第 1 章：案例研究：如何把你的 AI 專案徹底搞砸

CHAPTER 1Case Study: How to Completely F*ck Up Your AI Project

很多人說回頭看最清楚，但我個人很愛做事後檢討。把過去團隊怎麼搞砸看清楚，能教我未來怎麼避免重蹈覆轍。如同我在前言提到的，85% 的 AI
專案都會失敗（1）。先從示範事情可能怎麼出錯開始，是學這個主題很有力的框架。

A lot of folks say hindsight is 20/20, but personally, I love postmortems. Seeing clearly how my team screwed up in the past teaches me how to avoid creating the same messes in the future. As I mentioned earlier in the introduction, 85 percent of all AI projects fail (1). Starting by demonstrating how things might go wrong provides a powerful framework for learning the subject.

以我做過 35 個 AI UX 專案的經驗，最常見的失敗原因其實不是「AI」或「技術」本身——多半是 UX 設計、研究、測試與流程出了問題。很多時候，是多個問題一起作用，逐步削弱團隊的努力，形成所謂「千刀萬剮式的死亡」。在本書中，我會詳細拆解 AI
專案失敗的情況，並提供許多真實案例，幫你提早辨識紅旗、避開這些坑。

In my experience with 35 UX for AI projects, the most common causes of failure were not “AI” or “tech” failures per se—most were failures of UX design, research, testing, and process. Often, multiple issues work together to weaken the team’s effort and cause “the death of a thousand cuts.” Throughout this book, I will be covering AI project failures in detail and providing many real-life examples to help you recognize red flags early and avoid these pitfalls.

以下案例研究提供了一個很具代表性的真實例子，說明我們的團隊如何因為多個問題交織，沒能正確界定問題，最後在一團迷霧般的混亂中讓專案脫軌。

The following case study provides a salient real-life example of how our team failed to correctly frame the problem due to multiple issues that worked together to create a fog of confusion and caused the project to go off the rails.

一鍋沸騰的義大利麵

A Boiling Pot of Spaghetti

想像一個複雜的工業流程，包含酸性氣體去除單元（AGRU），它是液化天然氣（LNG）淨化流程中的關鍵環節。不必鑽太多細節，這個流程就像在爐子上煮一大鍋義大利麵：你把溫度調高，麵會更快煮熟，24 小時內就能煮出更多。但把火力加大，也會提高鍋子溢鍋的風險，讓黏黏的麵湯流得整個爐台都是（這往往得停機、啟動昂貴的工業清理程序，還把原本愉快的煮麵日變成面對暴怒主管的緊張對峙）。

Imagine a complex industrial process involving an acid gas removal unit (AGRU) that is a crucial part of the Liquid Natural Gas (LNG) purification process. Without going into too much detail, this process is akin to boiling a giant pot of spaghetti on a stove. If you increase the temperature, the spaghetti will be done faster, and you can cook more pasta in 24 hours. However, increasing the temperature of the stove burner also increases the risk of the pot boiling over, creating a sticky spaghetti mess all over the stove (which requires shutting down the cooking, deploying an expensive industrial cleaning process, and ruining a fun day of boiling pasta in favor of a tense encounter with an angry boss).

某家提供這種巨型 AGRU「工業煮麵鍋」的供應商（名字就不說了）想到一個絕妙主意：用 AI
來預測鍋子何時快要溢鍋。我們七人組——領著不錯薪水的資料科學 / 開發 / UX 專業團隊——花了 6 個月想把它做成，但很遺憾，整個專案徹底失敗。我們的 AI
專案之所以崩盤，正是因為接下來這五個常見且致命的失敗原則。

One industrial supplier of these giant AGRU “industrial pasta pots” (who shall remain nameless) thought of a brilliant solution: They could use AI to predict when the pot was about to boil over. My team of seven well-paid data science/dev/UX professionals spent 6 months trying to make it work, but sadly, the entire project was a complete and utter failure. Our AI project tanked due to the following five common critical failure principles.

