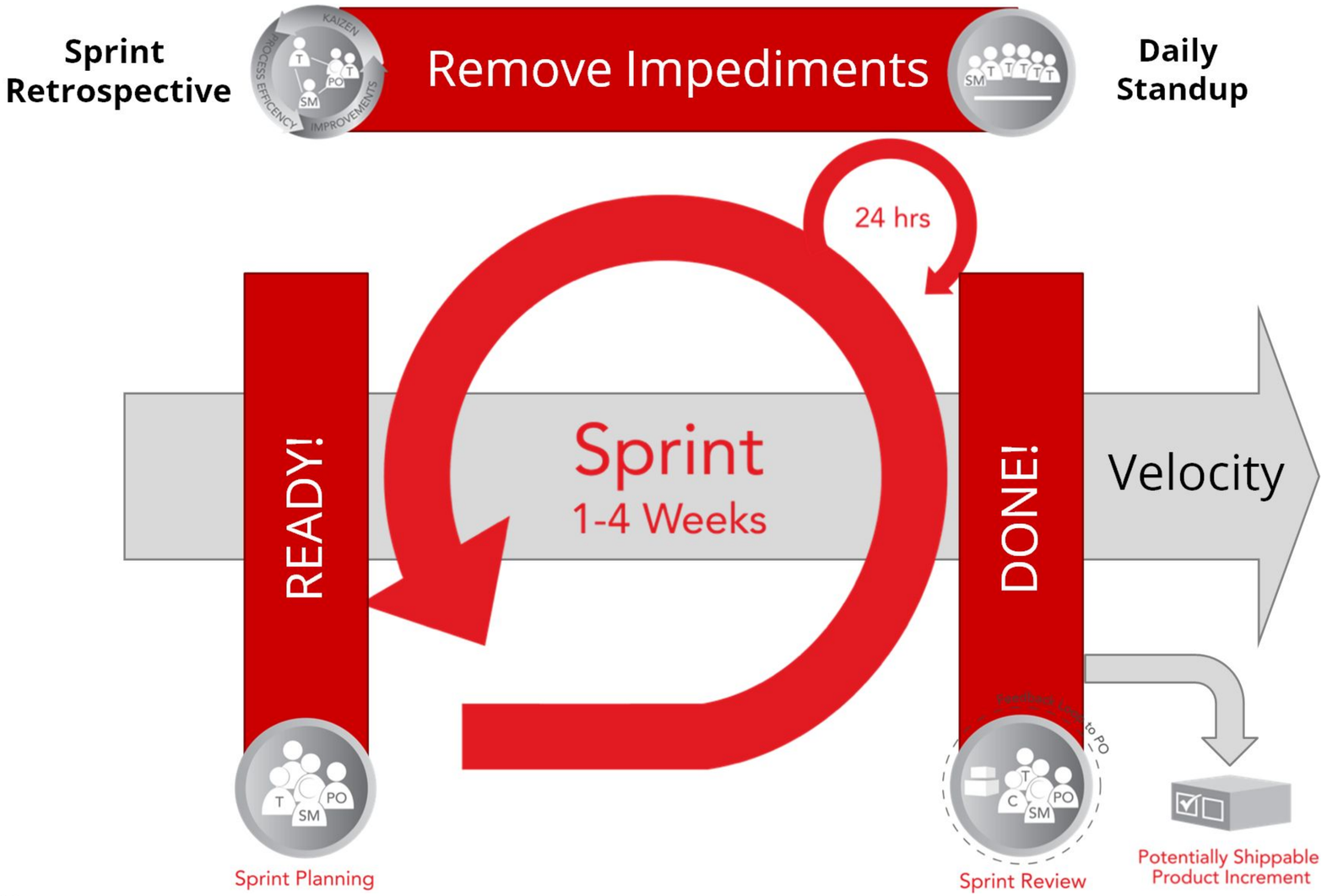
Ready and Done (*)

As a Product Owner, the Team and I need clear **definitions of READY** and **DONE** so the Team can complete the backlog quickly and effectively

身為Product Owner，團隊和我需要清楚的準備好的定義，以及完成的定義，

所以團隊和我可以快速有效地完成待辦

Teams Succeed by Driving READY Stories to DONE, while Removing Impediments
將”準備好的待辦故事”做到”完成”，並同時移除阻擾工作的障礙，團隊就能成功



scruminc.

What does it mean to be READY?
準備好是什麼意思？

Defined clearly enough that all members of the team understand what must be done

非常清楚地定義，才能讓團隊所有成員知道必須做什麼

- Includes team-developed tasking, if needed 包括團隊開發的任務
- Assume some ongoing discussion to refine, coordinate and clarify
  繼續不斷的討論以利優化、協調和澄清

Includes clear statement of resulting business value that allows the Product Owner to prioritize

包括明確地陳述商業價值，以利PO做優先排序

- Includes any required **enabling specs**, wire frames, etc.
  包括任何要求的準備以利完成工作的規格、線框圖等等

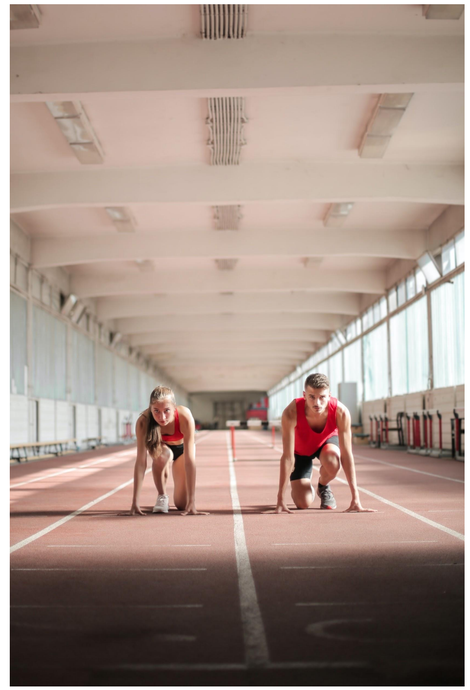
Fully meet INVEST criteria for user stories

完全符合用戶故事之INVEST準則

- Estimated and sized to complete easily within one Sprint
  估算好以利於一次的Sprint中順利地完成

Free from external dependencies 沒有外部相依性

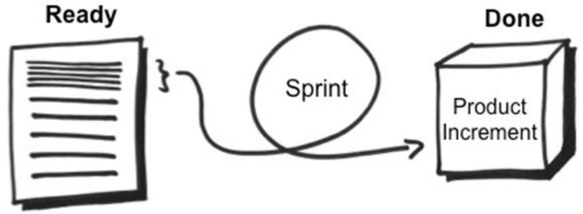
- I.e. there is nothing beyond the team's control that must be done first in order to complete the story 就是所有必須先做的事都在團隊的掌握中，以利故事的完成



READY Backlog

- 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
- **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** 團隊如何在Sprint展示會議中展示

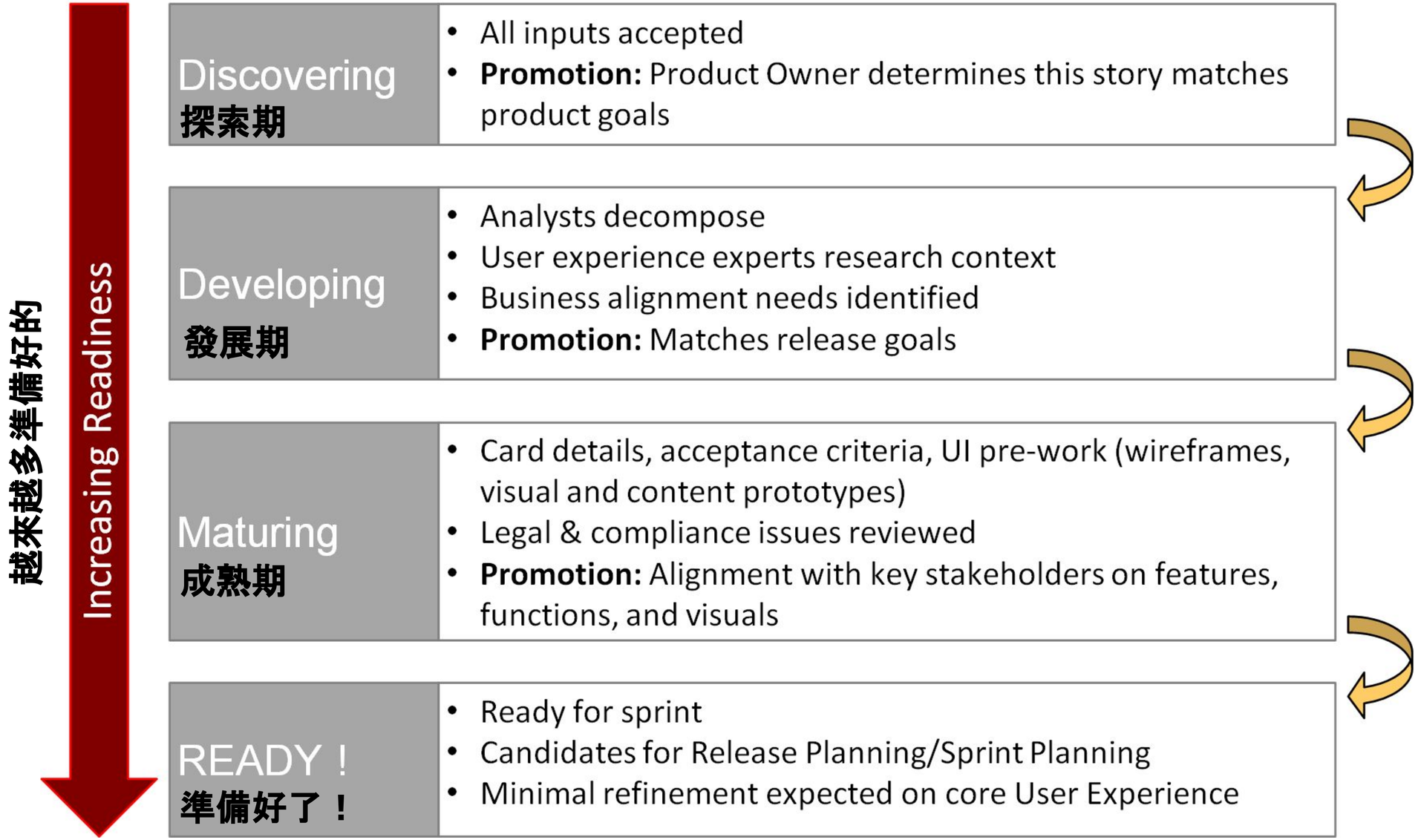
Default Definition of Ready (DoR)



[Scrum Pattern: Ready Backlog]

scruminc.

User Story Readiness Progression 準備就緒的進度



scruminc.

What does it mean to be DONE?
完成意味著什麼？

- "Definition of Done" (DoD) decided on beforehand – along with acceptance tests
  - **DoD** can be standard **across a group of common stories**
    完成的定義可以作為一組共同的故事的標準
  - **Acceptance Criteria** can be **specified for unique ones**
    驗收標準可以為個別的故事的標準
- Done means the feature has been developed, tested AND meets all required acceptance tests
  完成表示功能已開發，已測試並符合所有必需的驗收測試
- Ideally, **Done means the feature could be shipped** to a customer 理想情況下，完成是指可以將該功能交付給客戶
- Product Owner **officially "accepts"** Done features from Team at the Sprint Review meeting
  PO在Sprint展示會議"正式接受"團隊完成的功能

scruminc.

Examples: Definition of Done 完成的定義舉例

Default Definition of Done
預設"完成的定義"
- Releasable 可發布的

Default Definition of Done
- Unit/Integration tested 已單元/整合測試
- Ready for acceptance test 準備好驗收測試
- Deployed on demo server 已配置在展示伺服器

Default Definition of Done
- Acceptance tested 已驗收測試過
- Release notes written 已寫好發布記錄
- Releasable 可發布的
- No increased technical debt
  沒有增加的技術債
  (I haven't messed up the codebase or cut corners on quality)
  (我還沒搞砸代碼或在品質上偷工減料)

What else must be done before shipping the code? 交付之前還必須做些什麼？

- For example "customer acceptance test + user documentation" 例如客戶驗收測試+用戶手冊

Why not? Who does it? When? What happens if a problem turns up?

為何不?誰做? 何時? 如果出現問題會怎樣?

Burn up this work in release burndown! 在發布燃盡做完這些事！

Source: Henrik Kniberg

DoR and DoD Exercise

Come up the Definition of Ready and Definition of Done for your team (20 mins) 為您的團隊提出「準備好」和「完成」的定義（20 分鐘）

- Together with your team 與您的團隊一起
- **List what will be your team's DoR and DoD** (10 mins) 列出您團隊的 DoR 和 DoD（10 分鐘）
- Scrum Master or someone **shares your DoR and DoD** (3 mins for each team) to the class 向全班分享您團隊的DoR和DoD（每個團隊 3 分鐘）

Question: 問題

- Does your DoR and DoD include everything in the first version? 你們第一個版本中的DoR和DoD是否包含所有內容？
- Is your DoR and DoD **clear to everyone on your team**? 團隊中的每個人都清楚你們的DoR和DoD嗎？
- Does your DoD **follows the corporate quality standard**? 你們的DoD是否遵循企業品質標準？

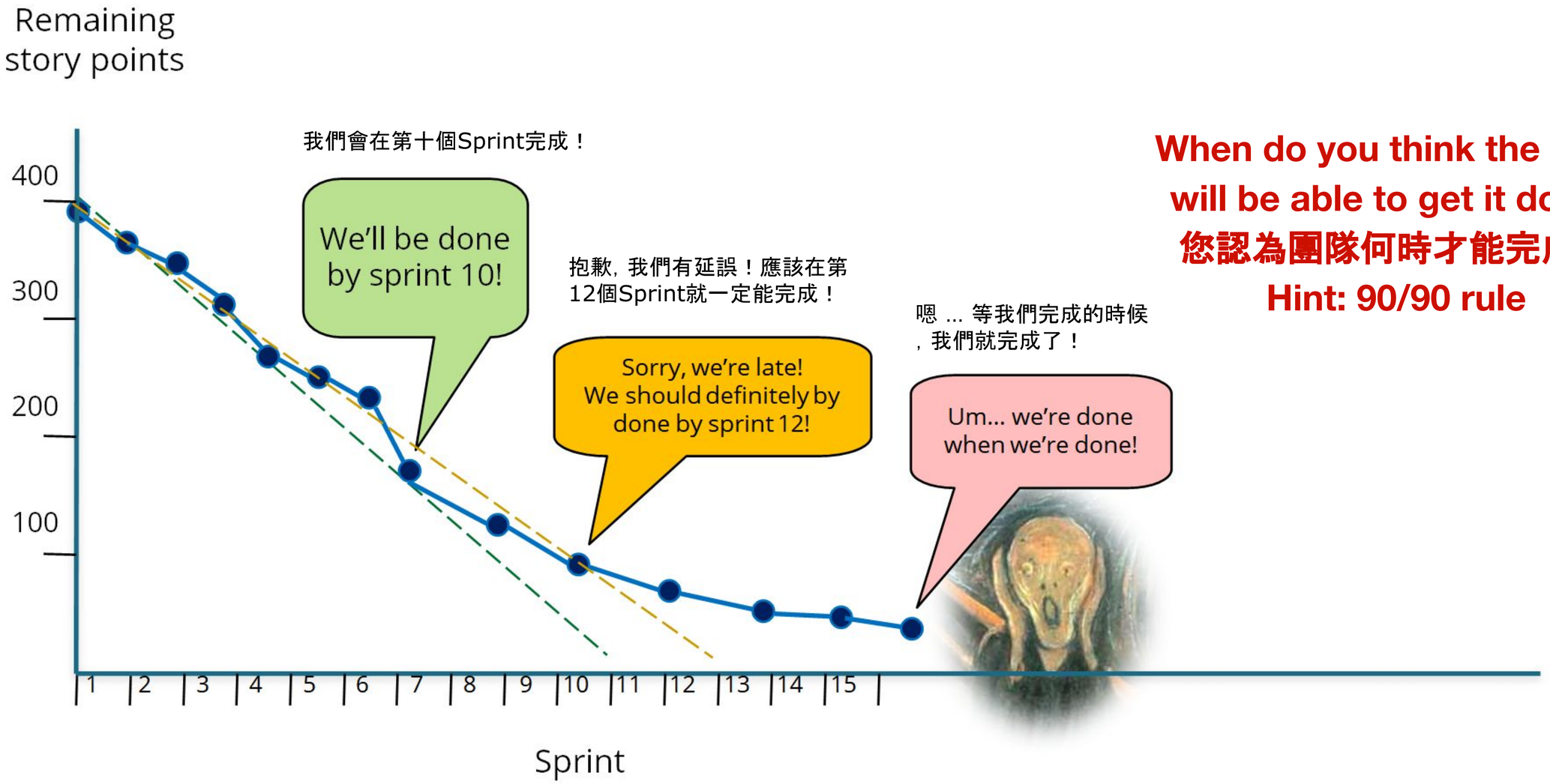
Technical Debt 技術債

As a Product Owner, I need a simple system for managing Technical Debt
so that it does not build up and slow the team down
身為Product Owner，我需要一個簡單的系統來管理技術債，
所以技術債不會持續堆疊並讓團隊慢下來

