GUIDE F1

TURN8

HOW TO **RUN A STRUCTURED PRODUCT DEVELOPMENT SPRINT** TOWARD AN INVESTABLE MVP

Product Development, Piloting & First Customer Acquisition

WHERE THIS IS USED

- Venture Studio programs
- Corporate Incubators
- Accelerators (corporate or government-backed)
- CVC portfolio companies
- AI Studio deployments
- Foundry-as-a-Service engagements

AUDIENCE

- EIRs / GMs
- Product Leads
- Tech Leads
- AI Studio Agent Leads
- CX Designers / Researchers
- Venture Studio Program Managers

PHASE

Phase Three: Build and Launch → MVP Build Sprint (Weeks 2–10)

1

EXECUTIVE SUMMARY

A sprint is a learning cycle — not a delivery cycle. This distinction is the most important concept in this guide. A delivery cycle produces a feature. A learning cycle produces evidence that a feature is worth building and then builds the smallest version of it that tests the riskiest assumption. Done correctly, a 6–10-week sprint produces a pilot-ready product and a documented set of learnings that either confirm the investment case or prevent capital from being wasted on the wrong direction.

This guide applies the Build–Measure–Learn framework to the Phase Three product sprint, adapted for three types of ventures that TURN8 works with: standard product ventures (features, UX, core workflow), AI-enabled ventures (where the Build–Measure–Learn loop operates on prompts, models, and data pipelines in addition to features), and AI Studio agent ventures (where the primary deliverable is a configured, tested, and instrumented agent rather than a traditional software product).

For all three types, the same core principles apply: start from the problem, not the model or the tech stack. Use the smallest viable thing — the Minimum Viable Product and, for AI ventures, the Minimum Viable Model. Prefer off-the-shelf APIs before custom training. Instrument every interaction to generate learning data. Run disciplined experiments to decide whether to persist or pivot. Protect the sprint from feature creep with a formal scope lock that keeps every decision traceable to the riskiest assumption from Guide B1.

2

THE CORE PROBLEM

The venture's job in Phase Three is to move from "countless risks, no evidence" toward "controlled risk, growing evidence." Every week of the sprint must reduce uncertainty. Most corporate MVP builds do the opposite.

The failure patterns split across two types:

Standard product ventures:

- Teams build "polished MVPs" that include every stakeholder request. The riskiest assumption is buried under a wish list. The product cannot be built in 6–10 weeks and cannot be used to test anything specific.
- Sprint timelines extend indefinitely because scope was never locked. A product that should take 6–10 weeks takes 6–12 months — capital is consumed before the first external user sees anything.
- Design and Engineering work sequentially, not in parallel. Design produces screens that cannot be implemented in the sprint window.

AI-enabled ventures:

- Teams start from the model or the tech stack rather than the problem. They build AI infrastructure before confirming anyone will use the product.
- The Build–Measure–Learn loop is applied to features only — not to prompts, models, and data pipelines. AI components degrade silently because no feedback loop was designed.
- Off-the-shelf API capabilities are bypassed in favor of custom model development before the use case is even validated. This is the AI equivalent of building the full product before testing the MVP.
- Instrumentation is treated as post-sprint work. Logging, feedback capture, latency monitoring, and model performance tracking are not planned as sprint scope — and so the venture produces a product but no learning data.

Both types share one underlying cause:

The sprint was designed as a delivery mechanism, not as a learning cycle. When the sprint ends, the team asks, "what did we build?" instead of "what did we learn, and what do we now know that we did not know before?" This guide resets that framing from sprint day one.

PREREQUISITES

Must Be Complete Before Starting:

- Guide B1 completed — signed MVP Specification with riskiest assumption, core use case, feature set (5–7 max), explicit out-of-scope list, success metrics, timeline, and budget
- Guide B2 completed — Usability and Validation Report from the clickable prototype phase; core use case confirmed before full build begins
- Guide B3 completed (AI ventures) — AI Studio agent architecture designed; workflow automation candidates mapped; system prompts drafted and tested against a 50-input test set
- Guide E1 completed — Product Lead, Tech Lead, and AI Studio Agent Lead (if applicable) confirmed with mandate letters and 100% allocation
- Guide E3 completed — any data assets, APIs, or corporate infrastructure required for the MVP have formal access agreements in place

The single most important prerequisite:

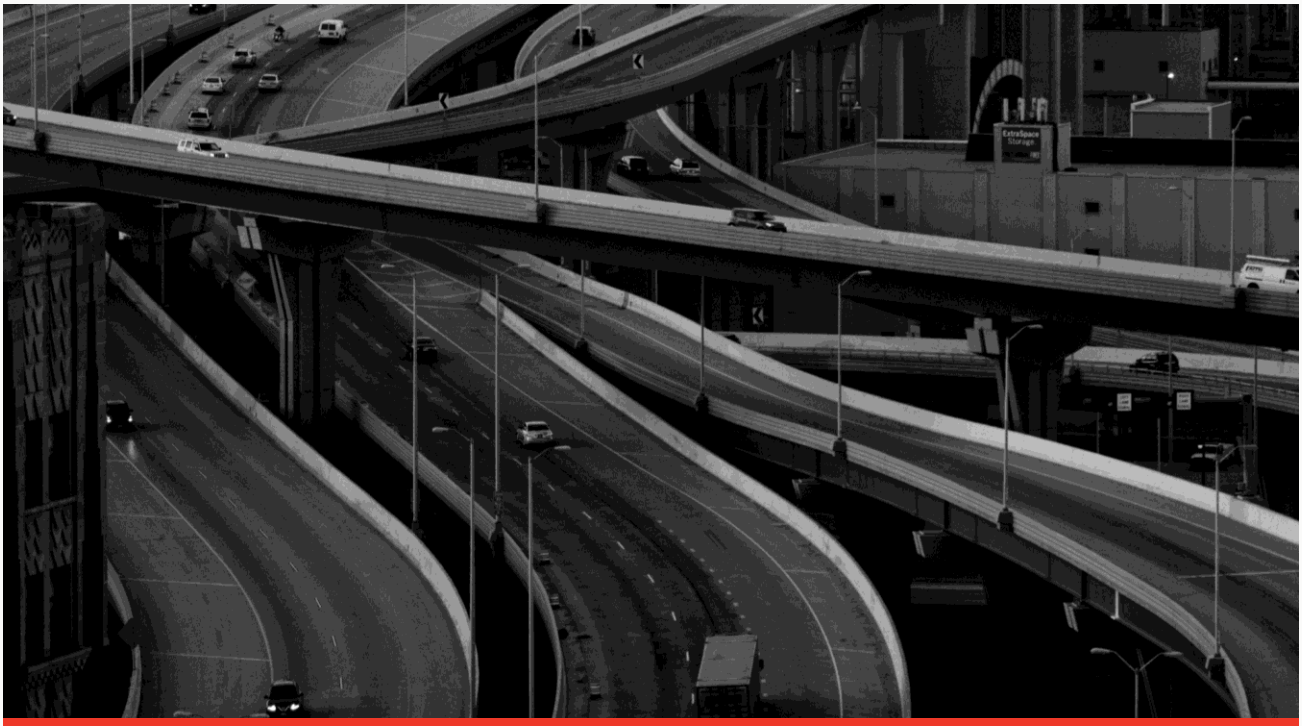
The riskiest assumption must be written in one sentence and confirmed by the Executive Sponsor in writing before the sprint board opens. Not a goal. Not a vision statement. A falsifiable claim: "We believe [customer] will [behavior] because [reason]. We will know this is true when [specific metric] reaches [specific threshold]." Everything in the sprint either tests this assumption or is out of scope.

4

EXPECTED OUTPUT/ SUCCESS CRITERIA

You have completed this guide when the following are true:

- ☑ The MVP is built and passes the internal usability test — every team member completes the core use case without assistance
- ☑ The product communicates its value: users can articulate what problem it solves and what they can now do that they could not do before
- ☑ Every feature is traceable to the riskiest assumption — nothing was built that does not test something specific
- ☑ For AI ventures: the Minimum Viable Model is live, instrumented, and producing feedback data from every interaction
- ☑ For AI Studio agent ventures: at least two agents are deployed, tested against real-world inputs, and producing > 90% accuracy on standard cases
- ☑ A Sprint Completion Report is produced and referenced in the Go / Hold / Stop decision for pilot launch
- ☑ At least one pilot participant is confirmed and ready to receive the product within 5 business days of sprint completion



STEP-BY-STEP INSTRUCTIONS

STEP 1 ALIGN THE SPRINT ON THE RISKIEST ASSUMPTION

Before any product decision is made, every team member must be able to recite the riskiest assumption from memory. Not a paraphrase — the exact statement from the B1 spec. This is not ceremonial. It is the operational anchor that filters every sprint decision for the next 6–10 weeks.

1.1 **Open the sprint with a reading of the B1 riskiest assumption** — Every team member reads it aloud. Display it visibly in the workspace — physical whiteboard, sprint board header, or project management tool banner — for the full sprint duration

1.2 **Distinguish between the MVP and the MVM for AI ventures** — These are two separate sprint deliverables that must both be planned

- **MVP (Minimum Viable Product):** The smallest workflow that delivers value to the user. Even if there is no "real" AI yet — it could run on rules, scripts, or a manual process behind the scenes. The MVP exists to test whether users will engage with the workflow.
- **MVM (Minimum Viable Model):** The simplest AI model that supports the MVP workflow. Almost always an off-the-shelf API call at this stage: one LLM API call, one vision API, one search API. The principle is "one prompt, one call, one metric." The MVM exists to test whether the AI component produces outputs users trust and act on.

