MODULE 3 · SESSION 1

Engineer Reliable AI Operations

How failure tolerance, service objectives, freshness and fallback behaviour combine into a reliability operating model.

Theory session

Beginner

Running case: VerdantCart Markets

ORIENTATION

Where this session sits

Module 3 moves from control to the strategic operating choices

MODULE 1
AI System Foundations

MODULE 2
Controlled AI Operations

MODULE 3
Strategic AI Operating Choices

SESSIONS IN THIS MODULE

3.1 **Engineer Reliable AI Operations** — YOU ARE HERE
Failure tolerance, service objectives and customer impact

3.2 Control Inference Spend at Scale
Reducing cost without giving up quality or coverage

3.3 Select an AI Vendor Portfolio
Capability, switching exposure and strategic control

SESSION OUTCOMES

What this session explains

Making an AI service dependable enough to build a business on

01

Reliability is defined, not felt

A service objective states what “working” means as a number, so everyone argues about the same thing.

02

Freshness is a reliability dimension

A correct answer about a world that has moved on is a wrong answer.

03

Degrade on purpose

Every AI feature needs a defined behaviour for when it cannot answer well.

04

Reliability is layered

Instacart changed the refresh rate, the interface, the architecture and the solver.

DEFINITION

Service level indicators, objectives and error budgets

Three words that make reliability a decision instead of an argument

WORKING DEFINITION

An indicator is the thing you measure. An objective is the level you promise. The gap between the objective and one hundred per cent is the error budget — the amount of failure you have deliberately decided to accept.

1

Indicator

"The share of availability scores computed within the last hour." Measurable from real traffic.

2

Objective

"99% of scores are under one hour old." A promise, chosen with the business, not by engineers alone.

3

Error budget

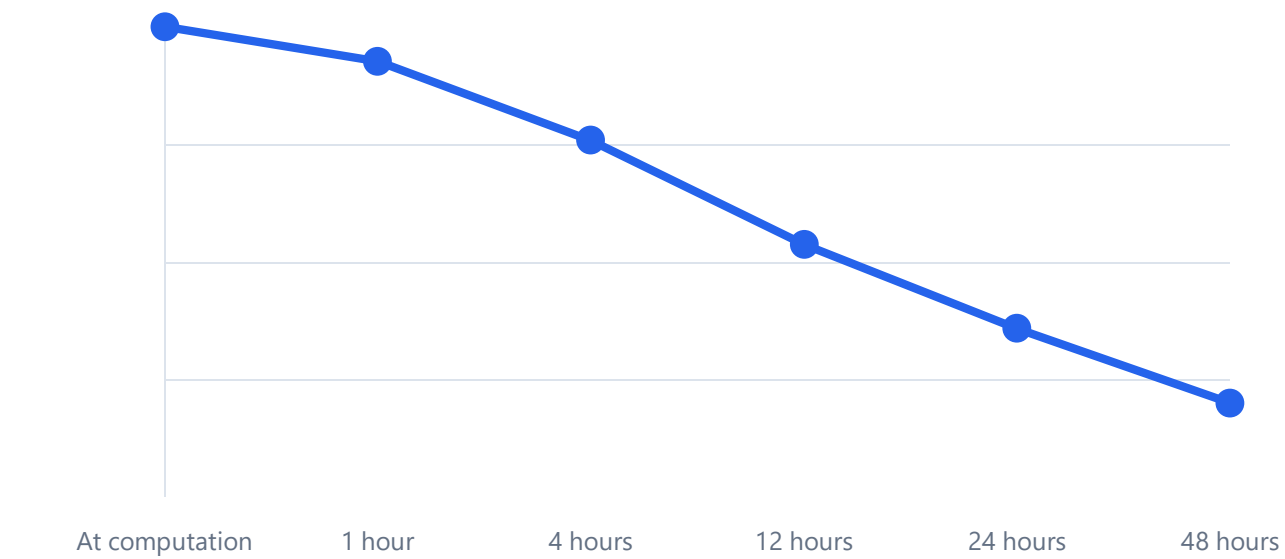
The 1% left over. Spend it on change, or spend it on outages — you cannot spend it twice.

Beyer, B., Jones, C., Petoff, J. & Murphy, N. (2016). Site Reliability Engineering. Sebastopol: O'Reilly.