Technical Debt 技術債

- **The structure of the product blocks progress** (example: monolithic vs microservice)
  產品的結構會阻擋進展
- **Symptoms 症狀:**
  - The product **cannot easily be refactored** to meet market demands
    產品無法輕易重構以滿足市場需求
  - Too many **dependencies** between components, bad architecture
    組件之間的依賴過多, 架構不良
  - Too many **defects**, poor construction
    缺陷太多, 建構不佳
  - Product is **hard to understand and hard to change**
    產品難以理解且難以更改
- **Diagnosis 診斷:**
  The product is not lean!
  產品不精實

Technical Debt & Release Planning
技術債與發布計劃

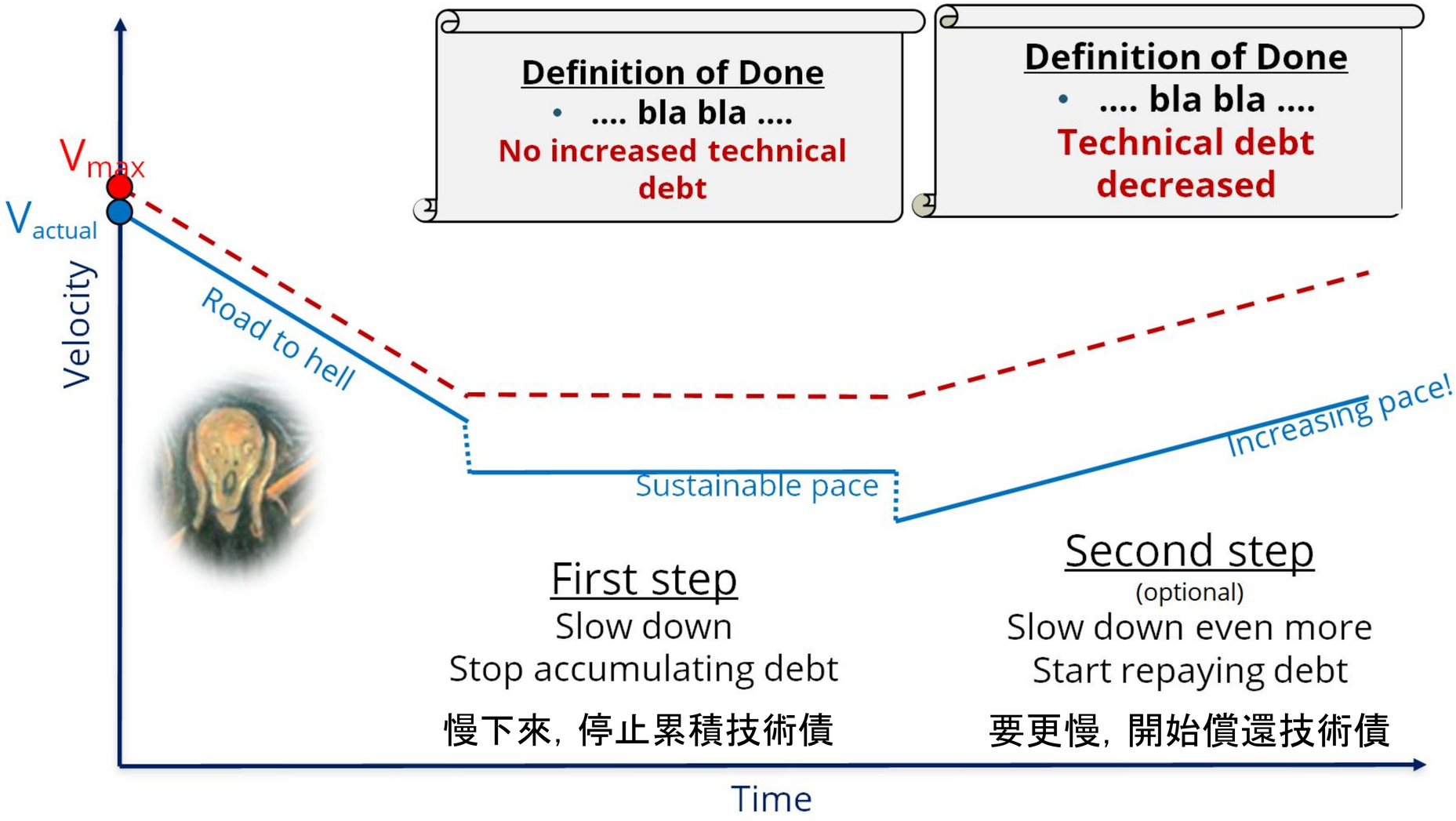


When do you think the team will be able to get it done?
您認為團隊何時才能完成？
Hint: 90/90 rule

Source: Henrik Kniberg

scruminc.

Dealing with Technical Debt 處理技術債



Source: Henrik Kniberg

Product Owner Role in Events
PO在Scrum事件扮演的角色 (*)

As a PO, I need to understand my role in the Scrum events,
so that I can help make them as effective as possible
身為Product Owner，我需要了解我在Scrum會議所扮演的角色，
所以我才能讓會議盡可能有效率

Roles in Sprint Planning
在Sprint Planning中的角色

Product Owner

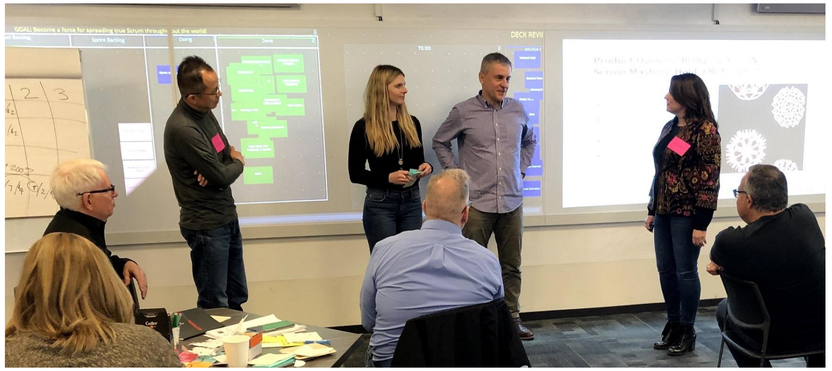
- Present the backlog 介紹待辦事項
- Answers questions to clarify the backlog 回答問題
- Identify highest priority 確定最高優先排序

Scrum Master

- Facilitate the meeting 引導會議
- Confirm team capacity 確認團隊產能
- Ensure stories are ready and have a Definition of Done 確保故事已準備好並有完成的定義

Developers

- Ask questions 提問
- Decide how much backlog to pull into the Sprint 決定該Sprint的待辦事項
- Agree on a Sprint goal 贊同Sprint目標



Sprint Planning from PO's Point of View

- Review velocity and **set Sprint capacity** 檢視速率並設定Sprint的工作量
  - Adjust for vacations, holidays, etc. 就休假或假日等而作調整
- **Forecast** how many User Stories will be completed in the Sprint using **Yesterday's Weather** 利用"昨天的天氣"來預估Sprint能完成的用戶故事(點數)
  - The team **commits to do the best** they can to **hit the forecast**
  - There will be **normal variation** (about +/-20%) 正常變動為上下20%
  - This variation means that sometimes we will be late, sometimes early
- **Set the Sprint Goal** as a team 團隊一起設定Sprint目標
- Review **ONLY Ready** stories 只審查準備好的故事
- Finalize **Acceptance Criteria** 確定驗收標準
- Ensure **uniform understanding of Definition of Done** 確保對"完成的定義"有一致相同的理解

Product Owner in Daily Scrum

- Be there 出席
- **Answer questions**, particularly about Sprint Backlog
  回答所有問題，特別是有關Sprint待辦事項
- Clear **communicating the Sprint Goal**
  清楚溝通Sprint目標
- Help the Scrum Team identify and remove impediments
  協助Scrum團隊發現並消除障礙

Product Owner in Refinement

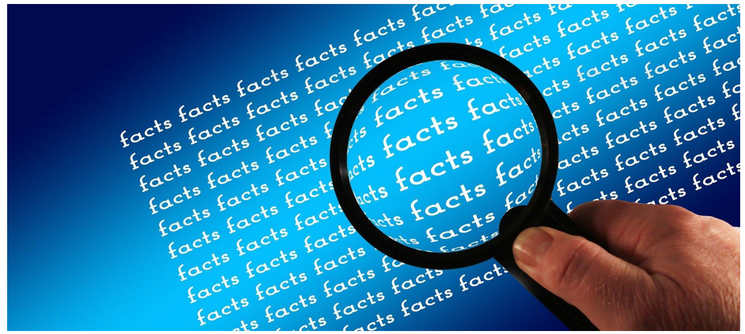
- Work with Scrum Master to setup refinement meetings
- Help the team **break down** and estimate stories
  協助團隊優化並估算故事
- Make sure **stories are Ready for the next Sprint**
  確保下一個Sprint的故事準備好了

Product Owner in Sprint Review

- **Collaborate** with Scrum Team and Customers/Stakeholders during the Sprint Review and get Team members to help
  與Scrum團隊和客戶/利益相關者協作並得到團隊協助
- **Invite** Customers / Stakeholders 邀請客戶/利益相關者
- **Review stories that are Done** 檢閱已完成的故事
- Have **Team demonstrate** a potentially shippable increment
  由團隊展示可交付的增量
- Determine velocity of the Team by **formally** deciding what stories are done
  正式決定哪些故事是完成的，用來決定團隊的速率
- **Get feedback** from Stakeholders and **update Product Backlog**
  取得利益相關者的回饋，並更新產品待辦事項

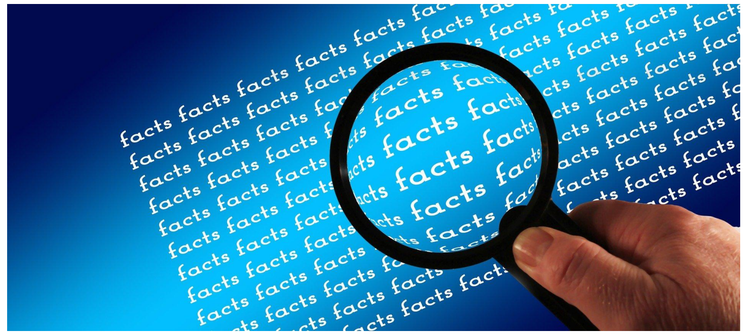
Product Owner Fact or Fake? 真相或捏造?

- Each person has 30 seconds to move 1 item in the correct FACT or Fake circle
  每個人用30秒在Fact or Fake移動一個便利貼
- Once someone moves it, don't move again
  有人移動過的便利貼，就不能再次移動
- After all items are placed, compare with the correct result 所有便利貼就定位後，和正確的結果作比較
- Discuss any question



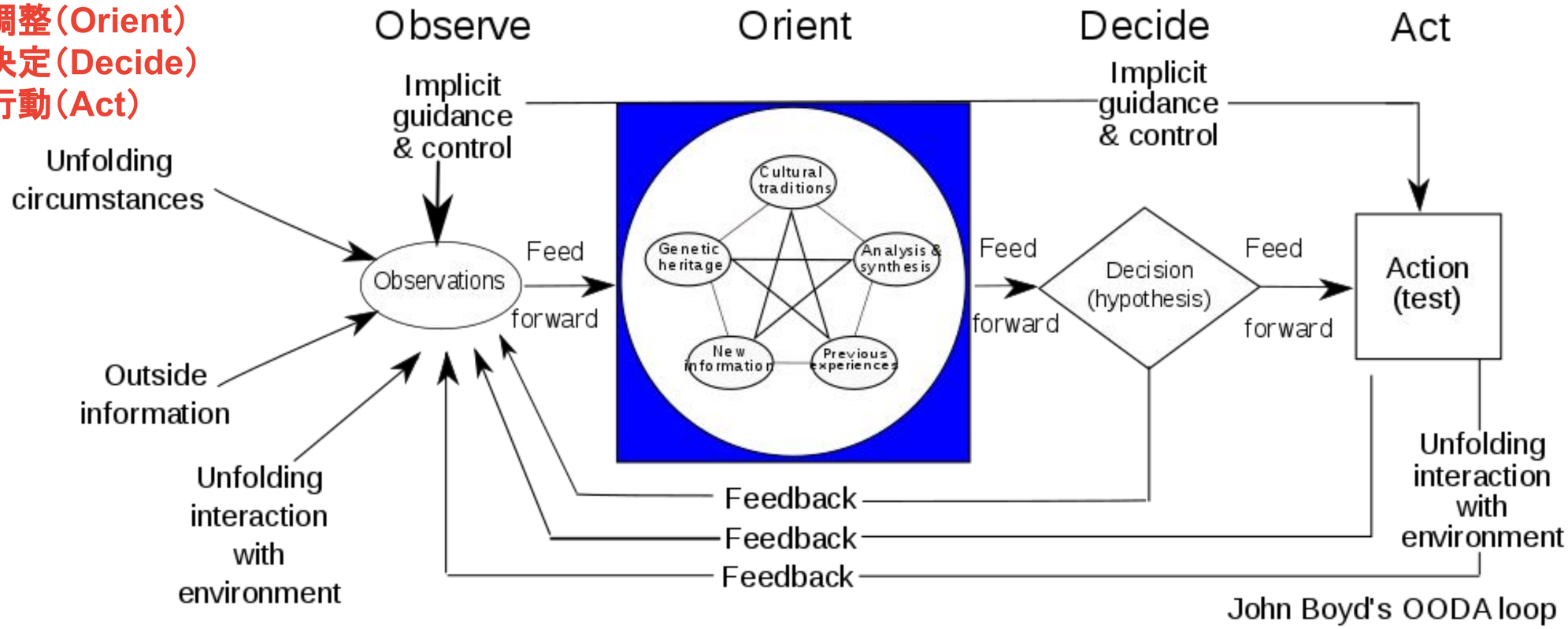
Product Owner Fact or Fake? 真相或捏造?

- Fact: 4, 8, 9, 10, 12, 13,
- Fake: 1, 2, 3, 5, 6, 7, 11, 14



Using OODA loop to be a Great Product Owner…

觀察(Observe)
調整(Orient)
決定(Decide)
行動(Act)



Delivers: The right product set to excite customers, At the right time, In the order that maximizes business value 在正確的時間，提供讓客戶興奮的正確產品

Responds: Dynamically to change faster than competitors 比競爭對手更快

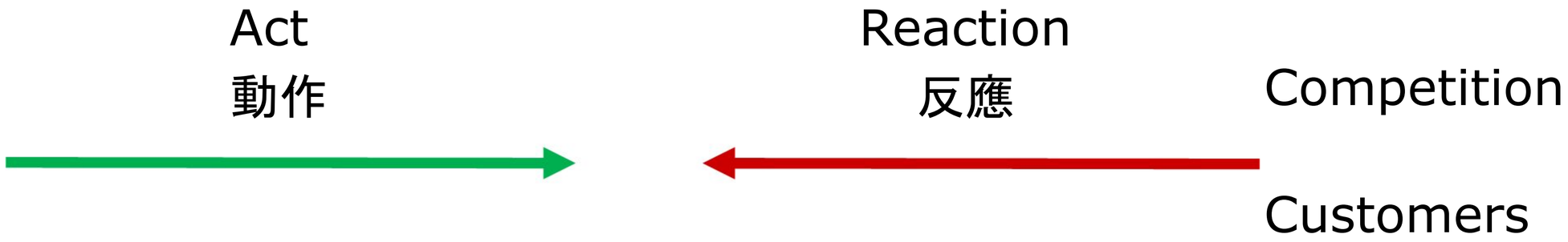
Identifies and clarifies: Remove uncertainty and maximize team velocity and value creation 消除不確定性並提高團隊速率和價值創造

A Great Product Owner fundamentally changes an organization's path by winning in the market!

偉大的PO經由在市場上贏得的成功，徹底改變了組織的路線/方向！

scruminc.