失敗 #1：想用 AI 取代受訓專家

Fail #1: Try to Replace a Trained Expert with AI

我的團隊很快就發現，每一套價值數百萬美元的「工業煮麵鍋」，都有一位專責技師操作，受過訓練，能把沸騰維持在剛剛好的程度：產量高、又不會溢鍋。當技師看到液位快速上升，就會把火力調低，避免溢鍋。久而久之，這些技師都會成為自己那套設備的溢鍋避免專家。

It did not take my team long to figure out that every “industrial pasta pot” (worth millions of dollars) was operated by a dedicated expert technician trained to maintain the right level of boil to achieve a good yield without boiling over. If the technician saw the liquid level rise rapidly, they would lower the heat, avoiding overboiling. After a short time, these technicians became experts in avoiding overboiling in their specific pot installation.

我們推論，我們的 AI 方案可以取代這些技師，讓他們失業，順便替公司省錢。這個策略在理論上聽起來堅不可摧，但這套商業計畫就像拿濕紙巾去擋子彈。

Our team theorized that our AI solution would replace these technicians, rendering them unemployed and saving the company money in the process. While this strategy sounds bulletproof in theory, this business plan was akin to trying to block bullets with a wet tissue.

首先，即便他們是專家，這些技師的薪資其實沒有高到哪裡去；而我們一開始推出的 AI 方案，反而會讓客戶花得比現有技師更多。我們推銷的 AI 也不是針對客戶這一套特定設備受訓的。（事實上 AI 根本還沒受訓，因為我們拿不到做機器學習訓練所需的資料；第 3 點會再提。）所以，不可能出現 AI 表現跟技師一樣好、卻還更貴的情況。

To begin with, despite being experts, these technicians were not so highly paid, and our AI solution out of the gate would cost the customer much more than their existing technician. The AI my team was selling them on was not trained on their specific pot installation. (AI was actually not trained at all since my team could not get the data for ML training; see point 3 later). So, there was no possibility of AI performing as well as the technician while actually costing more.

注意

NOTE

只要你的 AI 解決方案想取代既有、已上線的專家操作員，就要特別小心！這是非常大的紅旗，專案失敗的機率會大幅上升。若你的 AI 方案比既有專家還貴，不要只是「算了走人」——直接跑。

Any time your AI solution tries to replace an existing installed expert operator, take care! This is a huge red flag, and the likelihood of your project failing goes way up. If your AI solution costs more than the installed expert, don’t just walk away. Run.

如何為你的 AI 專案挑對使用案例的詳細指南，會在第 2 章〈挑對使用案例的重要性〉中說明。

The detailed guide on how to pick the right use case for your AI project is covered in Chapter 2, “The Importance of Picking the Right Use Case.”

失敗 #2：忘了算成本 vs. 效益

Fail #2: Forget About Cost vs. Benefit

在開始打造 AI 之前，先花時間搞懂你的使用案例的成本 / 效益分析。每一次 AI 行動都是一個「帶著某個成功或失敗機率的預測」，而每個預測結果都對應到特定的成本與效益。我們的專案團隊在開發 AI 方案前，沒有把這個專案的成本 / 效益量化。

Before engaging in an AI-building exercise, take the time to understand the cost/benefit analysis of your use case. Every AI action is a prediction with a certain probability of success or failure, and the outcome of every prediction has a specific cost and benefit. Our project team failed to quantify the cost/benefit of this project before developing the AI solution.

別犯同樣的錯。

Don’t make the same mistake.

雖然避免溢鍋相對容易（只要把火力調低），但一旦發生溢鍋，成本衝擊非常高。光一次溢鍋的成本，就可能是專家操作員年薪的好幾倍。因此，要合理化這套安裝的價格，AI 方案就必須在避免溢鍋上「準到離譜」：因為只要一次假陰性（沒有預警到溢鍋），就會把你一整年成功避免溢鍋所帶來的全部利潤一筆抹平，成本 / 收益比甚至可能是 1000:1 的劣勢（例如，1,000 次正確的真陰性，抵不過一次假陰性）。

While avoiding overboiling was relatively easy (just lower the heat), the cost impact of an overboiling event was very high. The cost of just a single overboiling of the pot was several times higher than the yearly salary of the expert operator. Thus, to justify the price of the installation, the AI solution needed to be ridiculously accurate at avoiding overboiling because a single false negative (failing to check the overboiling) would wipe out all of the profits from a full year of preventing overboiling or a factor of 1000:1 against (e.g., 1,000 correct true negative guesses against a single false negative).