CONCEPT 1 · FRESHNESS

How quickly an answer stops being true

The dimension most AI teams never measure



The steeper this curve, the more refresh rate matters and the less model accuracy alone can help.

WHAT INSTACART DID FIRST

Halve the age

Availability model runs doubled to every 60 minutes, and the historical window narrowed to 3–7 days.

WHY THAT WAS NOT ENOUGH

Scheduling has a floor

Running more often costs more and still leaves an average age. Event-based processing changed the shape, not the rate.

THE GENERAL RULE

Measure age, not just accuracy

Report the distribution of answer age next to accuracy. They are separate failures with separate fixes.

CONCEPT 2 · LEVERS

Four places reliability is actually won

Instacart used all four; most teams try only the second

THE DATA PATH	How fresh the inputs are, and how quickly a change in the world reaches the system.
THE MODEL	Accuracy, calibration, and knowing when it does not know.
THE INTERFACE	What the user is shown when confidence is low — out-of-stock badges and filters.
THE INFRASTRUCTURE	Timeouts, retries, capacity and solver reliability under load.

THE CHEAPEST LEVER

The interface

Telling a shopper an item may be unavailable costs nothing and removes most of the harm from a wrong score.

THE ONE PEOPLE FORGET

Infrastructure

Instacart adopted a commercial solver specifically to eliminate timeouts and alerts — a reliability fix with no model in it.

CONCEPT 3 · DEGRADATION

What the system does when it cannot answer well

Defined in advance, in descending order of quality

FULL SERVICE

Fresh inputs, confident model, complete answer.

REDUCED CONFIDENCE, FLAGGED

Answer given with an explicit caveat in the interface — the out-of-stock badge.

FALLBACK

A cached answer, a simpler rule, or the last known good value.

REFUSE SAFELY

Say the system cannot answer, and route to a human. Always better than a confident guess.

THE DESIGN QUESTION

What is the honest bottom rung?

Every feature has one. If you have not chosen it, the system will choose something worse under load.

THE CONNECTION BACK

This needs confidence

Graceful degradation depends on the model knowing when it is unsure — Session 1.1's interface rule.

TEST IT DELIBERATELY

Exercise the fallback

An untested fallback path is a second outage waiting for the first one.

CONCEPT 4 · OBJECTIVES

Setting a service objective you can actually defend

Four questions, answered with the business and not alone

Question	Why it matters	VerdantCart example
What does the user notice?	Measure the thing that affects them, not the thing that is easy to log.	Items shown that turn out to be unavailable at pick time.
What level is good enough?	Higher objectives cost disproportionately more.	99% within the hour, not 99.99% — the extra nines buy nothing here.
What happens when it is missed?	An objective with no consequence is a wish.	Breach freezes feature launches until the budget recovers.
Who owns it?	One named team, as in Session 2.1.	The availability platform team, reported in the weekly loop.

THE POINT OF AN ERROR BUDGET

Aiming for perfect reliability is the most common expensive mistake. The error budget exists so that the remaining failure is a deliberate choice rather than a surprise.

CONCEPT 5 · TWO POSTURES

Prevent failure, or recover from it

You need both, and organisations usually over-invest in one

PREVENTION

Reduce how often it breaks

EXAMPLES

Better models, more testing, stronger validation, redundancy.

STRENGTH

Fewer incidents reach the customer at all.

ITS LIMIT

Costs rise steeply, and novel failures ignore the ones you predicted.

RECOVERY

Reduce how long it hurts

EXAMPLES

Fast detection, fallbacks, rollback, and capacity to absorb a surge.

STRENGTH

Works against failures nobody anticipated — including a 500% demand surge.

ITS LIMIT

Does not help where the first failure is already unacceptable.

WHERE TO SPEND FIRST

For most early products, recovery is the better investment: it is cheaper, and it protects you from the failures you did not think of, which are the ones that actually arrive.

CONCEPT 6 · ADAPTATION

How much the system should tune itself

Instacart moved right along this line deliberately

FIXED THRESHOLDS

One setting, chosen once. Predictable, and steadily wrong as conditions move.

CONTROLLED ADAPTATION

Thresholds adjusted by experiment and feedback, within stated limits — bandits and delta levers.

UNBOUNDED SELF-TUNING

A system optimising itself with no floor. It will find the metric's blind spot.