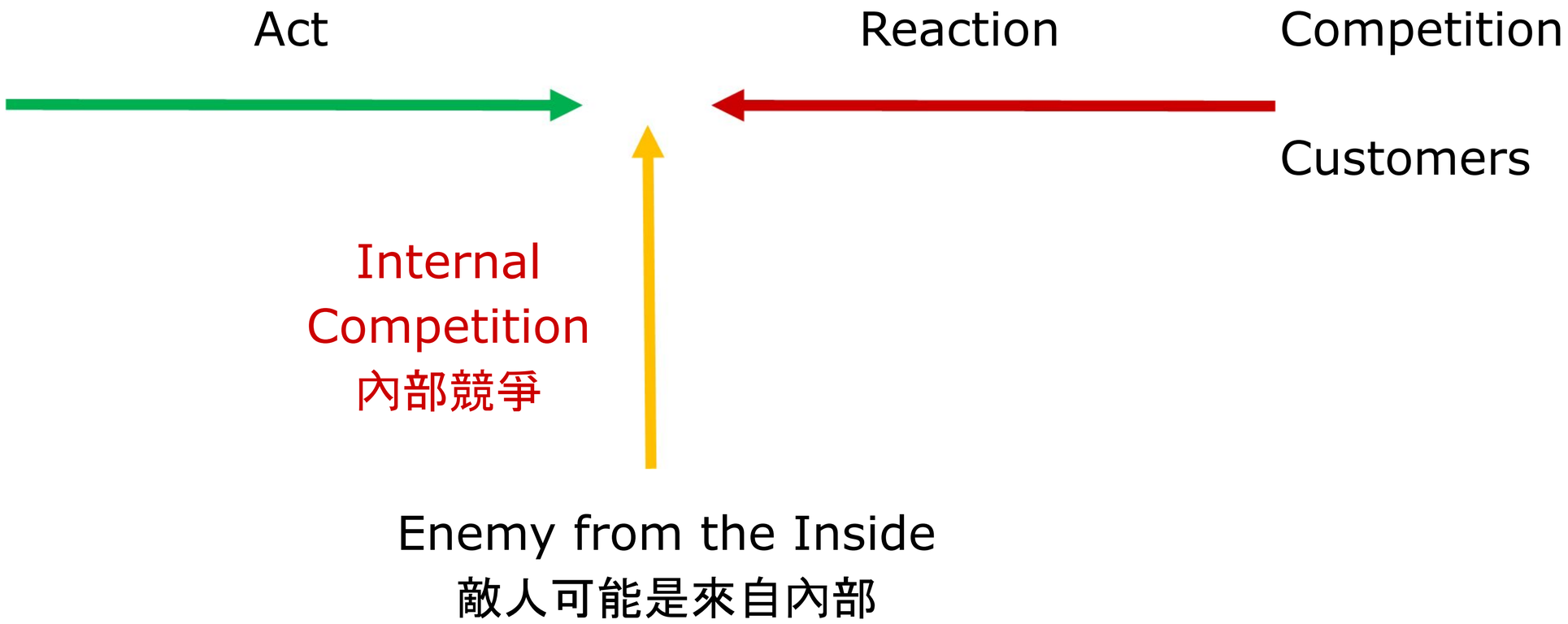
Systems Problems: Every Action Creates a Reaction
每一個動作都會產生反應



Act and Reaction creates Transaction and **Transaction creates Change**.
動作和反應創造交易，交易創造改變。

Example: iPhone generates Reaction from Nokia

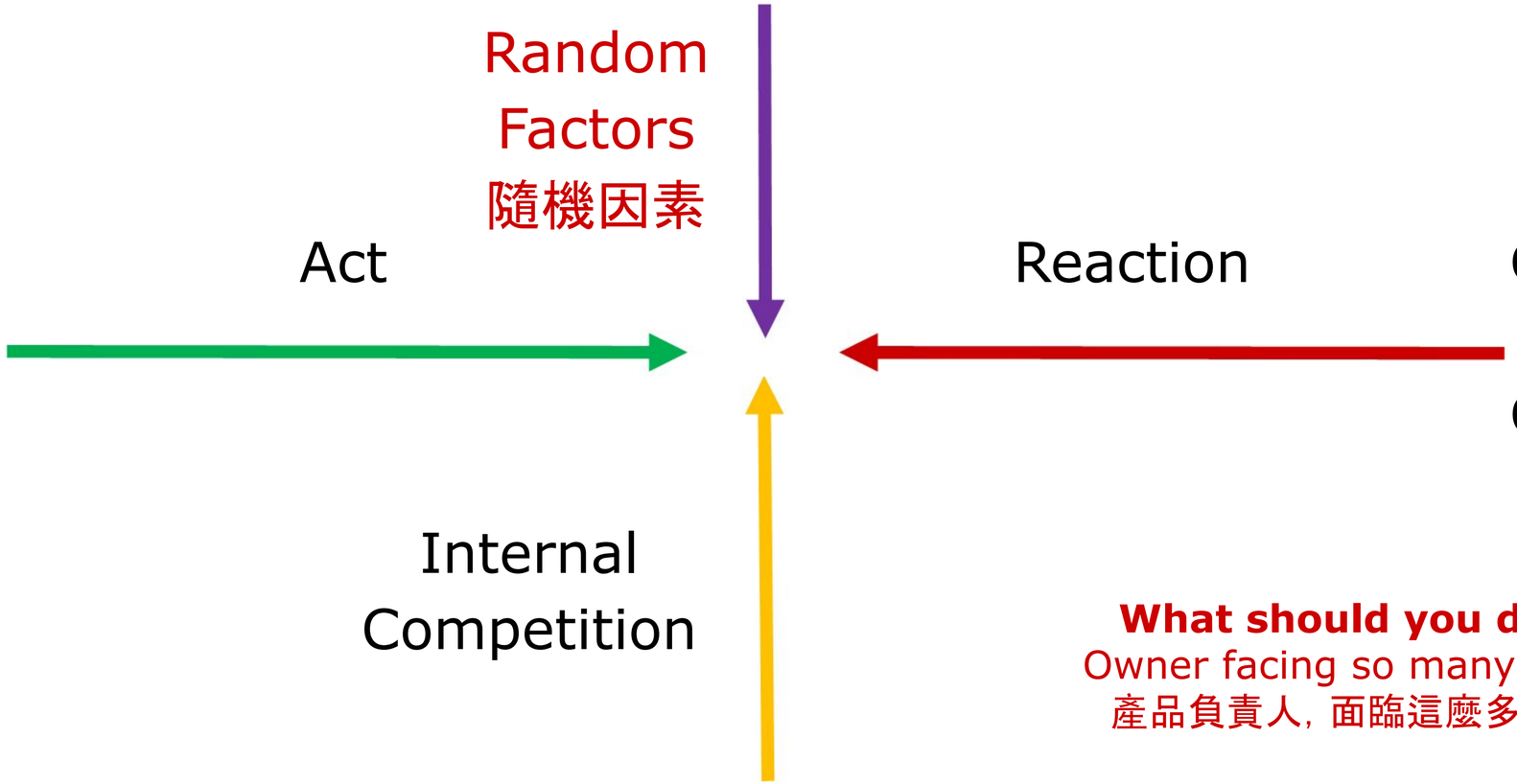
Systems Problems: Every Action Creates a Reaction



Systems Problems: Every Action Creates a Reaction

People get sick or leave; Technology doesn't work; Customer changes their mind

隊員生病或離職；技術面行不通；客戶改變想法



What should you do as a Great Product Owner facing so many challenges? 身為優秀的產品負責人，面臨這麼多的挑戰，你應該怎麼做？

scruminc.

John Boyd's Solution: The OODA Loop (OODA循環)

Observe is to know what is going on now. What is the market looks like? What is the opportunity? What is coming? What is your competitor doing? (or what is the unexpected competitor?) What is the new customer behavior?
It's about what data/information that you collect (about the market, about the competitor…)

觀察現在發生了什麼事。市場是什麼樣？機會是什麼？即將發生什麼事？你的競爭對手在做什麼？新的客戶行為是什麼？
您收集哪些數據/資訊(關於市場、關於競爭對手...)

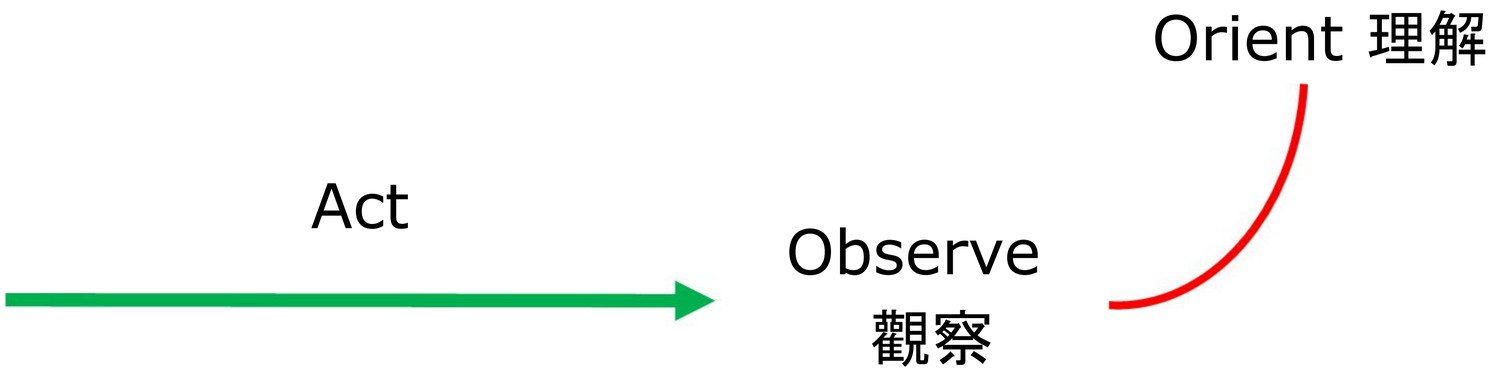
Policy Resistance: OODA Loop Systematically Wins

Quickly strike from an unexpected direction

快速地從未料及的角度切入

The reactive forces will try to realign

GET INSIDE THEIR DECISION LOOP

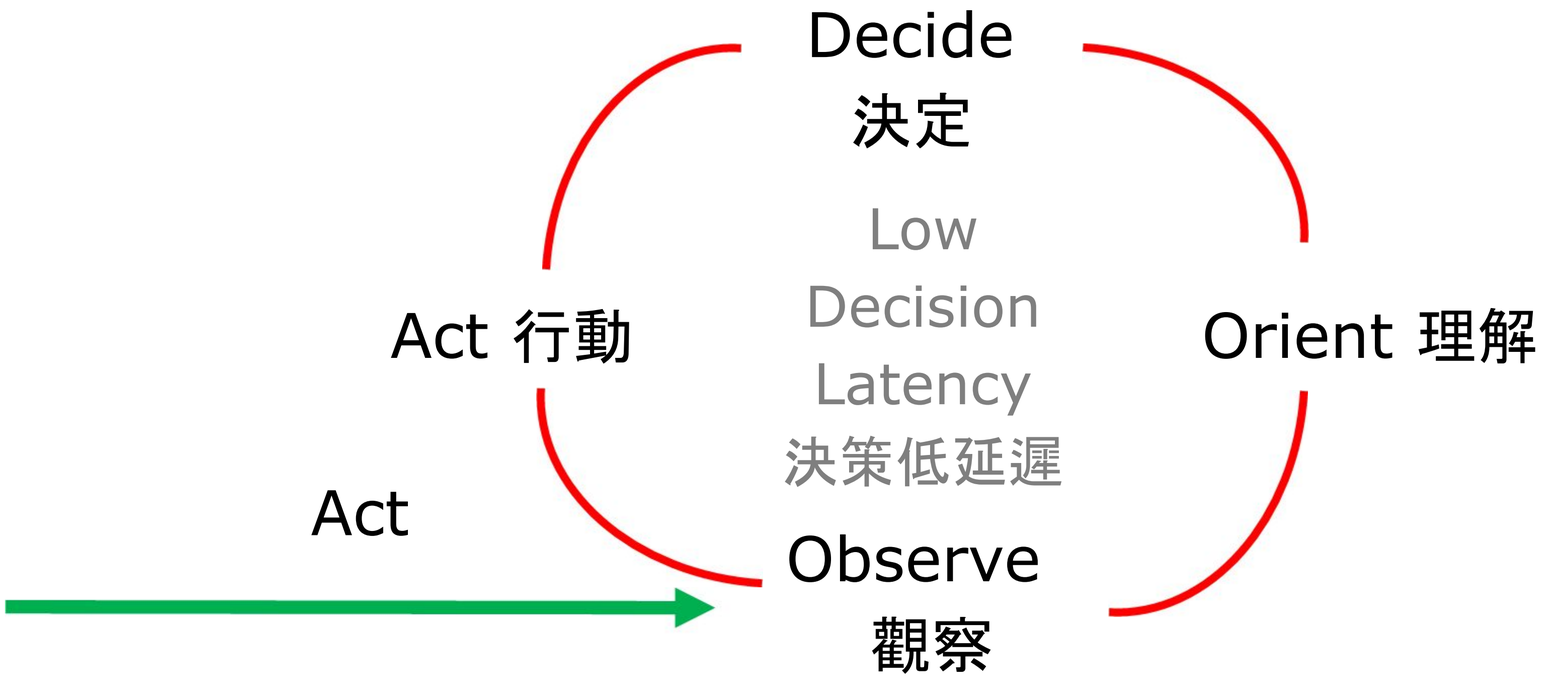


Orient is your current assessment of the overall situation based on **your belief, your value, your prior experience and knowledge.**
理解是你根據你的信念、你的價值觀、你先前的經驗和知識對當前整體狀況的評估。

"To Orient" is to figure out **where you are related to where you want to be** (the goal or destination). This is the same as "Situational Awareness".
「定向」就是弄清楚你與你想要去的地方（目標或目的地）的關係。

Policy Resistance: OODA Loop Systematically Wins

Quickly strike from an unexpected direction
The reactive forces will try to realign
GET INSIDE THEIR DECISION LOOP



Decide on **what product we should focus first** and **what feature to build first.**
決定我們應該先專注於什麼產品以及首先要建立什麼功能。

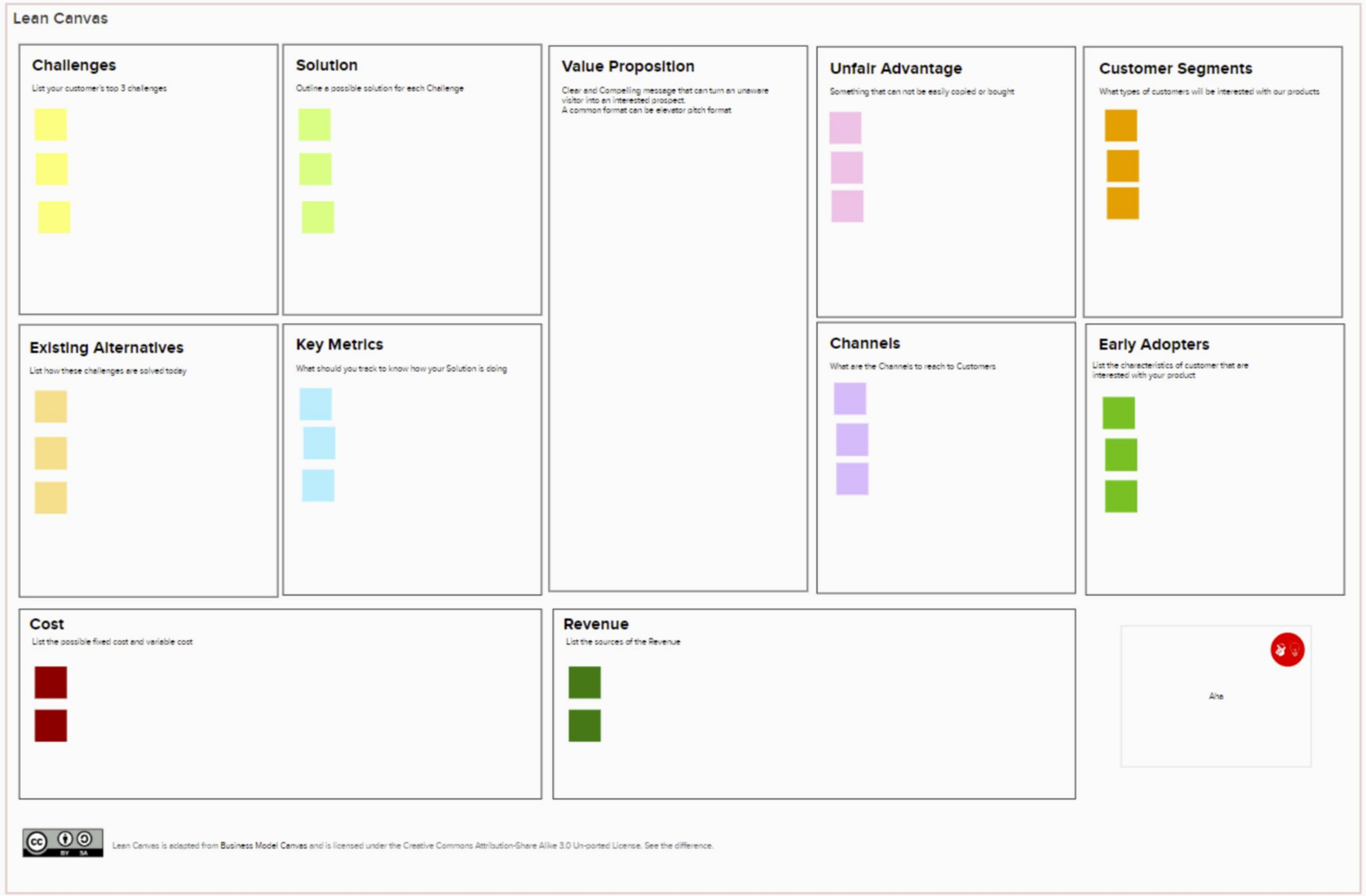
Act: Execute the Decision as soon as you make the decision
行動：一旦做出決定就立即執行

Testing your hypothesis, get feedback, then quick adjust
The OODA loop keeps going and going until you win.
測試你的假設，獲得回饋，然後快速調整
OODA循環不斷進行，直到您獲勝

scruminc.