1.3 **Confirm the sprint type: standard product, AI-enabled, or AI Studio agent** — The sprint type determines which tracks run in parallel during the build. A standard product sprint runs one track. An AI-enabled sprint runs two parallel tracks: the product UX track and the AI/model track. An AI Studio agent sprint may run the agent track as the primary deliverable, with UX as secondary

1.4 **Assign named owners — one per feature, one per AI component** — For AI ventures: the AI component is not owned by "the tech team." One person — the AI Studio Agent Lead or the Tech Lead — owns the MVM or agent specification, the system prompt, the test set, and the instrumentation plan

AI PROMPT — Sprint Type Confirmation and Riskiest Assumption Alignment

```
I am opening a Phase Three product sprint. The venture is
[describe]. The riskiest assumption from B1 is [paste]. The sprint
type is [standard product / AI-enabled / AI Studio agent]. For each
feature or AI component in the sprint backlog: (1) confirm in one
sentence how it tests the riskiest assumption, (2) assign a named
owner, (3) for AI components: confirm whether the capability will be
served by an off-the-shelf API (name it) or requires custom model
development (justify why the API is insufficient). Flag any item
that cannot be linked to the riskiest assumption — those are Out of
Scope candidates.
```

STEP 2 DESIGN THE BUILD–MEASURE–LEARN LOOP BEFORE BUILDING ANYTHING

The Build–Measure–Learn loop is not a retrospective process — it is a planning tool. Before any build work begins, the team defines what they will measure and what they will learn from each measurement. A sprint that begins without defined metrics produces output, not evidence. In corporate venture programs, "output without evidence" is the most common cause of premature scaling.

2.1 **Define three metrics before the sprint board opens** — Not after the sprint. Not at the first review. Before the first task is started

METRIC TYPE	WHAT IT MEASURES	STANDARD PRODUCT EXAMPLE	AI-ENABLED PRODUCT EXAMPLE
Task Success Metric	Does the user complete the core use case without assistance?	Task completion rate > 70% in internal usability testing (from B2 protocol)	User correctly submits a query and receives a relevant AI output without requiring reformulation or abandonment
Product / Model Performance Metric	Does the product or AI component perform correctly?	Core feature acceptance criteria: all 5 "Given-When-Then" criteria pass	AI accuracy rate > 90% on 50-input test set (20 standard, 20 edge, 10 adversarial); escalation rate < 10%; response latency < 3 seconds
Business Outcome Metric	Does this move the investment case forward?	At least 3 of 8 internal testers state willingness to continue using the product	At least 3 of 8 testers confirm the AI output saved them time or improved a decision. At least 1 requests continued access after the sprint.

2.2 **Design the instrumentation plan as sprint scope** — not a post-sprint task — For AI ventures: every interaction must generate learning data. This means planning the logging architecture before building. What gets logged: user inputs, model outputs, user corrections, latency, good/bad feedback signals. Who owns the data: named data owner. Where it is stored: confirmed before sprint day one

2.3 **Create feedback paths users can engage with** — Users must be able to signal "good / bad / needs change" with minimal friction. This is not user research — it is a labeled dataset being created in real time. Every good/bad signal is a training data point for future prompt iteration or fine-tuning. Design this path as part of the UX, not as an afterthought

2.4 **Identify the AI type: discovery-oriented or optimization-oriented** — These two types have different Build-Measure-Learn rhythms and different sprint structures

- **Discovery-oriented AI:** Finding new opportunities, analyzing markets, generating insights. Pairs with lean discovery and prototyping. AI expands the search space; experiments test which opportunities are real. Sprint focus: does the AI output surface information the user could not find on their own?
- **Optimization-oriented AI:** A/B testing, ranking, recommendations, targeting. Accelerates iterative refinement. Sprint focus: does the AI configuration measurably outperform the baseline? Use A/B tests and cohort tests to compare prompt variants and model configurations rather than building big releases.

2.5 **For standard product ventures: define the "product verbs"** — List every observable action the product enables that the user cannot currently perform. For each verb: the before state, the user interaction, and the after state. These three elements must all be visible in the live MVP for the acceptance criteria to be met

Reference: AI Lean Canvas — Required for All AI-Enabled Ventures

Before the sprint board opens, complete the four AI-specific additions to the standard lean canvas. These four dimensions govern the sprint architecture for AI components:

CANVAS DIMENSION	WHAT TO DEFINE	SPRINT IMPLICATION
Data Sources	Where does the data come from? Who owns it? What are the quality and coverage risks? Is it the corporate asset identified in Guide E3, a public dataset, or real-time user inputs?	Data access must be formally confirmed (E3 agreement) before any AI feature enters In Progress. Data quality issues discovered mid-sprint are a Stop signal.
Model Choice	Off-the-shelf API vs custom model. Latency and cost constraints. Explainability requirements for regulated sectors. Which specific API or model will be used first?	The MVM starts with an off-the-shelf API. Custom model development is not sprint scope. Document the API name, endpoint, and cost model in the sprint card.
Feedback and Learning Loop	How is new data captured after each user interaction? How is it labelled? How does it feed back into prompt updates or future fine-tuning?	The feedback loop must be designed and instrumented in the sprint. It is not a Phase Four deliverable. User corrections and good/bad signals are the labeled dataset for the next iteration.
Risk and Ethics	What are the bias risks? What happens when the model misclassifies? Are there regulatory or sector-specific AI constraints (e.g. financial services, healthcare, HR automation)?	Known bias and misclassification risks must be documented before deployment. Define the escalation rule: under what condition does the AI hand off to a human rather than producing an output?