WHAT KEEPS ADAPTATION SAFE

Control	What it prevents
A quality floor the optimiser cannot cross.	Cost or coverage gains bought with unacceptable quality.
Segment-level limits.	An overall gain that hides a badly harmed group.
A recorded reason for each change.	A configuration nobody can explain six months later.

CONCEPT 7 · FAILURE MODES

Three reliability mistakes in AI products

Each is a habit carried over from ordinary software

Measuring accuracy, ignoring age

Looks like A stable accuracy metric and unhappy users.

Why it fails The answers are correct about a world that has moved on.

Fix Report the distribution of answer age beside accuracy.

No defined degraded mode

Looks like The feature either works or throws an error.

Why it fails Under load it produces confident nonsense instead of an honest limit.

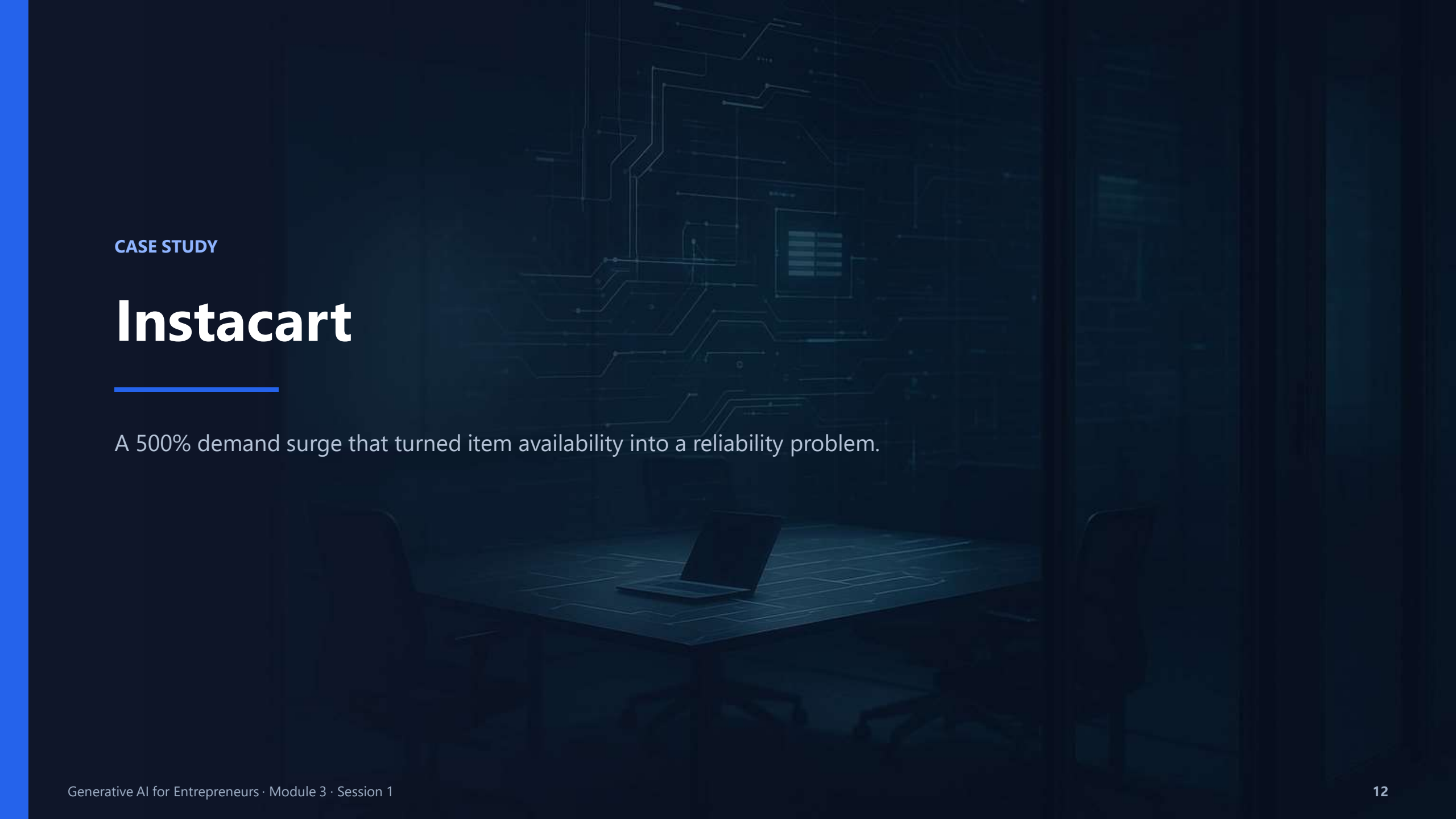
Fix Write the degradation ladder, and test the bottom rung.