From Idea to a concrete Action

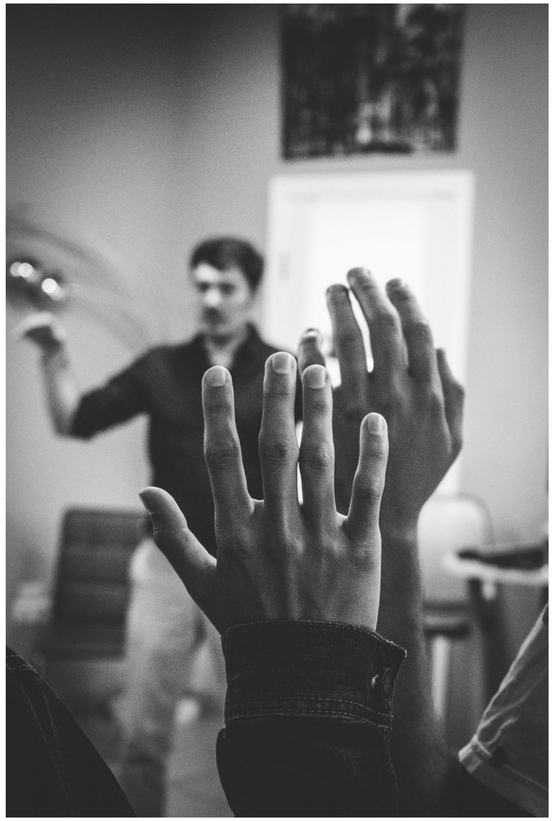
挑戰
解決方案
價值主張
不公平的優勢
消費者區隔
現有的其他選擇
主要衡量指標
通路
嚐鮮者
成本
收益



課前作業: Teach the Scrum Patterns

- Each team picks two patterns (avoid picking the same patterns), please see your team’s Mural (under Scrum Patterns) for the links
  每組挑選兩個patterns(請各組避開相同的patterns)
- Read it, understand it, and when we are back tomorrow
- **Your team will have 10 minutes to teach everyone** what these two patterns are **(Be creative, Be yourself, and Be ready)**
  下次上課，每組有10分鐘教大家兩個patterns
- Here is the link to the patterns

Questions?



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

Day 2 Course Feedback 課程回饋

Go to

www.menti.com

Enter the code

2636 0394

Or use QR code

Release Planning 發布計劃

As a Scrum Master or Product Owner, I need to help with Release Planning,
so that all stakeholders know when to expect what they want
身為Scrum Master或Product Owner，我需要協助發布計畫，
所以利益關係者才能知道他們要的能何時交付

What did I learn yesterday? 昨天我學到了什麼？

What did I learn yesterday? 昨天我學到了什麼？

Go to

www.menti.com

Enter the code

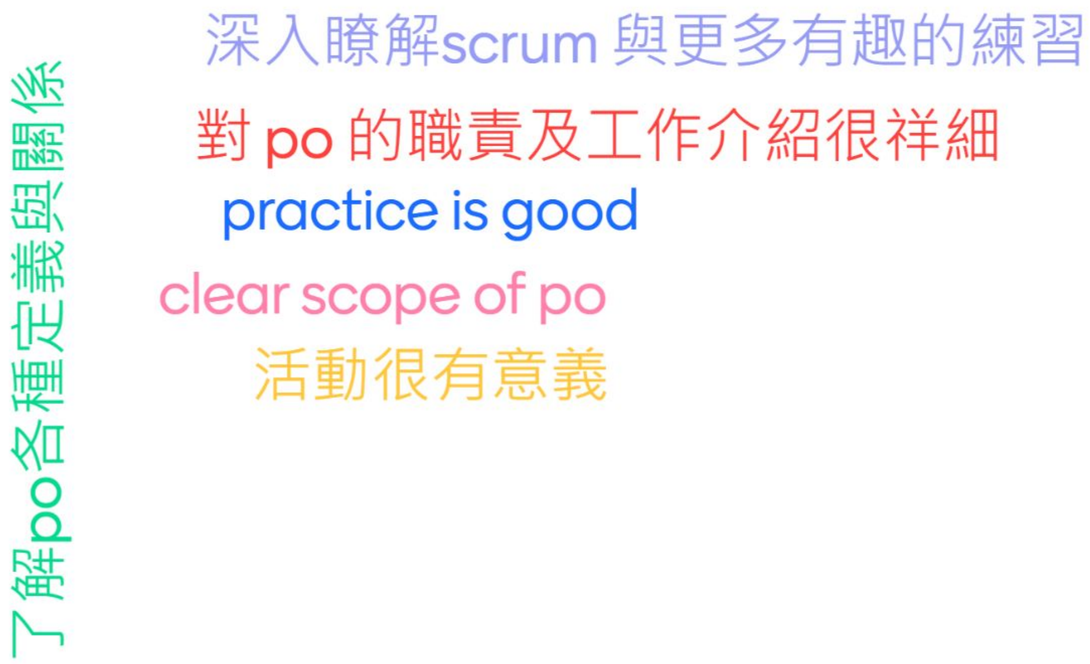
1693 1577

Or use QR code

Course Feedback Results 課程回饋結果

What we did great 我們在哪些方面做得很好

6 responses



6

scruminc.

Course Feedback Results 課程回饋結果

What we can do even better (Improvement)? 我們可以在哪些方面可以做得更好

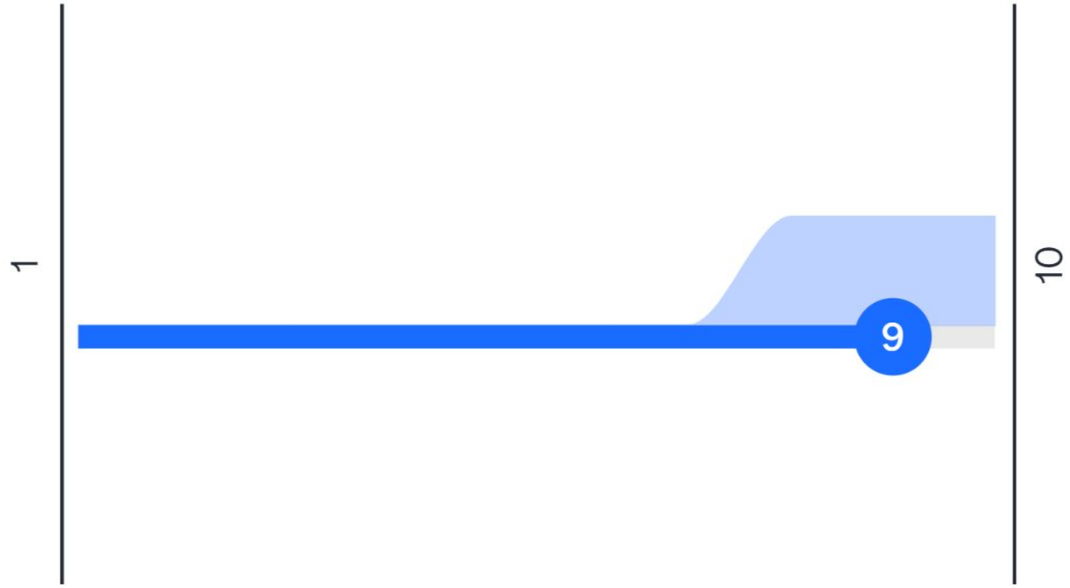
2 responses

less words in 1 page

clear exercise

Course Feedback Results 課程回饋結果

1 to 10 (10 means meeting your expectation) 1到10（10表示滿足您的期望）



Recap and Questions from yesterday
昨天的回顧和問題

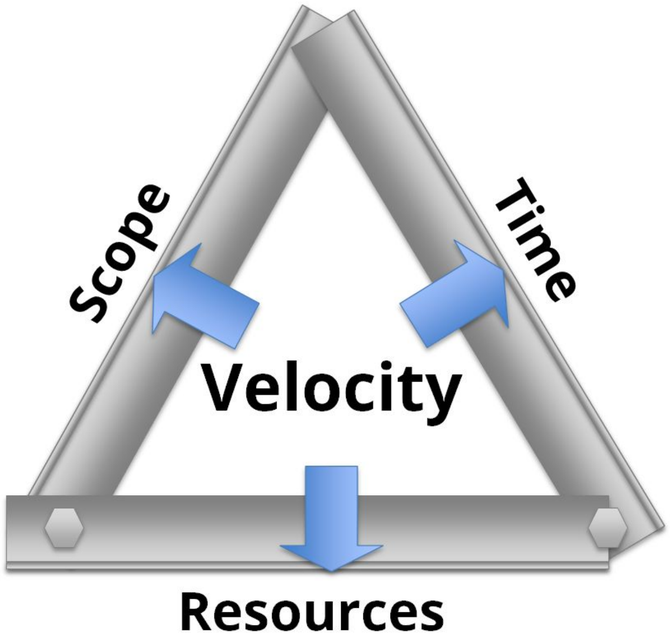
- Any question(s) from yesterday? 昨天還有什麼問題嗎？
  - Sprint 0 https://www.youtube.com/watch?v=0hRZffDD1ec&t=2066s
  - What else? 還有什麼其他問題嗎？
- Recap what we learned 回顧我們所學到的知識
  - **Your Scrum vs The Scrum Framework**
  - **Your team's Scrum cadence** 您團隊的Scrum節奏
  - We look at **different Scrum board** and how to use Scrum's data to prepare for your Status Report 我們研究不同的Scrum板以及如何使用Scrum的資料來準備您的Status報告
  - We learned the Increment 產品增量 and **MVP**
  - Sprint Backlog 短衝待辦
  - Product Vision 產品願景, **you built a Persona for your product** 您為您的產品建立了角色
  - We look at different ways for prioritization 我們研究不同的優先排序方法
  - You created your **first version of DoR and DoD** 您建立了第一個版本的 DoR 和 DoD
  - We learned the technical debt 我們了解了技術債 and **a Pattern to reduce technical debt** https://sites.google.com/a/scrumplop.org/published-patterns/product-organization-pattern-language/development-team/team-sprint
  - And what does PO role in different Scrum events PO 在不同的Scrum事件中扮演什麼角色

Scrum Allows us to Break the Iron Triangle
Scrum讓我們打破鐵三角

Complete More Scope in Less Time with Fewer People
用更少的人在更少的時間內完成更多

Traditional planning views **Scope, Time and Resources** as locked in a fixed relationship
傳統規劃將範圍、時間和資源視為固定關係

- In theory, any dimension can change to meet release requirements…
  任何一點都可以更改以滿足發布的要求
- …But, in practice **resources seen as easiest to change**, while scope & time seen as fixed constraints
  但是實際上，**資源被認為是最容易更改的**，而範圍和時間卻被視為固定的限制

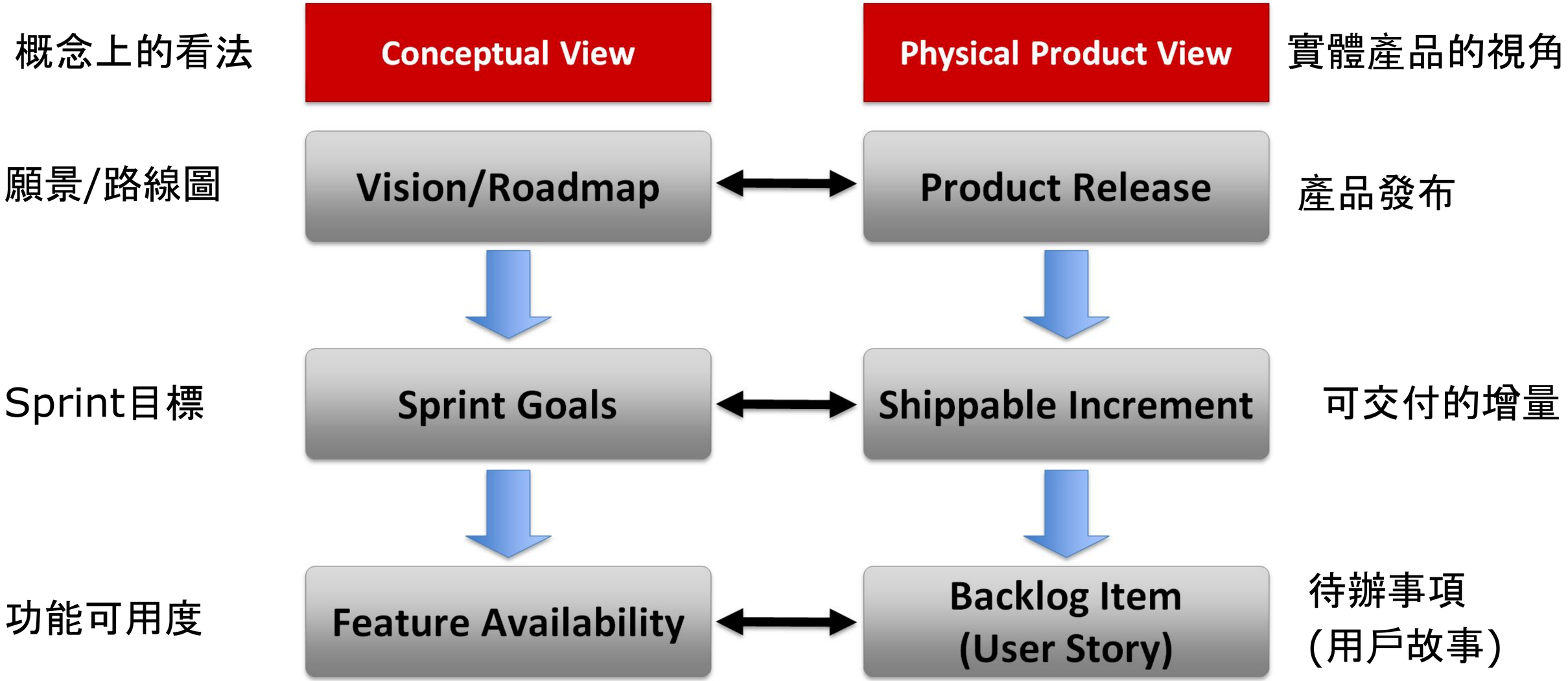


But in Scrum we find:
但是在Scrum，我們發現

- **Small & stable teams are key**
  小而穩定的團隊是關鍵
- **Flexing scope** actually much easier than changing resources
  彈性調整範圍比改變資源(人)要容易得多
  - Requires scope defined as independent features, and prioritized by value
    需要將範圍定義為獨立功能，並按價值優先排序
- **Increasing velocity** allows team to get more done in the same time
  不斷提高的速率使團隊可以用相同的時間完成更多工作
  - Accomplished by **removing impediments**
    經由消除障礙後完成
  - Sustainable working schedule, **not longer hours**
    可持續的工作排程，而不是更長的工時

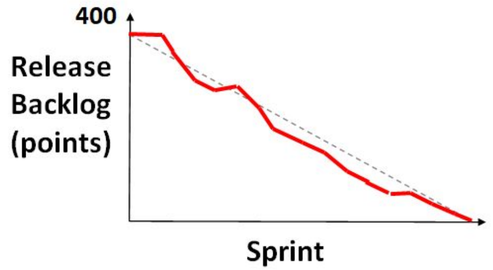
scruminc.

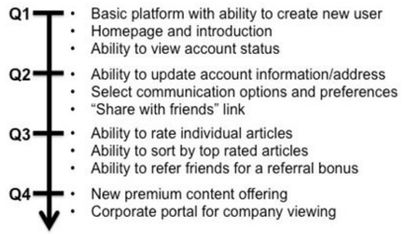
Release Plan Links Concept and Physical Worlds
發布計劃串聯概念和現實世界



Elements of a Scrum Release Plan 發布計劃的要素

- **Clear Vision** 清晰的願景
  - Tied to concrete business value 與具體商業價值建立連結
  - Aligns stakeholders 與利益相關者密切合作
- **Vision Decomposed Into Independent Features**
  願景拆分成獨立功能
  - Prioritized and estimated 優先排序和估算
  - ROI and customer need driven
    投資報酬率和客戶需求導向
- **Burndown chart of progress on prioritized backlog items**
  - Measured in Points! 以點計算
- **Feature availability timeline**
  - Best guess – subject to change
    最佳猜測-可能改變

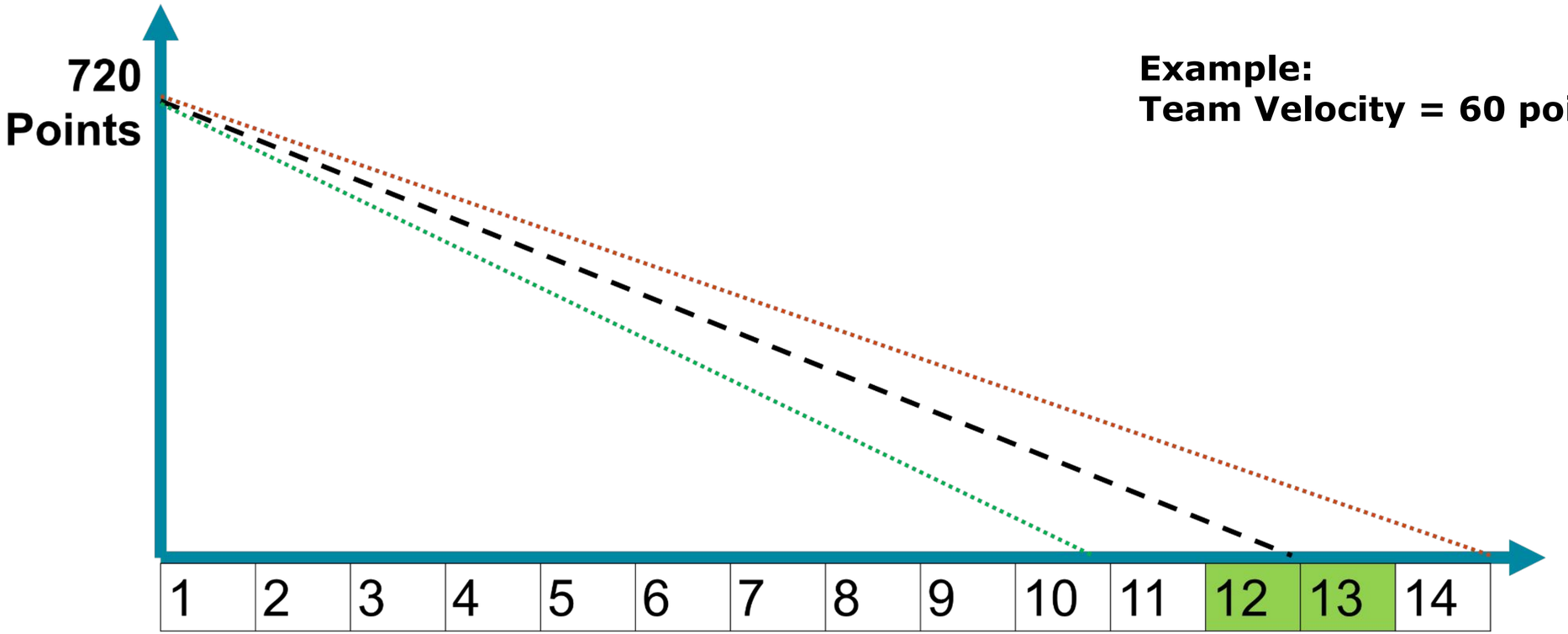




scruminc.