AI PROMPT — AI Lean Canvas Completion

```
I am completing the AI Lean Canvas for a Phase Three sprint.
Venture: [describe]. Riskiest assumption: [paste]. AI type:
[discovery-oriented / optimization-oriented]. Complete the four
AI canvas dimensions: (1) Data Sources — where does the data come
from, who owns it, what are the quality and coverage risks, is
there a formal E3 access agreement in place? (2) Model Choice —
which off-the-shelf API will serve the MVM, what are the latency
and cost constraints, are there explainability requirements? (3)
Feedback and Learning Loop — how are user interactions captured
and labelled, what is the feedback path design, how do
corrections feed back into prompt updates? (4) Risk and Ethics —
what are the bias and misclassification risks, what is the
escalation rule when the model produces a low-confidence output?
Output as a completed canvas table, not a discussion.
```

AI PROMPT — Build-Measure-Learn Loop Design

```
I am designing the Build-Measure-Learn loop for a Phase Three sprint. Venture: [describe]. Riskiest assumption: [paste]. Sprint type: [standard / AI-enabled / AI Studio agent]. Define: (1) the three metrics for this sprint — task success, product/model performance, and business outcome — with specific thresholds, (2) the instrumentation plan for AI components: what is logged, by whom, where it is stored, how feedback is captured, (3) the feedback path design — how users signal good/bad/change within the product flow, (4) for standard product: list the "product verbs" with before state, user interaction, and after state for each. This plan is reviewed before any build work begins.
```

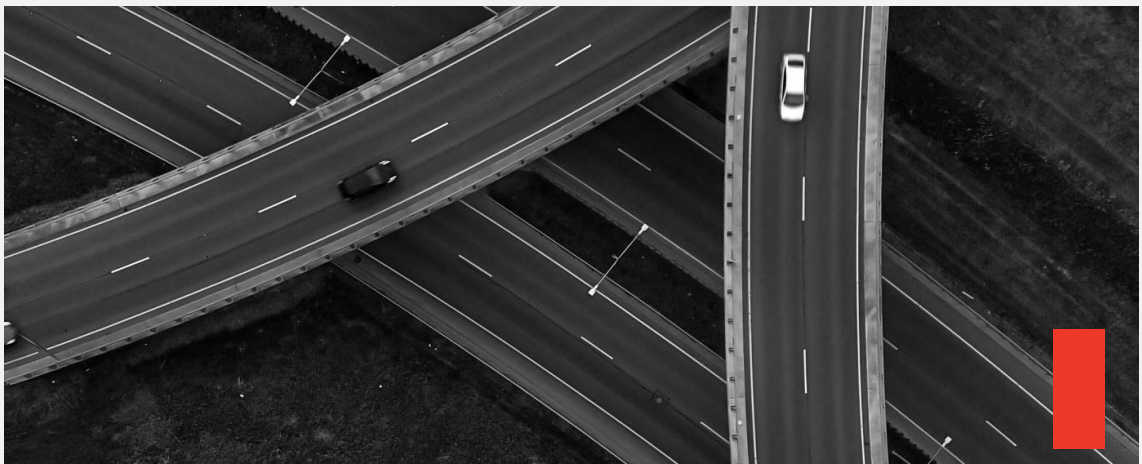
STEP 3 OPEN THE SPRINT BOARD AND ENFORCE SCOPE LOCK

The sprint board is the single source of truth for the entire build. It replaces all informal task lists, email threads, and verbal agreements. Anything not on the sprint board does not exist in this sprint. The scope is fixed from the moment the board opens.

3.1 **Set up the sprint board with the correct column structure** — Columns: Backlog | In Progress | In Review | Done | Out of Scope. Every feature from B1 and every AI component from B3 is a card. Every card has: name, acceptance criterion, named owner, sprint group (1 / 2 / 3), and link to the riskiest assumption it tests

3.2 **Order the backlog by three groups** — Group 1 — Core use case: features and AI components required to complete the primary workflow. These are the non-negotiable items. Group 2 — Value enhancement: items that improve the core use case but do not block it. Group 3 — Extensions: items requested but not essential. Only Group 1 is protected from the 50% cut rule

