PROJECT 5 OF 5 | START HERE LEARNER GUIDE

AI Governance Assessment

Engineer governance

Harborline Health piloted Member Assist, an AI-assisted Member Services workflow, before finishing a formal AI governance review. The business likes the pilot and wants to expand it. You arrive after the pilot is already running, and you govern the use case: not the vendor, not a generic policy, and not a framework checklist.

> **Companion video**
> Watch the walkthrough before you begin: https://youtu.be/8xczWFO5gvs

Your project folder

Item	What it is for
START_HERE_Learner_Guide.pdf	This guide.
Master_Workbook.xlsx	Your workspace, including the spreadsheet Governance as Code tabs.
1_Assess_Evidence	The twelve-file Member Assist packet for Stage 1.
4_Break_Incident_Evidence_Open_After_Build	New facts for Stage 4. Do not open until your initial recommendation exists.
Deliverable_Templates	Word template: the AI Governance Decision Brief (Stage 6).
Optional_Python_Governance_as_Code	Optional Deep Dive: the same rules in beginner-readable Python.

Contents

Welcome to Project 5 . . . 3
Stage 1: Assess . . . 5
Stage 2: Design . . . 7
Stage 3: Build (Analyze) . . . 8
Stage 4: Break . . . 10
Stage 5: Improve . . . 12
Stage 6: Defend . . . 13
Stage 7: Career Translation . . . 14
Before you call this finished . . . 15
Optional: Governance as Code in Python (Deep Dive) . . . 16

Welcome to Project 5

Read this once before Stage 1.

The engagement

Use case	Harborline Member Assist
Application vendor	Pallet Desk, Pallet Labs, Inc.
Model provider	CedarLM, Cedar Foundry
Review freeze	2027-09-21 12:00 UTC (`2027-09-21T12:00:00Z`)
Pilot start / requested expansion	2027-06-08 / week of 2027-11-02
Your role	GRC / AI governance practitioner. You recommend. You are not counsel and not the risk owner.

Depth, tools, and outputs

Option	Time	What it includes
Standard	About 4 to 6 hours	Spreadsheet-first: full loop, Governance as Code tabs, recommendation, Break, brief, Career Translation. Can reach the top of the rubric.
Deep Dive	About 8 to 12 hours	Standard plus the same rules in Python, 3 to 5 configuration tests, and extra interview practice.

#	Artifact	Where it lives
1	AI Use-Case Map and Risk/Relevance Assessment	**1 Use-Case Map**
2	Member Assist Governance Plan	**2 Governance Plan**
3	Intake and Governance Requirements (Governance as Code)	Tabs **Use_Case_Intake** to **Change_Test**
4	Initial use-case recommendation	**3 Initial Recommendation**
5	Light framework mapping (after the plan)	**3 Framework Mapping**
6	Break reassessment	**4 Break Reassessment**
7	Revised governance	**5 Revised Governance**
8	Monitoring and reassessment plan	**5 Monitoring Plan**
9	AI Governance Decision Brief	**Deliverable_Templates/6_Defend_AI_Governance_Decision_Brief.docx**
10	Business pushback responses	**6 Business Pushback**
11	Career Translation (Standard set)	**7 Career Translation**

The chain: learn this first

Stage	Question
Purpose	Why is Harborline using AI here?
Data	What information enters the system?
Model	What model or AI service processes it, including third-party dependencies?
Output	What does the AI produce?
Decision	How does that output affect an action or decision?
Human	Where does meaningful human oversight exist?
Impact	Who or what is affected if the system is wrong, unavailable, biased, manipulated, or misused?

> **Do not begin with NIST AI RMF**
>
> Frameworks come after you understand the use and have designed governance. Generic illustration (not your answer): a tool drafts meeting summaries nobody uses to change access; a human can ignore them; impact is low.