Release Planning: How long will it take?

發布計畫:這要做多久?

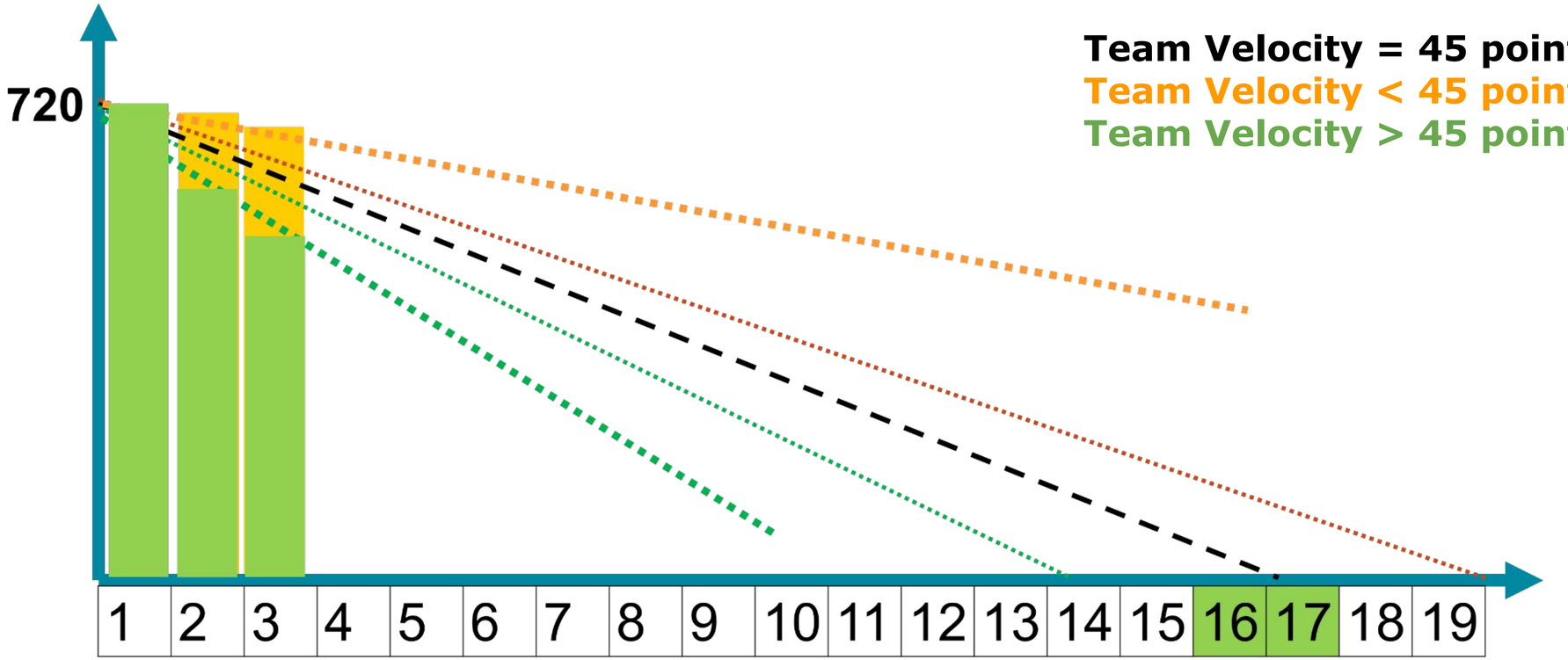


Example:
Team Velocity = 60 points

Release Planning: Work Profile 工作數據圖

Velocity 速率	100%	60 points
Kaizen 改善	5%	3
Tech Debt 技術債	10%	6
Interrupt Buffer 插件緩衝	3%	2
Emerging Work 新工作	5%	3
Other 其他	2%	1
Total 合計	**25%**	**15**
New Development 新開發	**75%**	**45**

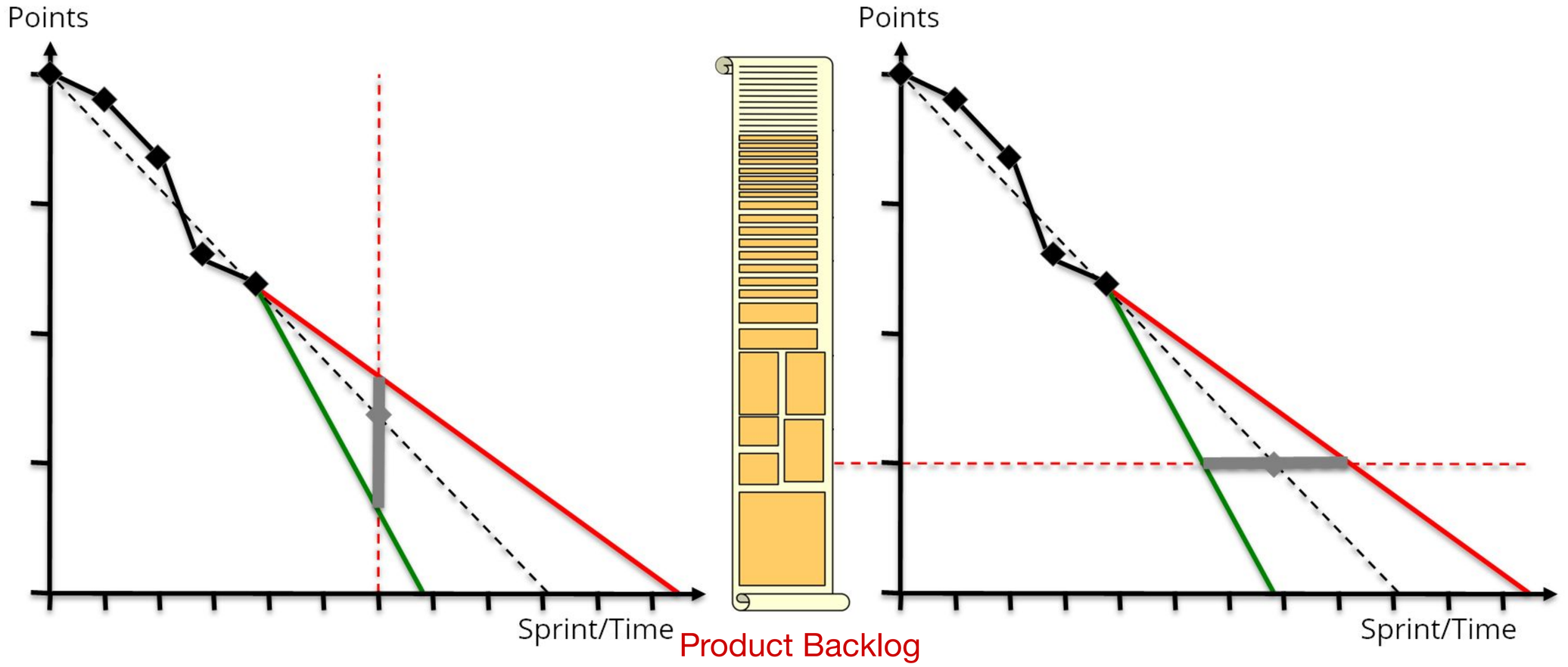
Release Planning: How long will it take?



scruminc.

When Can You Ship? 您什麼時候可以交付？

Release Burndown Chart Makes it Visible 發布燃盡圖讓您看到



What will you deliver by Month X?

When will you deliver Features 1-15?

scruminc.

Three More Considerations for Anticipating Burndown
預期的燃盡還有三件需要考慮的事情

Undone Work
未完成的工作

When the **Definition of Done is not equal to shippable product**, there is more work that must be estimated

當完成的定義不等於可交付的產品時，表示有更多的工作必須被預估

Emerging Requirements
新的要求

Additional user stories beyond those known in the backlog that are **"discovered"** as the project evolves and require the team to do more work

額外的用戶故事在專案進展時被"發現"，團隊就需要做更多

Bugs and Customer Feedback
錯誤和客戶回饋

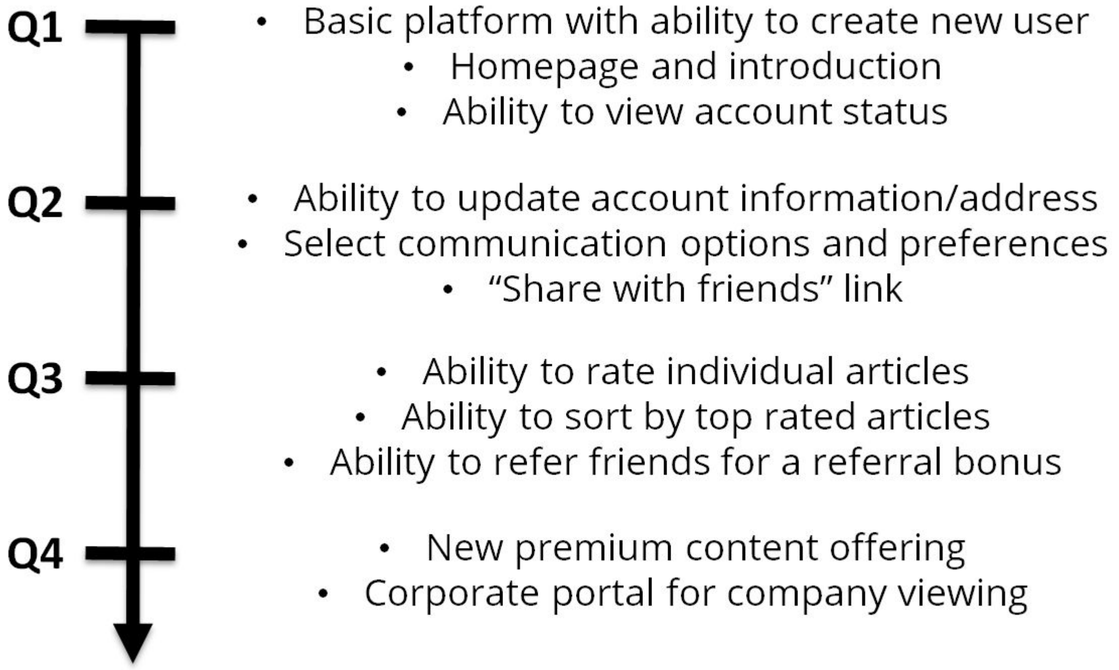
Additional work that cannot be anticipated in the release plan, but you know **it will come up as product functionality is released**

在發布計劃中無法被預期的額外工作，但您知道這些工作會在產品功能發布時出現

All factors must be accounted for to determine accurate burndown
所有因素必須考慮以確保燃盡圖的準確

Roadmap Helps Stakeholders Know when to Expect New Features
路線圖可幫助利益相關者了解何時會有新的功能

- Facilitates conversations on **feature priority**
  引導對於功能優先排序的對談
- **Aligns stakeholders** and avoid distraction
  協調利益相關者並避免注意力分散
- Ground rule: **Timeline is only a forecast, and subject to change**
  基本規則：時間軸只是預測，可能會改變



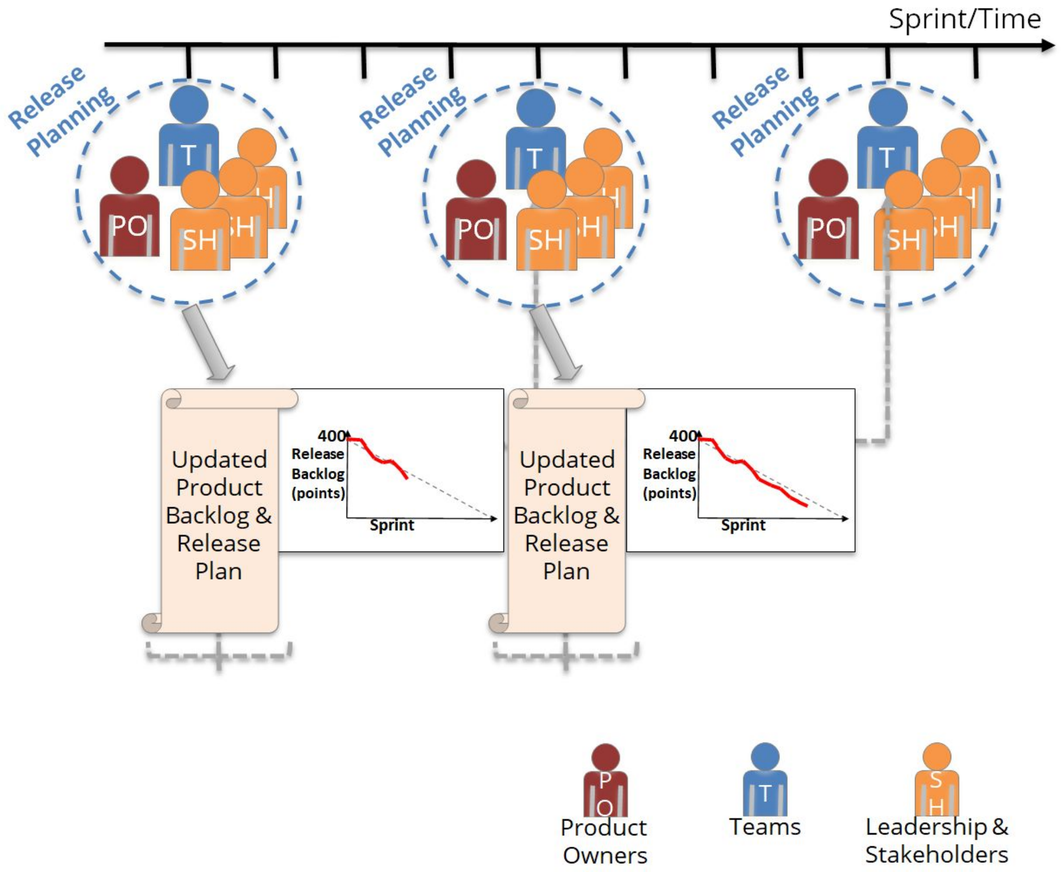
Roadmap Example:
https://portal.productboard.com/useloom/1-loom-s-product-roadmap/tabs/3-coming-soon

Release Planning Meeting 發布計劃會議

Ensures Plan is Regularly Updated with Empirical Data

確保計劃是用經驗數據做定期更新

- Not part of the core Scrum framework…but many teams need it to maintain holistic long-term perspective
  不是Scrum核心架構的一部分...但是許多團隊採用它來維持整體的長期觀點
  - **Planning distributed throughout project** not only at beginning
    計劃會在專案期間持續發佈，而不是只在開始階段
- **Similar to Sprint planning meeting** with product and release rather than Sprint time horizon
  類似Sprint規劃會議有產品及發布，而不是限於Sprint時間期限
  - Held **at regular intervals** 定期舉行
  - **Update release burndown** with latest velocity data
    使用最新的速率數據來更新發布燃盡圖
  - **Adjust Product Backlog as needed** to meet priority deliverables
    根據需要去調整產品待辦事項，以利做到優先順位的交付
- Produces updated release plan and burndown 產出更新的發布計畫及然進圖
- Allows teams to progress confidently
  讓團隊可以自信地向前推展



scruminc.