3.3 **Apply the MVM principle to AI components at sprint planning** — For every AI feature: can this be served by an existing off-the-shelf API? If yes, use the API. Custom model development is not sprint scope unless the team has already validated demand and hit a quality or cost ceiling with available APIs. "We can do better with our own model" is not a sprint justification — it is a Phase Four justification



3.4 Enforce the Change Control Protocol for every scope change

STEP	ACTION	OWNER	TIME LIMIT
1	Requester writes what is requested, which assumption it tests, which approved item it replaces, and why it is more important for testing the assumption than the item it replaces	Anyone	24 hours from request
2	Product Lead reviews: is this change more important for the riskiest assumption than what it replaces?	Product Lead	24 hours
3	Tech Lead (+ AI Studio Agent Lead if AI feature) confirms feasibility within the remaining sprint window without extending the end date	Tech Lead / AI Lead	24 hours
4	Executive Sponsor provides written approval	Executive Sponsor	48 hours
5	Sprint board updated: old item moves to Out of Scope, new item enters Backlog with full card	Product Lead	Immediately after approval

3.5 **Apply the 50% cut rule at the first overrun signal** — If any sprint week ends with fewer than 50% of that week's tasks marked Done, cut all Group 2 and Group 3 items from the backlog immediately. Redirect 100% of capacity to Group 1. Do not extend the sprint end date

STEP 4 RUN PARALLEL SPRINT TRACKS FOR AI-ENABLED VENTURES

AI-enabled ventures run two parallel tracks throughout the sprint: the product UX track (building the workflow the user interacts with) and the AI / model track (building and iterating on the intelligence that powers the product). These tracks are interdependent but have different rhythms and different failure modes. They require different owners and different review cadences.

TRACK A — PRODUCT UX TRACK (OWNED BY PRODUCT LEAD + CX DESIGNER):

- Build the core user workflow from the B1 spec using the B2 prototype as the design reference
- Apply real content from Day 1 — real product names, real pricing, real sample data. No placeholder text.
- Run a joint Design + Engineering review for every feature card before it enters In Progress
- Internal team usability test at mid-sprint: every team member completes the core use case without guidance. Every failure point is a blocking issue — resolve before proceeding

TRACK B — AI / MODEL TRACK (OWNED BY TECH LEAD + AI STUDIO AGENT LEAD):

- Start with the smallest viable model: "one prompt, one call, one metric." Use an off-the-shelf API unless a specific validated reason exists to do otherwise
- Write the system prompt using the B3 agent specification format. Deploy it. Test it against 50 real-world inputs (20 standard, 20 edge cases, 10 adversarial). Target: > 90% accuracy on standard cases
- Instrument every interaction from Day 1: log inputs, model outputs, user corrections, response latency, and basic business metrics. This is sprint scope, not Phase Four scope
- Create the feedback loop: a lightweight mechanism for users to signal good / bad / needs change. Every signal is a labeled data point for future prompt iteration
- Only invest in custom model training when: the use case is validated by pilot data, the off-the-shelf API has hit a confirmed quality or cost ceiling, and there is sufficient data to train or fine-tune. Until all three conditions are met, do not build custom models

WEEK	PRODUCT UX TRACK MILESTONE	AI / MODEL TRACK MILESTONE	JOINT CHECKPOINT
1–2	Core use case wireframes confirmed; design–engineering handoff for Group 1 features	System prompt written and tested against 50-input test set; API integration confirmed	Design + engineering + AI feasibility review for every Group 1 card
3–4	Group 1 features built; internal usability test (target: 100% task completion)	AI component integrated into Group 1 features; accuracy > 90% on standard cases; instrumentation live	Mid-sprint checkpoint: Go / Hold / Stop decision; 50% cut rule applied if < 70% complete
5–6	Group 2 features built; core use case fully functional end-to-end	Feedback loop active; edge cases and failure modes documented; escalation protocol tested	Second internal usability test with AI outputs included in sessions
7–8	Final internal test; acceptance criteria verified for all delivered features	AI performance dashboard live; first week of real interaction data collected	Sprint Completion Report drafted; pilot readiness confirmed

STEP 5 RUN WEEKLY CHECKPOINTS AS GO / HOLD / STOP DECISIONS

Sprint checkpoints are governance decisions, not status meetings. Every checkpoint uses the same Go / Hold / Stop framework applied across Phase Three (Guide E2). The sprint is a governance-managed build cycle — and every week, the decision is whether to continue as planned, intervene, or replan.

5.1 **Hold a weekly sprint checkpoint — 30 minutes, same day, same time** — Attendees: Product Lead, Tech Lead, AI Studio Agent Lead (if applicable). Three questions only: (1) What was planned? (2) What was done? (3) What is blocked?