Governance as Code, and its limits

Term	Meaning
Governance policy	What Harborline expects.
Governance logic	When a requirement applies.
Governance as Code	Selected logic in machine-evaluable form: spreadsheet formulas on Standard, optional Python on Deep Dive.
Governance decision	Human judgment about whether and how the use proceeds.

- **Two questions:** A. Does this requirement apply? (the workbook's Applies? column: Yes, No, or Review). B. Has Harborline demonstrated it is satisfied? (your current_status). A row that applies is not a row that is implemented.
- **Three-state inputs:** Yes, No, or Unknown. Empty means Unknown. Unknown does not silently equal No, and it does not automatically mean High or Pause.
- **Oversight by output:** evaluate summary, classification, priority, draft response, and external send separately. "A human reviews it" is not enough.
- **Impact classification** (Low, Moderate, High) is a Harborline scenario label for this exercise, not an industry standard. You choose and defend it. No formula outputs it.
- **Code can surface requirements. Humans still make governance decisions.** Rules never accept residual risk.

> **Four recommendation states (for the use, not the vendor)**
>
> Proceed, Proceed with conditions, Pause pending remediation or evidence, or Do not proceed with this use. Never "Reject Pallet." There is no official required state.

Stage 1: Assess

Reconstruct the actual use case.

What you are trying to accomplish

Map Member Assist through Purpose, Data, Model, Output, Decision, Human, and Impact, comparing what is claimed with what you can evidence.

Why this matters

AI governance starts with understanding the use, not with checking framework boxes. The pilot SOP describes one workflow; staff feedback and output samples may describe another.

Open these

What	Where to find it
The twelve packet files	**1_Assess_Evidence**
Your map and assessment	**Master_Workbook.xlsx**, tab **1 Use-Case Map**

File	What it is
1_Assess_Evidence/Business_Intake.pdf	Elena Voss's expansion request
1_Assess_Evidence/Workflow_Data_Flow.pdf	Engineering sketch of the workflow
1_Assess_Evidence/Pilot_SOP.pdf	Claimed operating procedure
1_Assess_Evidence/Vendor_Overview.pdf	Pallet Desk and CedarLM
1_Assess_Evidence/System_Card_Extract.pdf	Incomplete system card
1_Assess_Evidence/Tenant_Config_Note.pdf	Region, retention, training toggle
1_Assess_Evidence/Privacy_Data_Note.pdf	Privacy questions, not legal verdicts
1_Assess_Evidence/Pilot_Metrics.pdf	Adoption metrics and their limits
1_Assess_Evidence/Staff_Feedback.pdf	How staff actually work
1_Assess_Evidence/Output_Samples.pdf	Sample AI outputs
1_Assess_Evidence/Issue_Log.pdf	Pre-expansion issues
1_Assess_Evidence/AI_Principles_Draft.pdf	Harborline draft AI principles

Do this

1. Read all twelve files in **1_Assess_Evidence**. Start with **1_Assess_Evidence/Business_Intake.pdf** and **1_Assess_Evidence/Workflow_Data_Flow.pdf**.
2. On **1 Use-Case Map**, fill the chain table: for each stage, what is claimed, what you can actually evidence, and the source.
3. Compare the claimed boundary with actual use, and name the stakeholders.
4. In the observations table, label what you found (use-case fact, claimed but unverified, unknown, ordinary vendor or app issue, AI-specific risk, distraction, escalate to Privacy or Legal). Include at least one AI-specific and one ordinary vendor issue.
5. List material risks (relevance, not volume), unknowns, escalations, and at least one AI risk you will not treat as material, with why.
6. Choose an initial Harborline impact classification (Low, Moderate, or High) and write the rationale. You may challenge the scale.

What you produce	You are done when
• Completed **1 Use-Case Map**.	☐ The chain is complete, with claimed versus evidenced. ☐ At least one true negative or irrelevant buzzword is recorded. ☐ You have not started framework mapping or opened Break.

Next: Move to Stage 2: Design.

Stage 2: Design

Design governance for this use, not a generic AI policy.