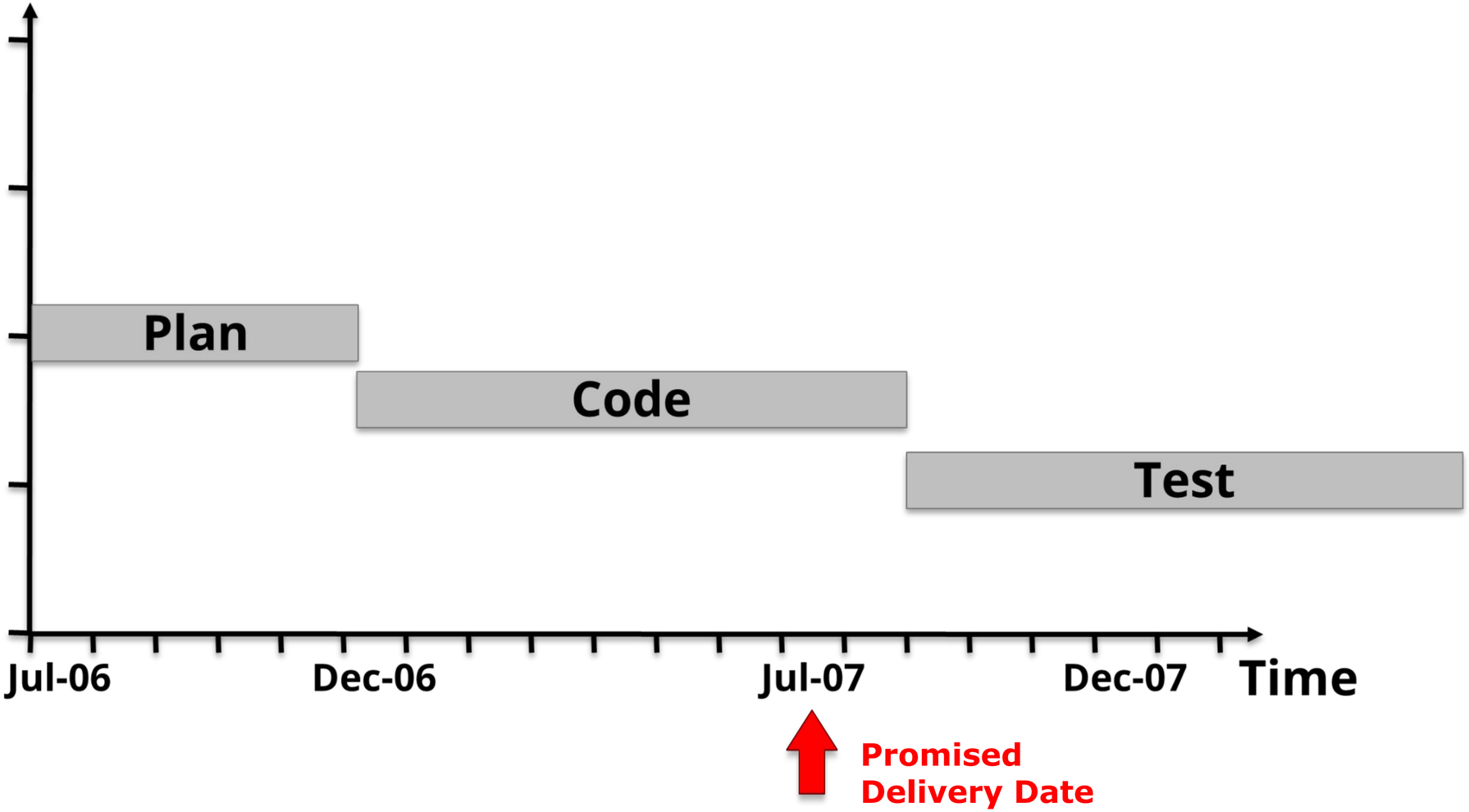
Three Common Approaches to Release Planning
發布計劃的三種常見方法

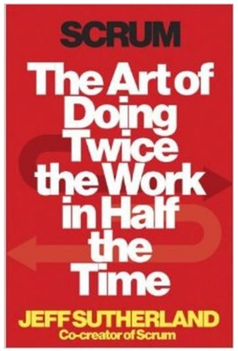
- **Deadline-based** 依據截止日
  - **External deadline** specified for team, they must complete as much of a given backlog as possible before that date 外部截止日，必須在該日期之前盡可能地完成更多
- **Regular-Departure** 定期
  - Set **cadence** of product releases. **(e.g. quarterly)** 設定產品發布的節奏(例如每季)
  - Ready features are included in the release, non-ready ones wait for next release
    準備好的功能會包含在該次的發布，還沒準備好了就等下次的發布
- **Value-Based** 依據價值
  - Team produces incremental potentially-shippable product each Sprint
  - When **PO decides** enough new value has been created, features are released to customers
    當PO確定已創造足夠的新價值時，功能便會發布給客戶

Case Study: Deadline–Based Release Planning

On July 7 2006, Medco CEO promised Wall Street analysts a completely new pharmacy fulfillment system to be implemented by July 7, 2007. Unfortunately, he didn't check with the development team first!

Medco公司的CEO告訴華爾街分析師，他們的系統會在2007年七月七日上線，但遺憾的是，他並沒問過開發團隊！



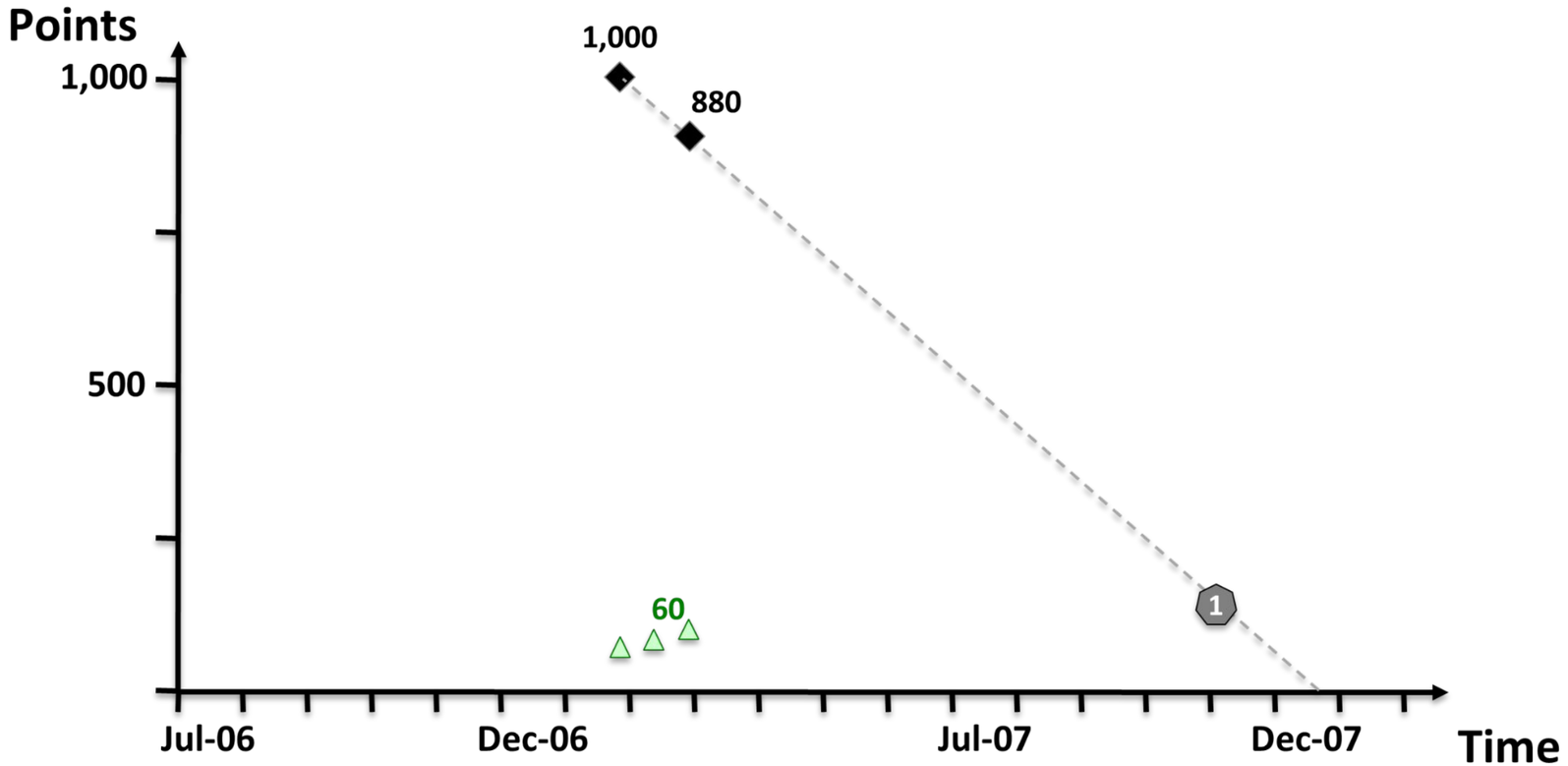


Chapter 6: Plan Reality Not Fantasy

Case Study: Deadline–Based Release Planning

On July 7, 2006, Medco CEO promised Wall Street analysts a completely new pharmacy fulfillment system to be implemented by July 7, 2007

- Scrum Inc team did a week-long release planning in January
  Scrum Inc 團隊在一月時用一週做了發布計畫



1 Teams generated Executive Action Team impediment list

1. .....
2. .....
3. .....
4. .....
5. .....
6. .....
7. .....
8. .....
9. .....
10. .....
11. .....

Question: If you are the person responsible for this project/product, what will you do first?

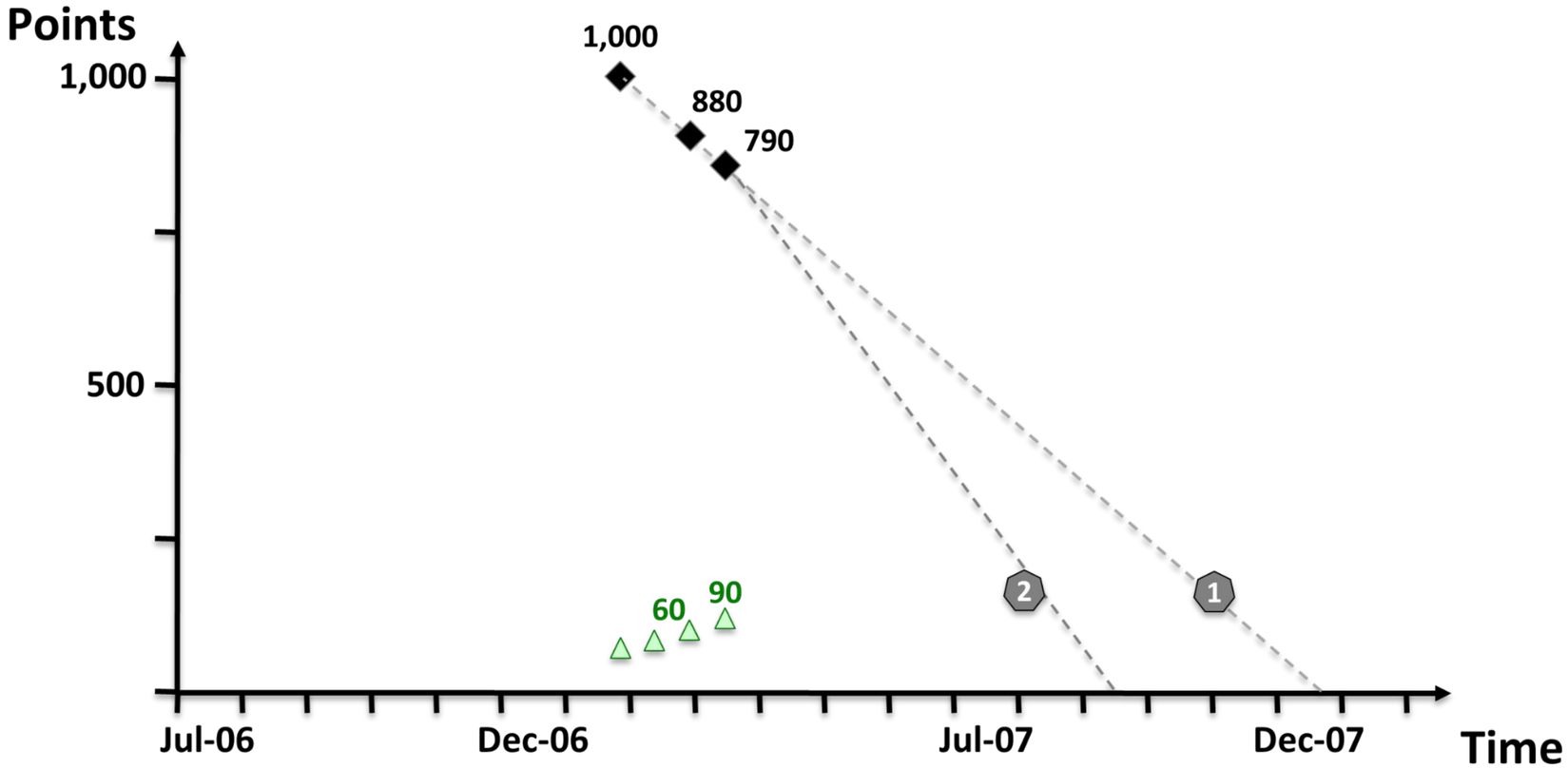
Promised Delivery Date

scruminc.

Case Study: Deadline–Based Release Planning

On July 7, 2006, Medco CEO promised Wall Street analysts a completely new pharmacy fulfillment system to be implemented by July 7, 2007

- Management team (EAT) removed impediments; **velocity increased to 90 points**
  管理團隊移除妨礙開發的障礙, 速率提升了90點



2 Executive Action Team removed all impediments in a week

1. .....
2. X
3. .....
4. .....
5. .....
6. .....
7. .....
8. .....
9. .....
10. .....
11. .....

Promised Delivery Date
CEO對外承諾的交付日期

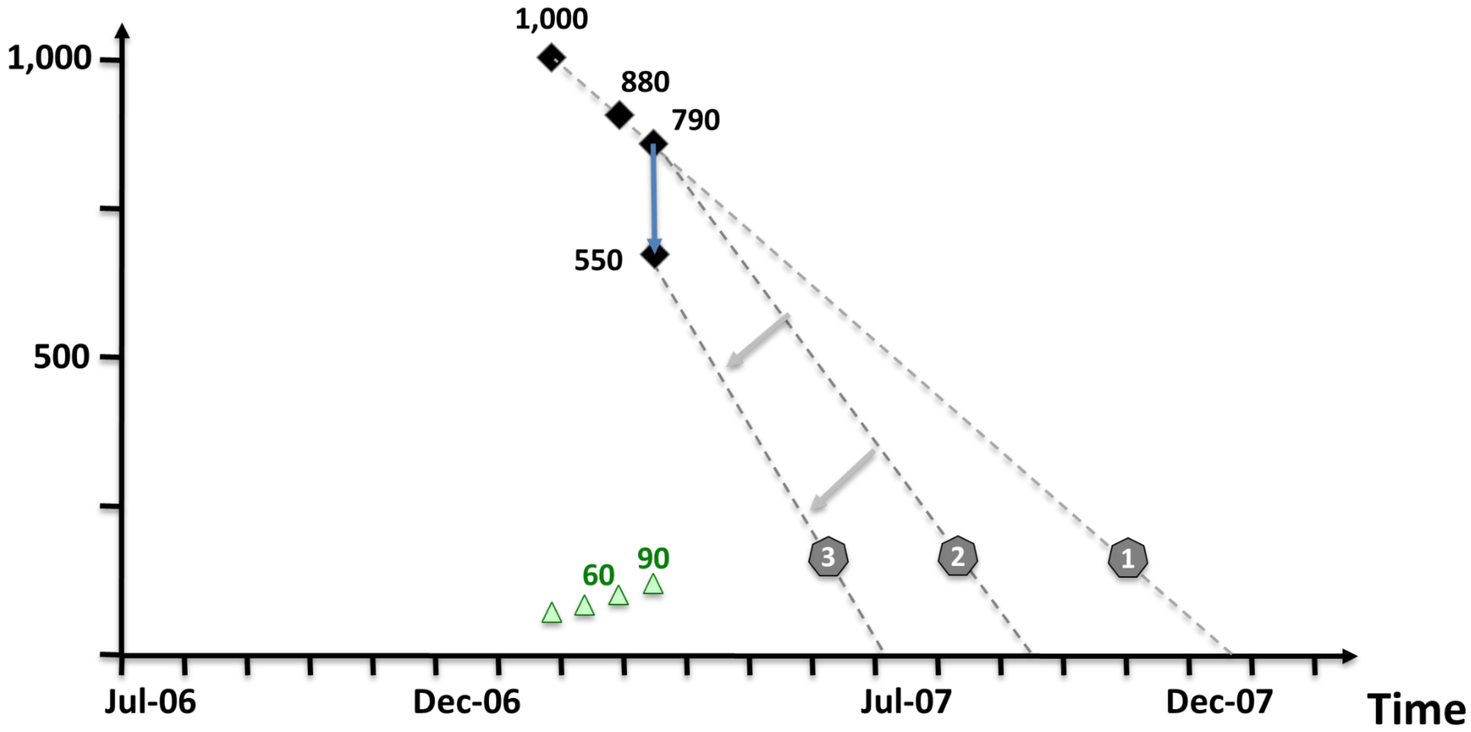
Question: You remove impediments, the teams are performing, but it's not going to happen, what should you do?

scruminc.

