CHAPTER 1

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

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.

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.

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.

A Boiling Pot of Spaghetti

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).

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.

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

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.

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.

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*.

> **NOTE**
>
> 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.

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."

Fail #2: Forget About Cost vs. Benefit

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.

Don't make the same mistake.

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).

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.

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.

NOTE

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.

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."

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

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.

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.

NOTE

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.

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.

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

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

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?

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?

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.

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.

It wasn't.

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

> 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.
>
> Passerby "What did you lose?"
> Drunk "My keys."
> Passerby "Where did you lose them?"
> Drunk "Over there in the bushes."
> Passerby "Then why are you looking for them here?"
> Drunk "Because here, under the streetlight, I can see what I'm doing!"

NOTE

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.

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."

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

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.

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?

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

- Temperature
- Pressure

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

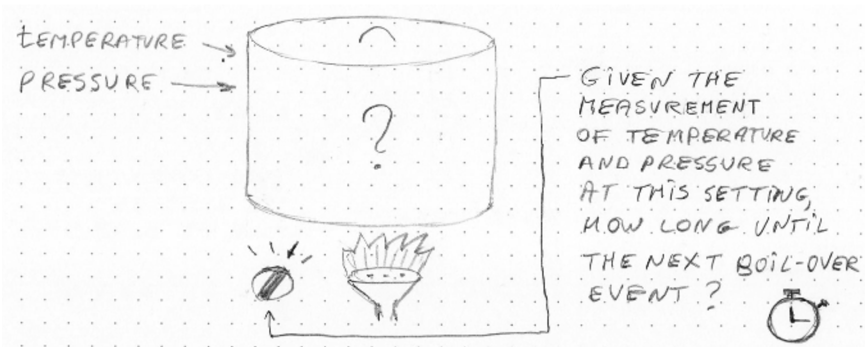

Figure 1.1 Digital twin of an AI model of the process

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*.

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!

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

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.

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

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.

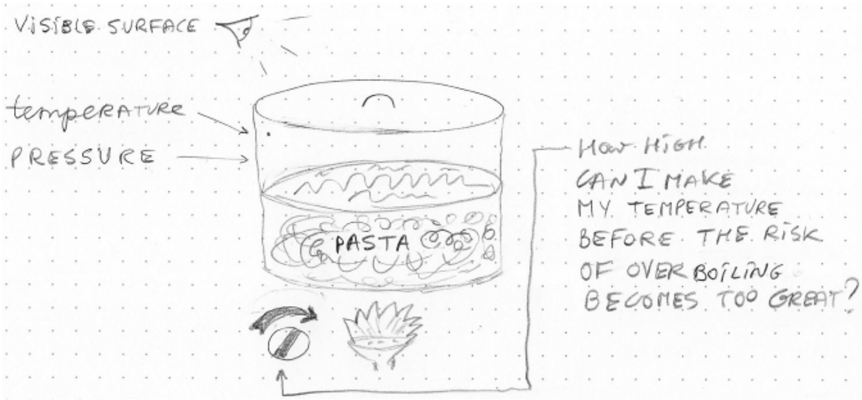

Figure 1.2 Digital twin of a real-life process

NOTE

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.

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

Final Thoughts

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

- Trying to replace a trained expert with an AI
- Forgetting to analyze costs vs. benefits
- Not getting the ML training data
- Not paying careful attention to the question your AI model was answering
- Not doing any user research because we had an SME

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.

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.

Reference

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