5.2 **Apply the three-tier decision framework**

SIGNAL	DECISION	IMMEDIATE ACTION
≥ 90% of week's tasks Done. Core use case progressing. AI track on target.	Go	Continue. Update the sprint dashboard. Confirm next week's priorities for both tracks.
70–89% complete. One track at risk but core use case unaffected. AI accuracy below target but improving.	Hold	Name the specific blocker. Assign a resolution owner and 48-hour deadline. If unresolved in 48 hours, apply the 50% cut rule to the affected area.
< 70% complete. Core use case at risk. AI component failing standard test cases. A technical dependency has broken.	Stop and replan	Apply 50% cut rule immediately. Cut all Group 2 and 3 items. Rebuild the backlog around the core use case only. Reset internal milestone dates. Do not extend the sprint end date.

5.3 **Hold the mid-sprint checkpoint as a formal governance review** — At the halfway point (week 4 of 8): core use case must be ≥ 70% complete. AI component must be integrated and producing outputs at > 90% accuracy on standard cases. If either condition is not met, apply Stop and Replan — do not wait for week 6

5.4 **Document every checkpoint decision in the sprint decision log** — Format: "Sprint Week [N] Checkpoint — Decision: [Go / Hold / Stop]. Evidence: [% complete, AI accuracy, blockers]. Action: [specific next step, owner, date]." This feeds into the E2 decision log and the quarterly board pack

STEP 6 PROTECT THE SPRINT FROM FEATURE CREEP AND PREMATURE AI COMPLEXITY

Feature creep and premature AI complexity are the two specific threats to the Phase Three sprint. Feature creep adds scope without evidence. Premature AI complexity adds infrastructure without validation. Both consume the same resource: sprint capacity and runway.

6.1 **Display the riskiest assumption in every stakeholder communication** — Top of every sprint update email, every checkpoint summary, every board report section covering product progress. When a request arrives, the first question is: "Does this change how we test the riskiest assumption?"

6.2 **For AI ventures: enforce the off-the-shelf API first rule** — The MVM uses the simplest available model. This is not a temporary compromise — it is the correct engineering decision at this stage. Validate demand with an API. Switch to a custom model only when the API has hit a confirmed ceiling and the data exists to train something better

6.3 **Distinguish feature requests from customers vs. stakeholders** — Customer-driven requests (from B2 testing or early pilot conversations) are evidence. They are evaluated on merit and may enter the next sprint through Change Control. Stakeholder-driven requests (from the Executive Sponsor, BU heads, or the team itself) are hypotheses. They require customer validation before entering the backlog

6.4 **Reframe "the MVP is too simple" with the de-risking argument** — When a stakeholder pushes back on scope: "This is not a product launch. It is a learning cycle. The goal is to test one specific assumption as quickly as possible. If the assumption is confirmed, we build the full product. If it is disproven, we have saved months of build time and preserved the capital for a better opportunity"

STEP 7 COMPLETE THE SPRINT AND PRODUCE THE SPRINT COMPLETION REPORT

The Sprint Completion Report is the formal transition document from build to pilot. It confirms the MVP is pilot-ready, documents what was learned during the sprint, and feeds directly into the Guide F2 pilot program and the investment case for Tranche 1 milestone confirmation.

7.1 **Run the final internal usability test on the last day of the sprint** — Every team member completes the core use case without assistance. Target: 100% task completion internally. For AI ventures: every AI-assisted step must work end-to-end without manual intervention

7.2 **Verify all acceptance criteria are met — both UX and AI components** — Walk every card on the sprint board. For each: acceptance criterion verified? For AI cards: accuracy threshold met? Instrumentation live? Feedback loop active? Any unmet criterion is a blocking item before the report is signed

Sprint Completion Report — Required Structure:

Section	Content	Length
1. Riskiest Assumption	The B1 riskiest assumption, unchanged. Confirms the sprint remained focused on the right thing.	1 sentence
2. Sprint Type	Standard product / AI-enabled / AI Studio agent. Tracks run and their outcomes.	2 sentences
3. Features and AI Components Delivered	All items delivered with acceptance criteria status (Met / Not Met). Items moved to Out of Scope with reason. For AI: accuracy rate, escalation rate, latency.	½ page — table
4. Build–Measure–Learn Summary	For each week: the hypothesis tested, what was built, what was measured, and the specific learning. Three metrics across the sprint: task success, product/model performance, business outcome.	½ page
5. Internal Usability Test Results	Task completion rate, failure points identified and resolved, protocol used.	¼ page
6. AI Instrumentation Status	For AI ventures: logging confirmed live, feedback loop active, first interaction data collected, model performance dashboard operational.	¼ page
7. Scope Changes	All Change Control Protocol decisions during the sprint — request, decision, outcome.	¼ page
8. Pilot Readiness Statement	Binary: pilot-ready or not. If pilot-ready: named participants confirmed, first session date set.	¼ page
9. Next Sprint Learning Input	Top 3 things learned during this sprint that change the next sprint's hypothesis or backlog. This is the "Learn" output of the Build–Measure–Learn loop.	¼ page

7.3 **Obtain Executive Sponsor written sign-off before pilot launch** — Email confirmation is sufficient. The signed report is filed in the content library (from Guide E2) and referenced in the F2 pilot program