Case Study: Deadline–Based Release Planning

On July 7 2006, Medco CEO promised Wall Street analysts a completely new pharmacy fulfillment system to be implemented by July 7, 2007

- Team executed Scrum Emergency Procedure 團隊採用了Scrum緊急處理流程
- After go live, stock price doubled 上線後, 股價翻倍



Medco Stock

You have to Drop the low value features
There is no silver bullet
Do NOT ask people to work overtime (and it's too late to add more people)

Promised Delivery Date

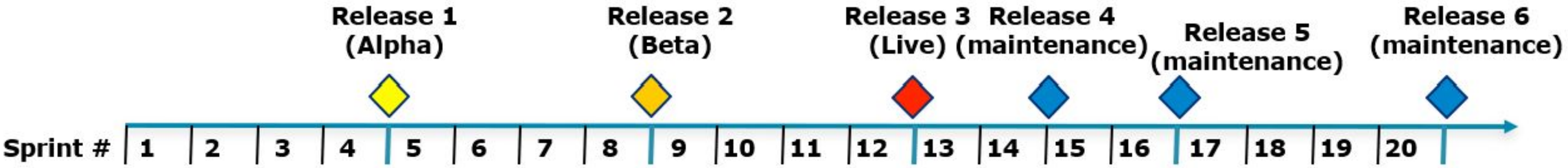
scruminc.

The PO Owns Release Plan, Dates, & ROI
PO負責發布計劃、日期和ROI

- **Get one Sprint READY backlog** 將一個Sprint的待辦事項準備好
  - Team can get started 團隊可以開始
- **Get two Sprints READY backlog** 將兩個Sprint的待辦事項準備好
  - Team can accelerate Sprint to Sprint 團隊可以加速
- **Build out Release Plan** 做好發布計劃
  - Customers can count on delivery dates 客戶可以指望交貨日期
  - Company can predict revenue 公司可以預測收入
- **Build one year roadmap** 做好一年的路線圖
  - Customers can be briefed on company strategy 可以向客戶簡報公司戰略
  - Company has a clear vision 公司有明確的願景

Release Cycles 發布週期

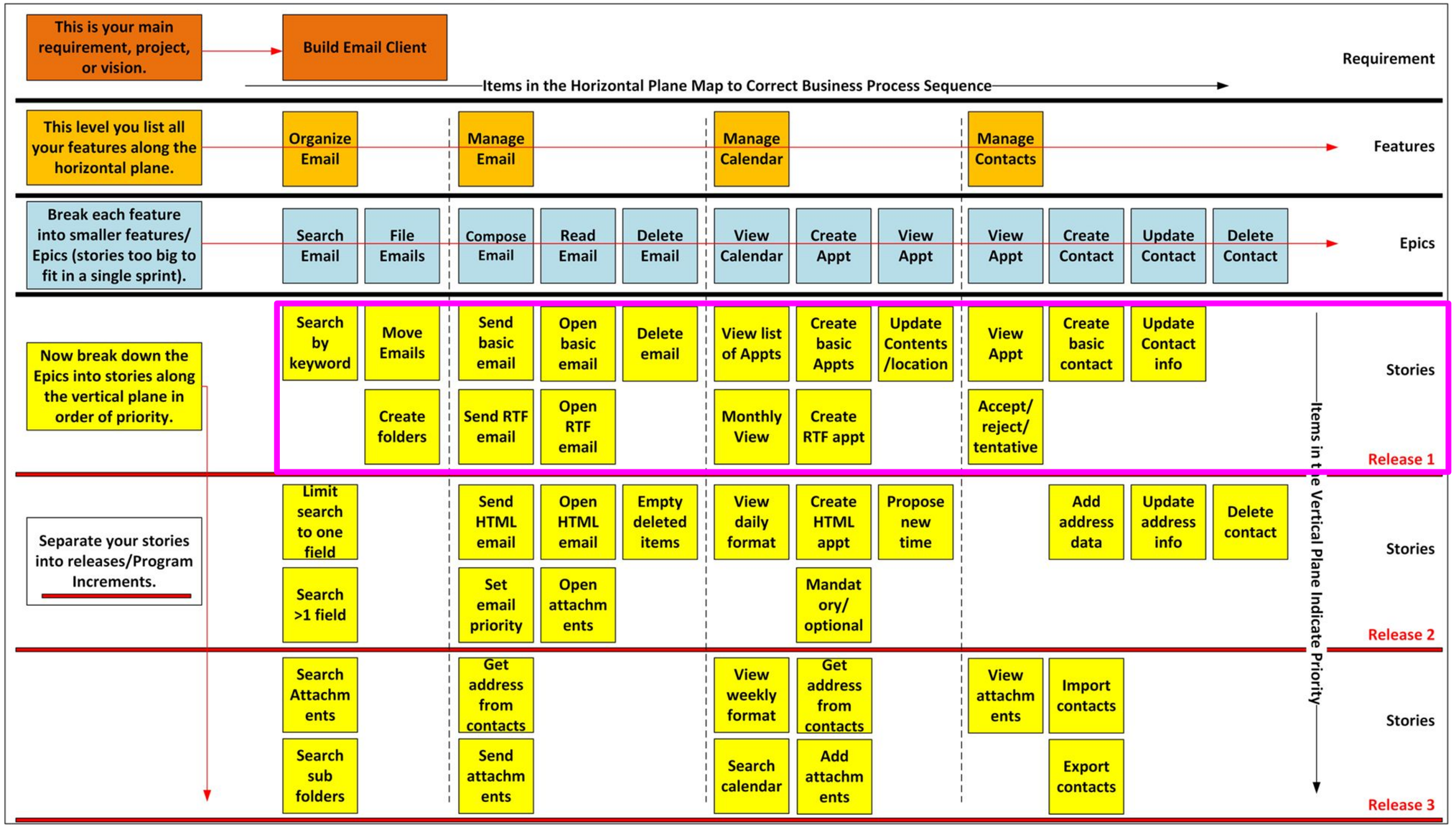
- Goal 目標: Every Sprint results in **potentially releasable product increment**
  每個Sprint都有可發布的產品增量
- Releasing every Sprint is NOT mandatory
  在每個Sprint發布並不是強制必做的
- The **Product Owner decides when to release**
  PO決定何時發布



Source: Henrik Kniberg

scruminc.

Story Mapping and Release Plan
故事地圖對照與發佈計劃



MVP
最簡可行產品
(Minimum Viable Product)

scruminc.

Release Burndown Chart 發布燃盡圖:
The Key to On Time Project Delivery 準時交付專案的關鍵

- Answers the key question: "Will we be done on time?"

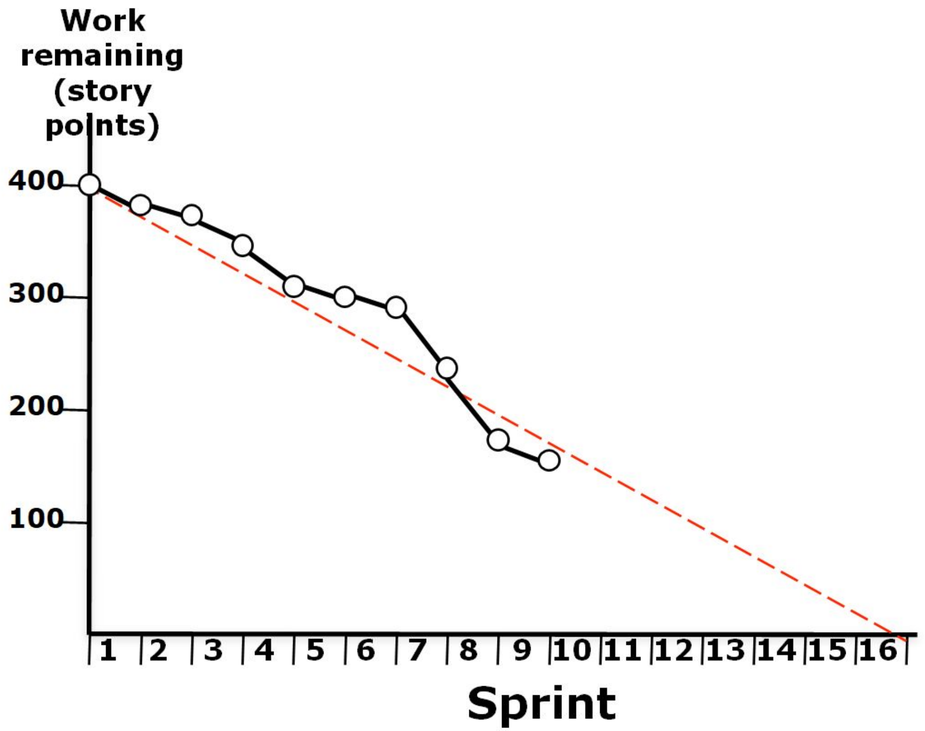
回答關鍵問題:"我們會按時完成嗎?"

- Useful for "what if?" analysis and managing tradeoffs of Scope, Velocity and Time

對假設分析有用和管理範圍, 速率和時間的權衡取捨

- Vital for identifying and addressing unreasonable expectations

對解決不合理的期望是重要的



Source: Henrik Kniberg

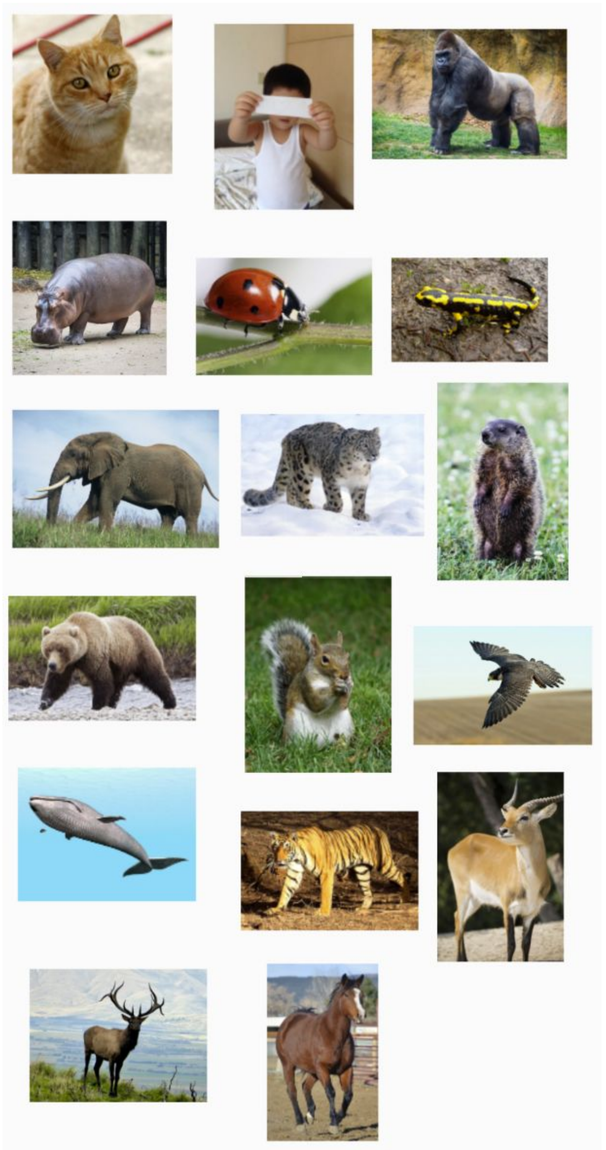
scruminc.

Large Scale Estimation 大規模估算

As a member of a Scrum Team, I need to know how to quickly estimate large amounts of the backlog, so that we can stop talking about work & start doing it.

身為Scrum團隊成員，我需要知道如何快速估算大量的待辦，
所以我們可以停止談並開始做

Affinity Estimation
近似估算



XS | S | M | L | XL

Rules of the Game 遊戲規則:

WITHOUT TALKING 不要說話

- Each team member takes a turn placing the name of an animal in a column of appropriate size **relative to the reference point**
  每位小組成員輪流移動動物照片放在**相對參考點**那一列
  Or you can move someone's card 或者您可以移動其他人所放的卡片

YOU MAY **NOT MOVE THE PIT BULL 不能移動鬥牛犬**

- After all animals have been placed, each person gets one turn to move **ONE** animal 放好所有動物後，每個人都可以再移動某一隻動物一次

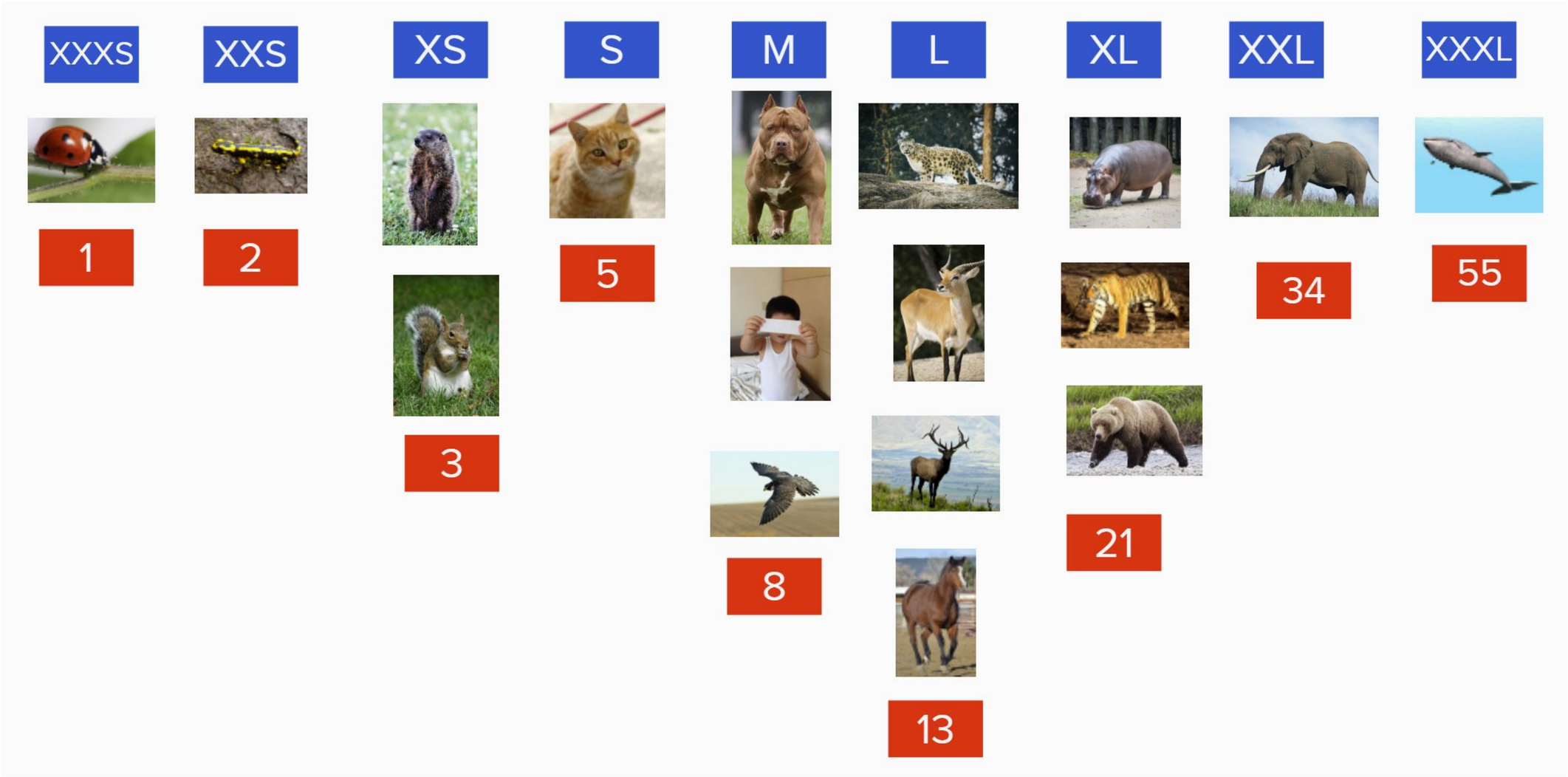
Affinity Estimation: Relative Size

近似估算: 相對大小

XS | S | M | L | XL

Affinity Estimation: Relative Size to Points

“相對大小” 對 ”點數”