What you are trying to accomplish

Write the Member Assist Governance Plan, with human oversight requirements set per output.

Why this matters

Proportionate governance depends on what this AI actually does. A summary someone can ignore and a classification that routes a member's message are different risks and need different oversight.

Open these

What	Where to find it
Governance plan	Tab **2 Governance Plan**
Your use-case map	**1 Use-Case Map**

Do this

1. On **2 Governance Plan**, complete owner, purpose, approved scope, prohibited scope, data boundary (with Privacy escalation points), and impact classification with rationale.
2. Write human-oversight requirements separately for summary, classification, priority, draft response, and external send or resulting action.
3. Complete model and vendor requirements (Pallet Desk, CedarLM, training use, version notice), testing, short monitoring requirements, incident and escalation, documentation, change management (use versus model versus vendor), reassessment triggers, and risk acceptance authority.
4. Do not map frameworks yet.

What you produce	You are done when
• Completed **2 Governance Plan**.	☐ Oversight is defined for all five outputs. ☐ Approved and prohibited scope are explicit. ☐ An acceptance authority is named.

Next: Move to Stage 3: Build (Analyze).

Stage 3: Build (Analyze)

Encode selected governance logic, then make a human recommendation.

What you are trying to accomplish

Use the spreadsheet Governance as Code tabs to see which requirements apply, record whether each is satisfied, prove the rules move when the use changes, then write your recommendation and a light framework map.

Why this matters

Transparent rules make governance repeatable: change a use-case characteristic and the applicable requirements change. But the workbook is not an approval engine. You still decide.

Open these

What	Where to find it
How the Governance as Code tabs work	Tab **3 Build Guide**
Intake, oversight, rules, results, status, change test	**Use_Case_Intake**, **Human_Oversight**, **Governance_Rules**, **Requirement_Results**, **Requirement_Status**, **Change_Test**
Initial recommendation	Tab **3 Initial Recommendation**
Framework mapping	Tab **3 Framework Mapping**
Optional Python	**Optional_Python_Governance_as_Code/README.md**

Do this

1. Read **3 Build Guide**.
2. On **Use_Case_Intake**, review each starting value (taken from the **claimed** pilot). Change any value your Assess evidence contradicts and note why. Leave Unknown where you lack evidence.
3. On **Human_Oversight**, rate the eight oversight dimensions for each of the five outputs.
4. Read **Governance_Rules**, then look at **Requirement_Results**. The Applies? column is formula-driven. Do not edit those formulas.
5. On **Requirement_Status**, record for every requirement whether Harborline has demonstrated it is satisfied, the evidence available, the action needed, owner, and timing.
6. On **Change_Test**, set after_what_if values (for example automated_action to Yes) and watch G-07, G-08, and G-09 move. Clear the what-if afterwards.
7. On **3 Initial Recommendation**, circle one recommendation state and complete scope, conditions, residual risk, owner, authority, monitoring, reassessment date or trigger, and why the workbook did not make this decision.

8 Only now, on **3 Framework Mapping**, map a few of your controls to Harborline's draft principles and the NIST AI RMF functions (Govern, Map, Measure, Manage). About one page. No giant crosswalk.

9 **Deep Dive (optional):** implement the same rules in Python (end of this guide).

What you produce	You are done when
• Updated intake, oversight, and status tabs; **3 Initial Recommendation**; **3 Framework Mapping**.	☐ Applies? moves when automated_action or classification review changes. ☐ Requirement_Status can show Applies = Yes and Not satisfied. ☐ Your recommendation is written, not a cell output. ☐ You have not opened Break.

Next: Only now open the Break evidence and move to Stage 4.

Stage 4: Break

The use changed. The technology did not.

What you are trying to accomplish

Update the use-case facts, watch the governance requirements move, and decide what changed in classification, recommendation, authority, monitoring, and evidence.

Why this matters

Governance can change because the authority given to the AI changed, even when the vendor, model, and data types stay the same.