這種嚴重偏斜的成本 / 效益結構，使得要說服客戶用昂貴、未證實、未受訓的 AI 方案，去取代他們已證明有效、已部署且受過訓練的方案（低成本、全職的專家鍋爐操作員），變得非常困難。

Skewed cost/benefit impact made it very hard to convince the customers that they should replace their proven, installed, and trained solution (a low-cost, full-time expert pot operator) with an expensive, unproven, untrained AI solution.

更讓導入 AI 變得不可能的是：我們公司拒絕承擔因 AI 判斷錯誤而造成的溢鍋成本，整件事從一開始就注定做不成。

As an additional disincentive to adopting AI in this case, our company refused to cover the cost of overboiling caused by a faulty AI guess, making the whole thing a complete non-starter.

注意

NOTE

如果 AI 猜錯的潛在成本遠遠大於猜對的效益，就別做了。如果一次糟糕的 AI 猜測代價是災難性的——快跑。

If the potential cost of a wrong AI guess far exceeds the benefit of a correct AI guess, walk away. If the cost of a bad AI guess is catastrophic, run.

如何用價值矩陣做你自己的成本 / 效益分析的詳細步驟，會在第 5 章〈價值矩陣——AI 準確率是狗屁，UX 必須怎麼做〉中說明。

The detailed walk-through on conducting your own cost/benefit analysis with a value matrix is covered in Chapter 5, “Value Matrix—AI Accuracy is Bullshit. Here’s What UX Must Do About It.”

失敗 #3：沒有機器學習訓練資料？沒問題！

Fail #3: No ML Training Data? No Problem!

我們公司雖然製造這些鍋子，但並不自己使用；只有客戶在用。這讓資料收集從一開始就很棘手。每個鍋子的成本太高，導致全球也只裝了幾千套——不足以自動蒐集到可泛化的機器學習（ML）資料。

While our company made pots, it did not use them; only our customers did. This made collecting data challenging from the start. The high cost of each pot meant that only a few thousand pots were installed worldwide—not enough to automatically collect generalized machine learning (ML) data.

更糟的是，每一套安裝都略有不同：管線不同、熱源不同，大氣溫度、壓力、濕度、流量、風扇等都有些微差異，使得每套都是客製化的。（就像我在我家的爐子上煮一鍋麵，對你家爐子上煮麵的條件毫無參考價值——即使我們用的是同一款鍋子！）安裝 A 的 AI 模型無法用在安裝 B。也就是說，每一套鍋子都需要自己的客製 AI 系統。

What made things even worse was that every pot installation was a little different: different pipes, different heat sources, slight variations in atmospheric temperature, pressure, humidity, rate of flow, fans, and the like made each installation bespoke. (In the same way that me boiling a pot of pasta on my stove tells you nothing about the conditions of boiling pasta on your stove, even if we both use the same pot!) The AI model from installation A could not be used in installation B. This meant that every pot required its own custom AI system.

注意

NOTE

如果你沒有資料可訓練 AI/ML，或沒有一個容易、便宜的方式取得資料，就別做了。如果你的解決方案要求每套安裝都要一個客製 AI 模型——快跑。

If you do not have the data to train your AI/ML or have no easy, cheap way to obtain the data, walk away. If your solution requires a custom AI model for each installation, run.

想深入了解如何辨識 ML 訓練資料中的偏誤，以及應對的技巧，請看本書第 4 部分〈偏誤與倫理〉。

For a detailed discussion on how to spot the bias in your ML training data and techniques for dealing with it, look to Part 4, “Bias and Ethics,” of this book.