AI PROMPT — Sprint Completion Report

```
I am producing the Sprint Completion Report for a Phase Three MVP sprint. Venture: [describe]. Sprint type: [standard / AI-enabled / AI Studio agent]. Riskiest assumption: [paste]. Features delivered: [list with acceptance criteria status]. AI component results: [accuracy rate, escalation rate, latency, instrumentation status]. Internal usability test: [task completion rate, failure points, resolution]. Scope changes: [list]. Pilot participant status: [confirmed / pending]. Write the complete 9-section Sprint Completion Report. Include a Build-Measure-Learn summary for each sprint week. The report must be readable by an Executive Sponsor who was not in the sprint — format accordingly. End with a binary pilot readiness statement.
```

TROUBLESHOOTING

SYMPTOM	LIKELY CAUSE	FIX
The team is building AI infrastructure before confirming the use case works	Starting from the model or tech stack rather than the problem. The MVM principle was skipped.	Stop the infrastructure build. Return to Step 2. Can the use case be served by a single off-the-shelf API call? If yes, use that first. Custom model development begins only after demand is validated and the off-the-shelf API has hit a quality or cost ceiling
Feature creep from internal stakeholders mid-sprint	No formal scope lock; the riskiest assumption is not visible; stakeholders optimize for presentation quality, not learning	Display the riskiest assumption at the top of every sprint update. Apply the Change Control Protocol (Step 3). Reframe every addition: "Which approved feature does this replace, and what evidence from a customer supports this change?"
AI model produces inconsistent outputs in user testing sessions	System prompt is under-specified; edge cases not handled; no fallback or escalation rule defined	Revise the system prompt using the B3 agent specification format. Add explicit escalation rules. Run the 50-test set (20 standard, 20 edge cases, 10 adversarial) and measure accuracy before any external user session
The sprint timeline extends because the AI component takes longer than expected	AI instrumentation (logging, feedback capture, latency monitoring) was not planned as sprint scope — it was treated as an afterthought	Instrument for learning is sprint scope, not post-sprint scope. Logging inputs, model decisions, user corrections, latency, and basic business metrics must be in the sprint backlog from Day 1
The MVP works technically but users do not believe the product can deliver the promised outcome	The "product verbs" — the visible before-and-after demonstration of value — are absent. The product functions but does not communicate	Apply the three product design principles. Rewrite acceptance criteria to include the Believable test: "When the user completes the core use case, they must be able to articulate what outcome they achieved."
The sprint produces a product but no learning — the Build-Measure-Learn loop did not close	The Measure phase was not planned. Metrics were not defined before building. The sprint produced output, not evidence.	Define the three metrics before building anything: (1) the primary task success metric, (2) the model/AI performance metric, (3) the business outcome metric. Every sprint cycle must close the loop: Hypothesis → Build → Measure → Learn → Update
Corporate BU requests an AI feature that is not in the MVP spec	BU sees a demo and wants the AI to do additional things. This is the AI version of standard feature creep.	AI feature requests from BUs are treated the same as any scope change request. They must go through the Change Control Protocol and be traceable to the riskiest assumption. "It is just a prompt change" is not sufficient justification
The AI component degrades over time after initial pilot success	No continuous learning mechanism designed; user corrections and feedback are not captured and fed back into the model	Design the feedback loop before deployment: every interaction must generate data. User corrections, good/bad signals, and edge cases are labeled data for prompt improvement or fine-tuning. The feedback path is sprint scope, not Phase Four scope

7

VALIDATION STEPS

Confirm each of the following before declaring the sprint complete and the product pilot-ready:

- [ ] Riskiest assumption visible on the sprint board and in all checkpoint summaries throughout the sprint
- [ ] Every delivered feature and AI component is traceable to the riskiest assumption
- [ ] Internal task completion rate is 100% — every team member completed the core use case without assistance
- [ ] For AI ventures: AI accuracy > 90% on standard test cases, instrumentation is live, feedback loop is active
- [ ] All acceptance criteria verified — UX and AI components
- [ ] Every scope change was processed through the Change Control Protocol with written Executive Sponsor approval
- [ ] Sprint Completion Report produced within 3 business days of sprint end, including Build–Measure–Learn summary
- [ ] Executive Sponsor has signed off on the Sprint Completion Report in writing
- [ ] At least one F2 pilot participant is confirmed and briefed

NEXT STEPS

Upon completing this guide with a pilot-ready product, proceed immediately to:

- **GUIDE F2** — How to Design and Execute a Pilot Program That Generates Investable Evidence
- **GUIDE F3** — How to Acquire Your First Customers and Build the Customer Zero Playbook (run in parallel with F2 for participant recruitment)

If the sprint closes with a Stop signal — the core use case is not working after the mid-sprint checkpoint and the 50% cut rule has been applied — escalate to the Venture Board per the E2 protocol before any further build. Document the top 3 learnings in the format: "We hypothesized X. We observed Y. We now believe Z." Return to the B1 specification with a revised riskiest assumption. A sprint Stop is the de-risking machine working correctly — it preserves capital and produces a specific, actionable learning.