Open these

What	Where to find it
Break evidence	**4_Break_Incident_Evidence_Open_After_Build/Operations_Update.pdf**
Your reassessment	Tab **4 Break Reassessment**
Intake and results	**Use_Case_Intake**, **Requirement_Results**, **Change_Test**

Open this evidence only after your initial recommendation

Do not open **4_Break_Incident_Evidence_Open_After_Build** until **3 Initial Recommendation** is complete.

Do this

1. Read **4_Break_Incident_Evidence_Open_After_Build/Operations_Update.pdf**. It gives facts, not conclusions.
2. Update **Use_Case_Intake** to match the facts: automated_action to Yes; human_review_of_classification to No for the auto-routed subset; reduction_in_human_oversight to Yes if that matches; classification_affects_routing to Yes. Note the reason in each notes cell.
3. Watch **Requirement_Results** and **Requirement_Status** move. Record which requirements changed.

4 On **4 Break Reassessment**, answer every prompt: facts used, what they do not say, what changed and did not change, which Applies? flags moved, and your judgments on impact class, recommendation, authority, monitoring, and evidence.

5 List what you still have to decide with human judgment. The workbook must not output Pause or an official class.

What you produce	You are done when
• Updated intake and a completed **4 Break Reassessment**.	☐ Additional requirements or Review flags appear after your intake update. ☐ Your judgments are written, not computed.

Next: Move to Stage 5: Improve.

Stage 5: Improve

Treat approval as a lifecycle, not a one-time event.

What you are trying to accomplish

Revise governance after the change and write one combined monitoring and reassessment plan.

Why this matters

Routine monitoring asks whether the approved use is operating within boundaries. **Full reassessment** asks whether something changed enough that the governance decision itself should be revisited. This Break is a worked example of new automated action plus reduced oversight as a reassessment trigger.

Open these

What	Where to find it
Revised governance	Tab **5 Revised Governance**
Monitoring and reassessment plan	Tab **5 Monitoring Plan**

Do this

1. On **5 Revised Governance**, answer the eight questions: what changed in the use, which requirement changed, what did not change, the weakest original assumption, and whether classification, recommendation, and authority change, plus new monitoring or evidence.
2. On **5 Monitoring Plan**, choose a proportionate set of routine metrics with owner, frequency, threshold or trigger (Harborline scenario values, not industry standards), and action.
3. Mark each full-reassessment trigger as routine watch or full revisit.

What you produce	You are done when
• Completed **5 Revised Governance** and **5 Monitoring Plan**.	☐ Routine monitoring and full reassessment are distinguished. ☐ No single metric is presented as proof of safety.

Next: Move to Stage 6: Defend.

Stage 6: Defend

Brief steering and answer the business.

What you are trying to accomplish

Write the AI Governance Decision Brief and concise answers to seven business pushback questions.

Why this matters

Write for Elena Voss and steering, not as a policy essay. Be ready to explain why rules that know which controls apply still cannot approve the use.

Open these

What	Where to find it
Decision Brief	**Deliverable_Templates/6_Defend_AI_Governance_Decision_Brief.docx**
Business pushback	Tab **6 Business Pushback**

Do this

1. Open **Deliverable_Templates/6_Defend_AI_Governance_Decision_Brief.docx**, save a copy, and complete every section in about one to two pages, from your **revised** position.
2. On **6 Business Pushback**, answer all seven questions in short paragraphs. For question 7, distinguish machine-evaluable requirements from business risk acceptance and contextual judgment.

What you produce	You are done when
• Your brief copy and **6 Business Pushback**.	☐ The brief names the recommendation state, residual risk, and risk owner and acceptance authority. ☐ Pushback answers are concise.

Next: Move to Stage 7: Career Translation.

Stage 7: Career Translation

Turn the assessment into an honest professional story.

What you are trying to accomplish

Write the Standard career set: honesty sentence, description, bullets, 30-second explanation, and STAR.

Why this matters

Interviewers will test "we have a human in the loop" and "why not auto-approve?" Your answers come from this project.