失敗 #4：AI 模型回答什麼問題都無所謂

Fail #4: It Makes No Difference What Question Your AI Model Is Answering

即使光是缺資料就該讓專案直接終止，我們團隊用 AI 在建模的那個問題，才真正為專案蓋上棺材板。

While the lack of data alone should have killed the project, the question my team was modeling with AI sealed the project's demise.

人類操作員的任務，是回答這個問題：我最多可以把溫度調到多高，才不會讓溢鍋風險高到不可接受？

The human operator was tasked with answering the question: How high can I make my temperature before the risk of boiling over is too great?

相對地，我們團隊打造的 AI 模型，想回答的是另一個問題：在這個設定下，已知溫度與壓力的量測值，我距離下一次溢鍋事件還有多久？

In contrast, the AI model my team was building was trying to answer a different question: Given the measurement of temperature and pressure at this setting, how long do I have until the next boil-over event?

現在你就看出問題了。操作員的問題是為了提高客戶的利潤——因為你記得，火越大，一天結束時煮出的麵就越多。

Now you can see the problem. The operator's question aimed to increase the customer's profit because, as you recall, more heat meant more cooked pasta at the end of the day.

但 AI 想回答的問題，雖然跟營運有關，卻不一定直接對準「增加獲利」。只是因為這個問題比較方便建模，我的雇主就覺得『這樣也行』。

In contrast, AI was trying to answer a question that was related to operations but not necessarily directly aimed at increasing profits. However, it was a convenient question for our model to answer, so my employer decided it was good enough.

結果並不行。

It wasn't.

我們團隊就像那個著名（或說臭名昭著）的笑話裡的主角：醉漢找鑰匙。

My team was akin to the protagonist in that (in)famous joke about a drunken man looking for his keys:

深夜，一名醉漢在路燈下趴在地上，用手和膝蓋爬來爬去，拼命找東西。路人停下來想幫忙。

In the middle of the night, a drunken man is crawling on his hands and knees underneath a streetlight, intently looking for something. A passerby stops to help.

注意

NOTE

如果你的 AI 模型試圖回答的不是直接與最大化獲利相關的問題，而只是在回答一個資料科學問題——別做了。如果你的團隊堅持只在路燈下找，理由只是那裡『看得清楚自己在做什麼』——快跑。

If your AI model is trying to answer a question not directly related to maximizing profits but instead is answering a data science question, walk away. If your team insists on looking under a streetlight only because it's the only place they can see what they are doing, run.

想看更完整的說明：如何用數位分身（digital twin）來建模輸入與輸出，讓 AI 回答客戶真正關心的問題，請參考第 4 章〈數位分身——系統實體元件的數位呈現〉。

For a detailed write-up on modeling inputs and outputs with a digital twin so that your AI can answer the question your customer actually cares about, see Chapter 4, "Digital Twin—Digital Representation of the Physical Components of Your System."

失敗 #5：不用做使用者研究——我們有 SME！

Fail #5: Don’t Worry About User Research—You Have an SME!

我們公司鍋子安裝的 1,000 多座工廠大多偏遠，不容易進行使用者研究。於是我們做了各式各樣的假設，而且多半是錯的；如果我們能親眼去看一眼，很多問題一小時內就能釐清。

Each of the 1,000+ plants where my company’s pots were installed was remote and not readily accessible for user research. As a result, our team made all kinds of assumptions, most of them wrong, that could have been cleared up within an hour of seeing the situation for ourselves.

回想一下，我們的 AI 想回答的是：已知溫度與壓力的量測值，我距離下一次溢鍋事件還有多久？

Recall that our AI was trying to answer the question: Given the measurement of temperature and pressure, how long do I have until the next boil-over event?