MASTER CHECKLIST

A. SPRINT ALIGNMENT

☐ Riskiest assumption from B1 confirmed in one falsifiable sentence and approved by Executive Sponsor in writing

☐ Sprint type confirmed: standard product / AI-enabled / AI Studio agent

☐ Every team member can recite the riskiest assumption from memory before the sprint board opens

☐ For AI ventures: MVP and MVM distinguished as two separate sprint deliverables with separate named owners

☐ MVM confirmed as off-the-shelf API — custom model development is not sprint scope unless validated demand and confirmed API ceiling both exist

B. BUILD–MEASURE–LEARN LOOP DESIGN AND AI LEAN CANVAS

☐ Three metrics defined before any build work begins: task success, product/model performance, business outcome

☐ Specific thresholds set for each metric — binary pass/fail lines, not directional goals

☐ AI type confirmed: discovery-oriented (surfacing insights) or optimization-oriented (A/B testing, ranking, recommendations)

— AI LEAN CANVAS (required for AI-enabled ventures) —

☐ Data Sources confirmed: data origin, ownership, quality/coverage risks, E3 access agreement in place

☐ Model Choice confirmed: specific off-the-shelf API named; custom model development ruled out unless validated demand and confirmed ceiling both exist

☐ Feedback and Learning Loop designed: user corrections captured, good/bad signals stored as labeled data, prompt update cadence defined

☐ Risk and Ethics documented: bias risks, misclassification risks, regulatory constraints, escalation rule defined

— STANDARD PRODUCT ONLY —

☐ Instrumentation plan designed as sprint scope: logging architecture, feedback capture, data storage, named data owner

☐ User feedback path designed in the UX: good/bad/change signals captured with minimal friction

☐ "Product verbs" listed: before state, user interaction, and after state for each

☐ For AI Studio agents: 50-input test set created (20 standard, 20 edge, 10 adversarial) before sprint begins

C. SPRINT BOARD AND SCOPE LOCK

☐ Sprint board open with correct column structure: Backlog | In Progress | In Review | Done | Out of Scope
☐ All B1 features and B3 AI components on the board as cards with acceptance criteria, owner, and group assignment
☐ Backlog ordered: Group 1 (core use case) → Group 2 (enhance) → Group 3 (extend)
☐ Change Control Protocol template in place and briefed to all team members and stakeholders
☐ 50% cut rule briefed: Group 2 and 3 are cut at the first overrun signal without exception
☐ Off-the-shelf API first rule confirmed for all AI components in the sprint

D. PARALLEL TRACKS (AI-ENABLED AND AI STUDIO AGENT VENTURES)

☐ Product UX Track: Product Lead + CX Designer confirmed as owners
☐ AI / Model Track: Tech Lead + AI Studio Agent Lead confirmed as owners
☐ Joint Design + Engineering + AI review run for every Group 1 card before In Progress
☐ System prompt written using B3 agent specification format
☐ System prompt tested against 50-input test set before any user session begins
☐ Instrumentation live from Day 1: inputs, outputs, corrections, latency logged
☐ Feedback loop integrated into the UX — good/bad signals captured per interaction

E. WEEKLY CHECKPOINTS

☐ Weekly sprint checkpoint booked — same day, same time, 30 minutes maximum
☐ Three-question format used: planned vs. done vs. blocked
☐ Mid-sprint checkpoint (week 4 of 8) held — Go / Hold / Stop decision documented
☐ Core use case ≥ 70% complete at mid-sprint — Stop and Replan applied immediately if not
☐ AI accuracy ≥ 90% on standard test cases at mid-sprint
☐ Every checkpoint decision documented in the sprint decision log
☐ Stop and Replan activated when < 70% of week's tasks done — 50% cut rule applied

F. SCOPE PROTECTION

☐ Riskiest assumption at the top of every stakeholder communication during sprint
☐ "Too simple" objections documented and reframed with the de-risking argument
☐ All stakeholder-driven feature requests logged and deferred pending customer validation
☐ All customer-driven requests from B2 evaluation processed through Change Control Protocol
☐ MVM off-the-shelf API confirmed — no custom model development without validated demand and confirmed ceiling

G. SPRINT COMPLETION

☐ Final internal usability test completed — 100% task completion by all team members
☐ All acceptance criteria verified — UX and AI components
☐ AI instrumentation confirmed live: logging, feedback loop, performance dashboard
☐ Sprint Completion Report produced within 3 business days, including Build–Measure–Learn summary
☐ Executive Sponsor sign-off on Sprint Completion Report obtained in writing
☐ Top 3 sprint learnings documented in "We hypothesized X / observed Y / now believe Z" format
☐ At least one F2 pilot participant confirmed and briefed