Chasing perfect reliability

Looks like An unstated goal of never being wrong.

Why it fails Costs rise steeply while the last increment is worth almost nothing to users.

Fix Set an explicit objective and spend the error budget deliberately.



CASE STUDY

Instacart

A 500% demand surge that turned item availability into a reliability problem.

CASE STUDY · INSTACART

Pre-COVID reliability at much larger scale

Freshness, interface, architecture and solver — all four layers

THE SHOCK

Yesterday's signal aged too quickly

THE TRIGGER

Order volume surged 500% during COVID, creating urgent pressure on the infrastructure behind item availability.

THE PROBLEM

A model trained on older patterns became less representative as store conditions and shopping behaviour shifted.

THE TENSION

Showing more items against helping shoppers find items that could actually be picked.

THE RESPONSE

Freshness, then architecture

FIRST MOVES

Availability model runs doubled to every 60 minutes; historical lookback narrowed to 3–7 days; out-of-stock badges and filters added.

THEN THE REBUILD

Event-based processing combined with machine learning for near real-time availability scores.

THEN THE CONTROLS

Multi-model experimentation, threshold and delta levers, contextual bandits, and a commercial solver to remove timeouts.

READING THE CASE

Reliability improved to pre-COVID levels at much larger scale. The target required changes at several layers, not a better model at one of them.

Drawn from the session case pack.

APPLICATION

Bringing the theory to VerdantCart Markets

Building the reliability operating model

Step	What you decide	VerdantCart example
Name the user-visible failure	The thing customers actually experience.	An item shown as available that cannot be picked.
Set one indicator and one objective	Measurable from live traffic, agreed with the business.	99% of availability scores under one hour old, measured hourly.
Measure answer age	Separately from accuracy.	Age distribution reported next to accuracy in the weekly loop.
Write the degradation ladder	Four rungs, down to a safe refusal.	Full score, flagged low confidence, last known good, hide the item.
Attach a consequence	So the objective is real.	Budget exhausted freezes launches until it recovers.
Test the fallback on purpose	Monthly, in production conditions.	A scheduled exercise that disables the scorer for ten minutes.

WHAT COMES NEXT

Session 3.2 asks what all of this costs per request — and what becomes possible when that number falls.

SUMMARY

What to carry out of this session

1. **Define reliability as a number.**
   Indicator, objective, error budget — then the trade-offs become decisions.

2. **Freshness is a separate failure from accuracy.**
   Measure the age of answers alongside how often they are right.

3. **Choose the degraded mode in advance.**
   Under load a system with no defined fallback produces confident nonsense.

4. **Reliability is won at four layers.**
   Data path, model, interface and infrastructure — Instacart used all of them.

5. **Recovery usually beats prevention early.**
   It is cheaper, and it works against the failures you did not predict.

FURTHER READING

Sources for this session

Primary references behind each concept

SERVICE OBJECTIVES	Beyer, B., Jones, C., Petoff, J. & Murphy, N. (2016). *Site Reliability Engineering.* Sebastopol: O'Reilly.
PRACTICE	Beyer, B. et al. (2018). *The Site Reliability Workbook*. Sebastopol: O'Reilly.
RESILIENCE	Hollnagel, E., Woods, D. & Leveson, N. (2006). *Resilience Engineering.* Aldershot: Ashgate.
NORMAL ACCIDENTS	Perrow, C. (1984). *Normal Accidents.* New York: Basic Books.
ML SYSTEM DEBT	Sculley, D. et al. (2015). Hidden technical debt in machine learning systems. *NeurIPS 28*.
ADAPTIVE CONTROL	Li, L. et al. (2010). A contextual-bandit approach to personalized news article recommendation. *WWW 2010*.
ML SYSTEMS	Huyen, C. (2022). *Designing Machine Learning Systems.* Sebastopol: O'Reilly.