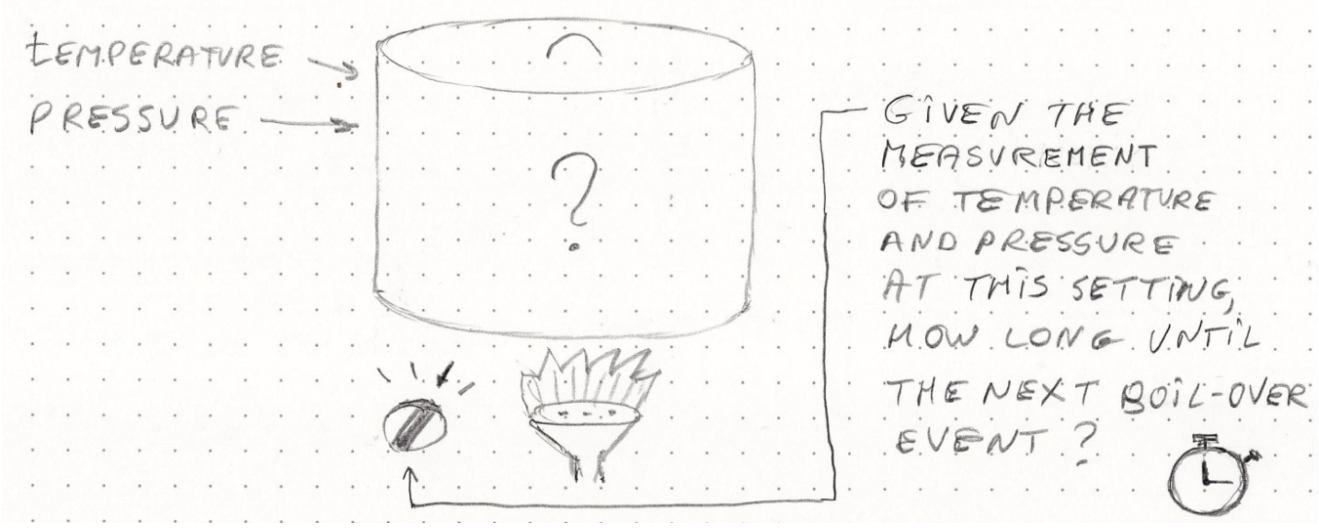
我們的領域專家（SME，subject-matter expert）告訴我們，AI 能拿來建模的感測器讀值只有兩個：

Our subject-matter expert (SME) told us that the only two sensor readings available to AI for modeling were

- 溫度
- Temperature
- 壓力
- Pressure

因此，我們的 AI 驅動系統的數位分身模型看起來像圖 1.1。（我會在第 4 章詳細介紹數位分身。）

Thus, our AI-driven system’s digital twin model looked like Figure 1.1. (I will cover digital twins in detail in Chapter 4.)



凡是煮過加蓋義大利麵的人都知道，溢鍋不是慢慢發生的——它很快、很爆炸、很髒亂。避免溢鍋的最佳方式，是看液面狀態，而不是看壓力或溫度。

Anyone who has ever boiled a covered pot of pasta knows that overboiling is not gradual—it is fast, explosive, and messy. The best way to avoid overboiling the pasta is to look at the surface of the liquid, not at the pressure or temperature.

在我們苦苦鑽研又一直失敗了好幾個月之後，我和團隊才發現：人類操作員還有一個額外的『感測器』，確保他們能成功——他們可以透過一個小玻璃窗看向鍋內沸騰的液面，用肉眼判斷沸騰狀況。就像你的煮麵鍋有個透明玻璃蓋一樣——對避免溢鍋事故非常有幫助！

After many months of toiling at the problem and failing, my team and I discovered that the human operator had an additional sensor that ensured their success: They could look through a small glass window onto the boiling surface of the pot and visually ascertain how the boiling was performing. It was like having a transparent glass lid on your pasta pot—a great help in avoiding overboiling accidents!

因此，真實流程的數位分身其實看起來像圖 1.2。

Thus, the digital twin of a real-life process looked like Figure 1.2.

果不其然，公司 SME 知道這個「沸騰液面觀察窗」。但他認為這不是必要資訊，因為液面的視覺資訊不容易用感測器量化。另兩個感測器（溫度與壓力）則是每套鍋子上本來就有、方便取得的數字，你很容易把讀值餵進 AI 模型。液面的視覺訊號沒有被儀器化；它『很麻煩』，而且需要受訓的人類才能靠觀察液面來判斷鍋子是否快要溢鍋。

Sure enough, the company's SME knew about this "boiling surface window." Still, he thought it was not essential to tell us about it because the visual of the boiling surface could not be easily instrumented with a sensor. The other two sensors (temperature and pressure) were convenient numbers already instrumented on every pot, and you could easily feed the readings into the AI model. The visual of the boiling surface was not instrumented. It was "messy," and it took a trained human to be good at judging whether the pot was about to overboil by looking at the surface of the liquid.