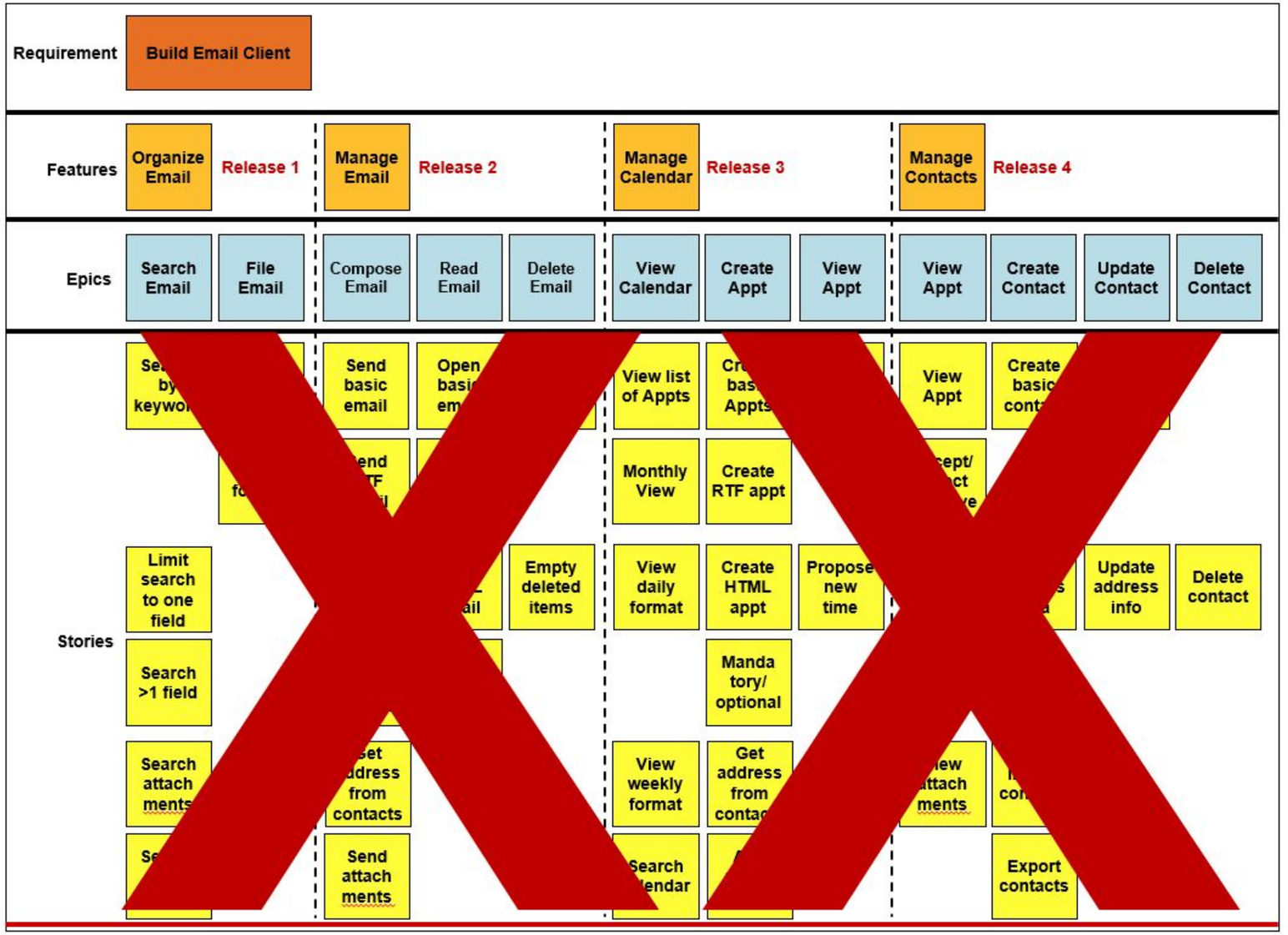


Large Scale Estimation 大規模估算

Requirement	Build “Client Email”			
Features	Organiza Email	Manage Email	Manage Calendar	Manage Contacts

Question: How will you estimate this big product?

Large Scale Estimation



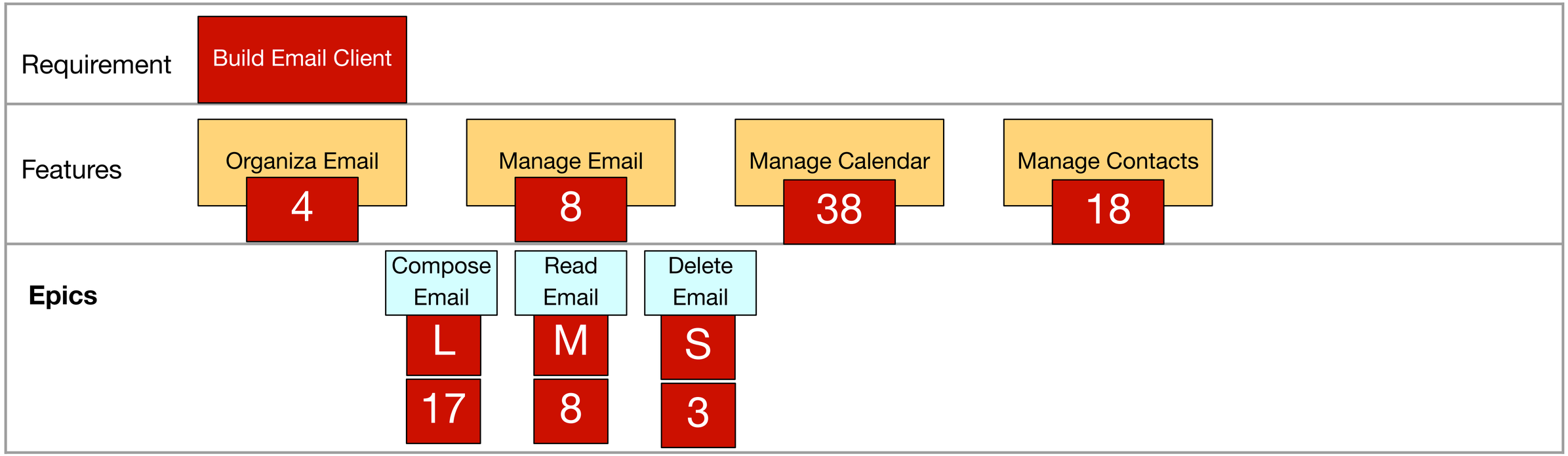
Large Scale Estimation (Feature Points)
功能點數

Requirement	Build Email Client			
Features	Organiza Email	Manage Email	Manage Calendar	Manage Contacts
	S	M	XL	L
	4	8	38	18

- Use **Affinity Estimation** to determine the relative size of the **top (Features) level items** to find a **middle ground. (M)**
  利用相似估算來決定上層項目的相對尺寸，就可以找到中間值(M)
- Without a Reference Library, pick a **Fibonacci number** (8) that makes sense for a medium-sized item (M)
  選一個費氏數列(8)能合理地用於一個中間尺寸的項目(M)
- Use **Estimation Cards** to determine the relative effort of the remaining top (Features) level items as a group
  利用估算卡片(估算撲克牌)來決定其餘上層(功能)整組項目之相對所需的費力/努力

scruminc.

Large Scale Estimation (Epic Points)
史詩點數

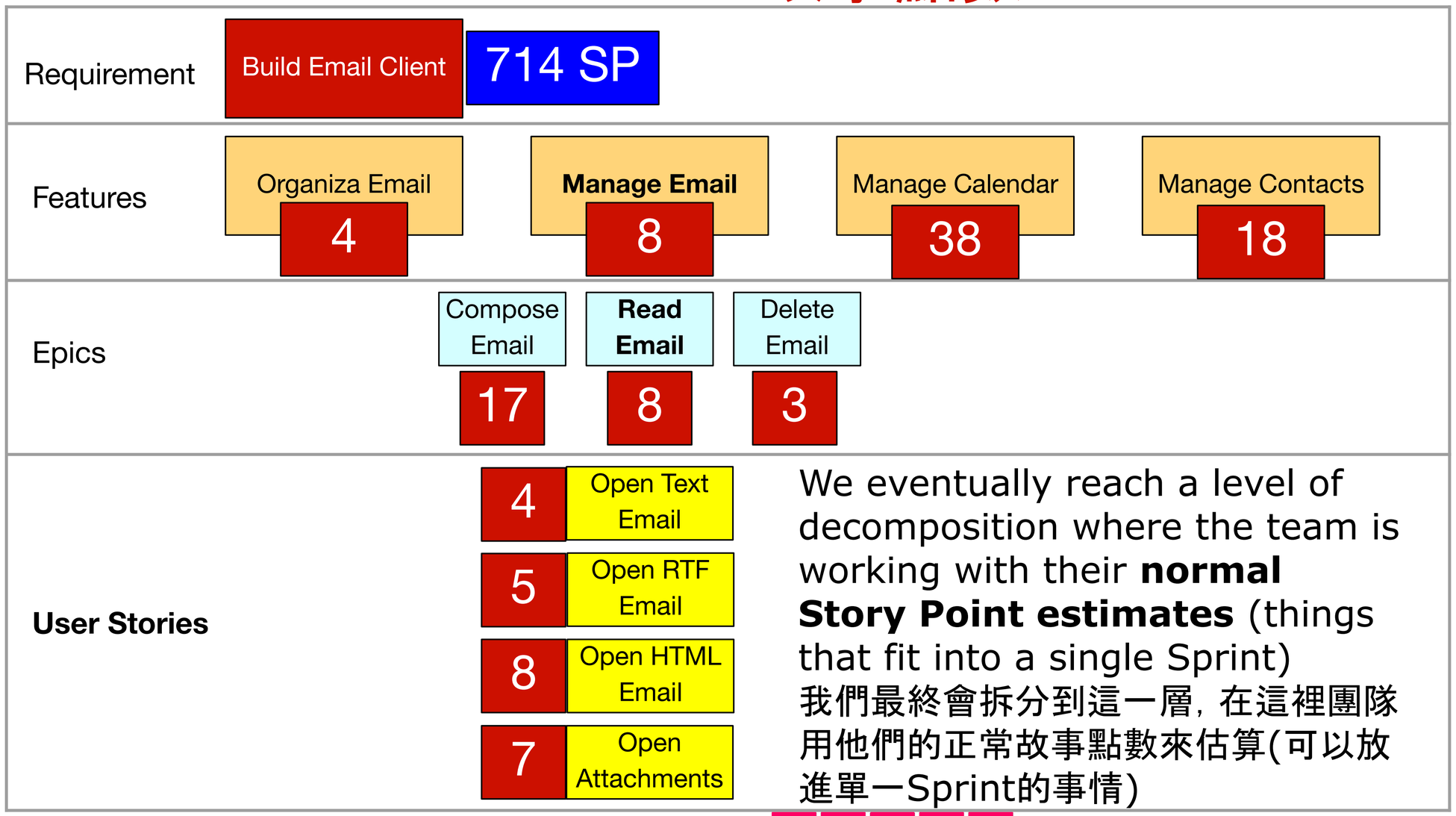


- Taking the medium-sized top (Features) level item (Manage Email), use **Affinity Estimation** again to determine the relative sizes of the **next level down (in this case, it's the Epics level)**
  拿中間尺寸的項目(管理email)，再利用相似估算來決定往下一層的相對尺寸
- Keep your reference size constant! (8)
  保持參考尺寸固定不變(8)
- Use **Estimation Cards again** to determine the relative effort of the remaining items as a group
  利用估算卡片來決定剩餘整組項目之相對所需的費力/努力

scruminc.

Large Scale Estimation (Story Points)

故事點數



We eventually reach a level of decomposition where the team is working with their **normal Story Point estimates** (things that fit into a single Sprint)
我們最終會拆分到這一層，在這裡團隊用他們的正常故事點數來估算(可以放進單一Sprint的事情)

Now we can get a rough estimate of the entire work

4+8+38+18=68 (FP)
so 68 FP = 238 EP
238 EP = 714 SP

17+8+3=28 (EP)
so 28 EP = 8 Feature Points (FP)
1 FP = 3.5 EP

4+5+8+7=24 (Story Points)
24 SP = 8 Epic Points (EP)
1 EP = 3 SP

scruminc.

Quality of Estimates 估算的品質

The **quality of estimate** depends upon a number of factors:
估算的品質取決於許多因素:

- **Clarity** of what is needed (and more importantly what is NOT needed) from the PO
  向PO澄清**什麼需要**(更重要的是**什麼不需要**)
- The team's **familiarity** with the work 團隊對工作的熟悉程度
- The **stability of the team**, as the length of time together increases the consistency of their **estimates** and work profile to predict their release **velocity**
  團隊的穩定度，因為一起工作的時間長度會增強他們估算的一致性和工作型態，以便預測他們的發布速率

Scrum Patterns 模式

As a member of a Scrum Team, I need to understand the key Scrum Patterns so that I can dramatically improve team performance after we've mastered the core Scrum

身為Scrum團隊成員，我需要了解主要的Scrum模式，
所以我可以在我們完全掌握Scrum核心後，引人注目地改進團隊的表現

Key Scrum Patterns
Required for Hyper-Productive Scrum
高產能Scrum團隊所需的關鍵模式

- **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** - How do you get started?
- **Ready Backlog** **準備就緒的產品待辦** - How do you avoid working on things not clear?
- **Yesterday's Weather** **昨天的天氣** - How do you pull backlog into a Sprint?
- **Swarming** **蜂擁行動** - How do you get work done quickly? (One Piece Continuous Flow)
- **Interrupt Pattern** **插件緩衝模式** - How to deal with interruptions during the Sprint?
- **Good Housekeeping** **良好的即時清除管理** (Daily Clean Code) - How to get defect-free product at Sprint end?
- **Happiness Metric** **幸福指標** - How do you ensure teams are happy and engaged?
- **Quantum Entanglement** **量子糾纏** - How do you build the team spirit in remote teams?
- **Emergency Procedure** **緊急程序** - How do you handle a Sprint towards failure?
- **Scrumming the Scrum** - **用Scrum來改善Scrum** How do you ensure you continuously improve?

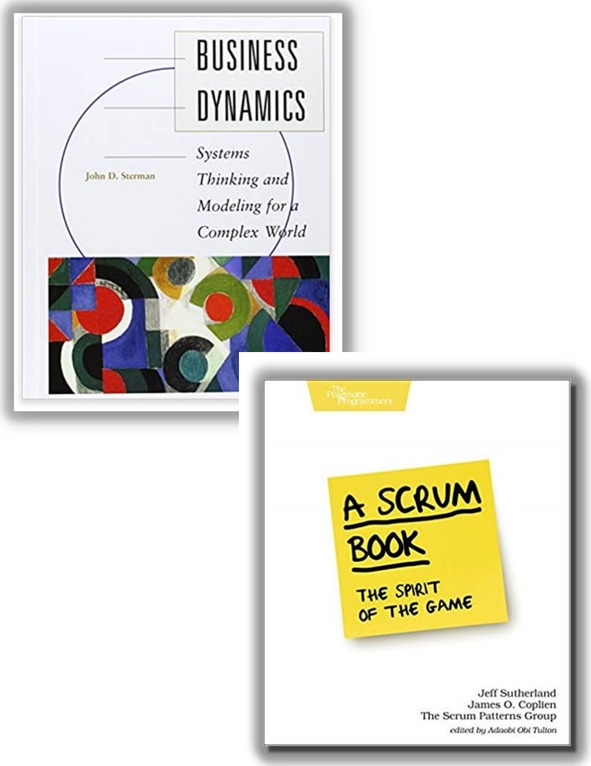
www.scrumplop.org

MURAL

scruminc.

A Pattern is a Solution to a Problem
模式是解決問題的方法

- Actions commonly generate resistance
  行動通常會產生阻力
- System dynamics must be investigated
- **The pattern is a solution** that generates side effects that overcome resistance and make the system work better
  模式是一種解決方案，能產生克服阻力的作用、使系統運作更好
- It takes a group of experts to grow a pattern
- **Patterns are supported by data from multiple organizations** that show it is a good general solution
  模式是得到許多組織數據的證實，它是好的通用解決方案
- Scrum Patterns are published in **"A Scrum Book: The Spirit of the Game"**



scruminc.

Hyper-Productive Team Patterns 高產能團隊模式

Teams That Finish Early Accelerate Faster 儘早完成交付的團隊加速更快

- **Small Teams 小團隊:** Adding more people to a team makes it slow
  加入更多成員會使速度變慢
- **Stable Teams 穩定的團隊:** Any change in team members causes delay
  團隊成員的任何變動都會導致延誤
- **Yesterday's Weather 昨天的天氣:** Taking just enough into a Sprint improves team performance
  加入剛好足夠的工作量以改善團隊績效

Yesterday's Weather

scruminc.

Exercise: Getting Work Done 完成工作

Requirement: Write the numbers “1” to “10”, the Letters “A” to “J”, and the abbreviations for the Months “Jan” to “Oct” 需求：寫下1~10、A~J、及月份的英文縮寫Jan~Oct

Track how many seconds it takes to complete all steps using two different work policies…

追蹤採用以下兩種政策各需要多少秒來寫下所有

政策A:
絕不能讓顧客空等

Policy A:
Never keep a customer waiting

Number	Letter	Month
1	A	Jan
2	B	Feb
3	C	Mar
4	D	Apr
5	E	May
6	F	Jun
7	G	Jul
8	H	Aug
9	I	Sep
10	J	Oct

Across the rows 1 by 1

橫向寫下每一列

Total time = ___

Policy B:
Limit Work in Process (WIP)

Number	Letter	Month
1	A	Jan
2	B	Feb
3	C	Mar
4	D	Apr
5	E	May
6	F	Jun
7	G	Jul
8	H	Aug
9	I	Sep
10	J	Oct

Down the columns 1 by 1

縱向寫下每一欄

Total time = ___

政策B:
限量進行中的工作

MURAL

scruminc.