Open these

What	Where to find it
Prompts and writing areas	Tab **7 Career Translation**

Do this

1. Say the honesty sentence: *"This was an independent portfolio project where I simulated an AI governance assessment for a fictional company. I mapped an AI use case, designed proportionate controls, expressed selected requirement logic as transparent spreadsheet rules, reassessed when the use changed, and wrote a decision brief with residual risk and named acceptance authority."*
2. Complete the Standard set on **7 Career Translation**.
3. **Deep Dive (optional):** README paragraph, skills, interview questions, defense drill.

What you produce	You are done when
• The Standard career set on **7 Career Translation**.	☐ You can say the honesty sentence and STAR without reading. ☐ No claims of building an AI system or approving Pallet.

Next: Complete the final checklist.

Before you call this finished

A final check across all seven stages.

- ☐ Use-case map
- ☐ Governance plan
- ☐ Governance as Code tabs updated
- ☐ Initial recommendation
- ☐ Framework mapping (after the plan)
- ☐ Break reassessment
- ☐ Revised governance
- ☐ Monitoring and reassessment plan
- ☐ Decision Brief
- ☐ Business pushback
- ☐ Career Translation Standard set

Reasoning questions

- What did I mark Unknown rather than No, and why?
- During Break, did the vendor stay the same while the authority given to the AI changed?
- Did I update Applies? versus satisfied after the use changed?
- What would trigger a full reassessment versus routine monitoring?
- Can I defend this to a business owner who wants the workbook to "just approve it"?

Optional: Governance as Code in Python (Deep Dive)

Same rules as the spreadsheet. Not a Day 1 activity.

Translate selected spreadsheet rules into beginner-readable Python. Your program may output applicable requirements, Review conditions, monitoring flags, and reassessment triggers. It must not output Approve, Reject, Proceed, Pause, Do not proceed, an impact class, or residual-risk acceptance. `reassessment_required = true` is allowed; `decision = "Pause"` is not.

File (inside)	Purpose
use_case.json	Claimed-pilot starting point (same values as the starter intake)
cases/	Five tiny configurations A to E to watch requirements change
src/governance_rules.py	TODOs: tri-state and applies helpers
src/evaluate_governance.py	TODOs: load JSON, evaluate, write outputs/governance-requirements.json
expected_output.md	Output shape
validate.py	Shape and guardrail checks. Not an answer key.

Python setup (one time)

1. Install Python 3.10 or newer from https://www.python.org/downloads/. On Windows, tick **Add Python to PATH** during setup.
2. Open a terminal. On Windows use PowerShell or Command Prompt. On macOS use Terminal.
3. Check the version: `python3 --version` (on Windows the command is often `python --version`). You want 3.10 or higher.
4. Change into the optional Python folder inside this project. The easiest way: type `cd` followed by a space, then drag the **Optional_Python_Governance_as_Code** folder onto the terminal window and press Enter.
5. Nothing needs to be installed. The exercises use the Python standard library only.

Run commands

```
python3 src/evaluate_governance.py
python3 src/evaluate_governance.py --case cases/case_e_automated_action.json
python3 validate.py
```

On Windows, type `python` wherever these commands say `python3`. Scripts find the data folder on their own, so you can run them from any location as long as the project folder structure is unchanged. Output is written to the **Optional_Python_Governance_as_Code/outputs** folder, which the scripts create if it is missing.

What you see	What it means
`NotImplementedError`	Expected. You still have a TODO in that function. Read the HINT lines next to it.

What you see	What it means
`python3: command not found`	Try `python`. If neither works, install Python and reopen the terminal.
`FileNotFoundError ... outputs/0N_...json`	Run the earlier numbered step first. Each step reads the previous step's output.
Wrong SLA or freshness results	Use the review freeze timestamp from this guide, never your computer clock.

Done when Cases B and E (or your Change_Test equivalent) show additional automated-action and oversight requirements, and the JSON has no decision keys.