因此，我們的 AI 模型從一開始就完全沒有機會解決這個問題。

As a result, our AI model had no chance in hell to solve this problem.

其實只要做一次田野研究，我們就會知道：如果沒有在「沸騰液面觀察窗」上加裝感測器，我們不可能成功。可惜領導層認為這樣的研究不必要，而且超出預算。

Even a single field research session would have told us that we had no chance of success without instrumenting a sensor on the "boiling surface window." Unfortunately, the leadership deemed such a session unnecessary and over budget.

注意

NOTE

如果你沒有一個運作良好的研究計畫，能幫你直接連結你的客戶——別做了。如果你連一次面對面、到現場的訪談都做不到——快跑。

If you do not have a well-run research program that will help you connect directly with your customers, walk away. If you cannot conduct even a single in-person, on-site interview with your target customers, run.

你可以在第 3 部分〈AI 專案的研究〉讀到更多關於做 AI 驅動專案研究的「新常態」。

You can read more about the “new normal” of conducting research for AI-driven projects in Part 3, “Research for AI Projects.”

最後的想法

Final Thoughts

總結來說，這個 AI 專案的失敗歸結為以下幾點：

To summarize, the failure of this AI project came down to the following:

- 試圖用 AI 取代受訓專家
- Trying to replace a trained expert with an AI
- 忘了分析成本 vs. 效益
- Forgetting to analyze costs vs. benefits
- 拿不到 ML 訓練資料
- Not getting the ML training data
- 沒有仔細注意你的 AI 模型到底在回答什麼問題
- Not paying careful attention to the question your AI model was answering
- 因為有 SME，就完全沒做使用者研究
- Not doing any user research because we had an SME

回頭看最清楚——回顧能讓你清楚看見團隊過去把事情搞砸的各種方式。這很不舒服，但如果我們想進步、避免未來重複犯錯，這樣的學習是必要的。我希望你讀完我們的錯誤後，能少犯一些自己的錯。我建議你把本案例的五個原則寫下來，貼在螢幕上方，讓它們在你做自己的 AI 驅動專案時一直提醒著你。

Hindsight is 20/20—looking back allows you to see clearly all the ways your team screwed up. While it’s uncomfortable, this learning is essential if we aim to improve and avoid the same mistakes in the future. I hope that by reading about our mistakes, you can avoid making some mistakes of your own. I suggest you write down the five principles in this study and tape them above your monitor so they remain top of mind as you work on your own AI-driven projects.

本書接下來的章節會更深入探討這些挑戰，提供更多真實案例，並解釋如何避開專案中的關鍵陷阱。下一章我會再分享另一個真實故事：某個專案一開始也想用 AI
取代受訓專家。幸運的是，我和團隊透過策略性的使用者研究重新界定問題，成功把專案扭轉回來。

Subsequent chapters throughout the book will explore these challenges further, provide additional real-life examples, and explain how to avoid critical pitfalls in your project. In the next chapter, I will share another real-life story about a project that initially tried to replace a trained expert with AI. Fortunately, my team and I

successfully turned the project around by reframing the problem through strategic user research.

參考資料

Reference

1. 1. Vartak, M.（2023）。透過聚焦機器學習的營運面，從 AI 中取得更高層次的價值。Forbes Technology Council。www.forbes.com/sites/forbestechcouncil/2023/01/17/achieving-next-level-value-from-ai-by-focusing-on-the-operational-side-of-machine-learning

1. 1. Vartak, M. (2023). Achieving next-level value from AI by focusing on the operational side of machine learning. Forbes Technology Council. www.forbes.com/sites/forbestechcouncil/2023/01/17/achieving-next-level-value-from-ai-by-focusing-on-the-operational-side-of-machine-learning