PRACTICE REPORT - PUBLIC EDITION - JULY 2026

Production Confidence Review

Agentic Fitness Coach

A web application combines personalised training plans, completed sessions, health information, and meals in one product. An AI agent uses this information to assess progress and adapt later training and nutrition recommendations.

The application was initially developed as a single-user prototype and is now used daily by selected test users. Expanding it into a multi-user application increased the requirements for data separation, operational reliability, and traceable agent behaviour.

Can the application be approved for a controlled, invitation-only pilot with its first paying users?

REPORT NAVIGATION

Contents

> The report separates the management decision, prioritised measures, and technical evidence.
> The assessment applies exclusively to the pilot operation described here.

SECTION

Decision at a Glance

Ready with Conditions

Decision scope: Controlled, invitation-only pilot operation with initial paying users

The application can be approved for a limited, supported pilot.

The primary user journey is usable, material boundaries between users are technically established, and the most important known agent failure is now checked reproducibly before releases.

A public self-service rollout or larger-scale operation is not yet approved.

APPROVED NOW

- Invitation-only access for a clearly limited pilot group
- Use of the reviewed primary user journey
- Operation with a defined application and agent version
- Personal support and direct response to reported problems
- Manual checks of critical operational workflows during the pilot

NOT YET APPROVED

- Public registration without access control
- Unsupported rollout to a larger user group
- Frequent releases without mandatory technical checks
- Untested changes to prompts, models, or agent logic
- Scaling without cost measurement per user and workflow
- Operation without active monitoring of critical background processes

SECTION

Why This Decision Is Defensible

1. The primary user journey works

Users can:

1. provide goals and relevant context
2. receive personalised training and nutrition recommendations
3. record workouts, meals, and health information
4. have their progress assessed
5. have later recommendations adapted based on their history

This journey exists and is used regularly in the current test operation.

2. The most important known agent risk has been made more controllable

During testing, the agent interpreted a medium-term goal inadequately. The resulting training plan was technically valid and plausibly written, but did not develop load and progression appropriately over the planned period.

This failure case and other representative scenarios were added to a repeatable evaluation process.

Critical misinterpretations can therefore be detected before a new agent version is released.

3. The remaining risks are manageable within the limited pilot

Monitoring, release safeguards, isolation tests, and cost control are not yet complete. Within a small, known pilot group, these risks can be temporarily limited through:

- controlled user provisioning
- fixed software and agent versions
- direct support
- manual operational checks
- clearly assigned responsibilities
- limited change frequency

These transitional controls are not sufficient for a broad rollout. They are defensible for a limited pilot.

SECTION

What Needs Attention Now

1. Actively Detect Critical Background Failures

PRIORITY: HIGH

An enabled background process failed repeatedly during testing. The failures were technically discoverable but did not trigger an active notification.

Business impact

An important user workflow can remain disrupted even while the application still appears available. Problems may only be identified through user complaints.

Required before

Expanding the pilot group

Next step

- detect repeated failures automatically
- alert after a defined threshold
- identify the affected user and workflow with minimal data
- test alerting with a controlled failure
- document response and recovery

---

2. Make Release Safeguards Mandatory

PRIORITY: HIGH

Changes to the main branch are deployed automatically. There is no separate production-like staging environment and no complete mandatory release gate.

Business impact

Failures may first become visible in the live pilot system. This concerns both conventional software defects and changes caused by:

- new prompts
- model changes
- modified agent tools
- new data formats
- adapted integrations

Required before

Regular product changes during the pilot

Next step

Before every production deployment, run at least:

- build
- static checks
- automated tests
- agent evaluations
- isolation tests
- smoke test of the primary user journey

---

3. Establish Broader Evidence for User and Data Separation

`PRIORITY: HIGH`

User-bound access, limited agent permissions, and targeted isolation tests are present.

The review found no actual disclosure of another user's data. However, not all critical access paths are tested automatically with multiple users and failure cases.

Business impact

The likelihood appears limited, but the potential impact of a failure would still be high.

Required before

Public registration

Next step

Add automated tests for at least:

- access to another user's training data
- access to another user's meals or health information
- agent execution in the wrong user context
- incorrect user assignment in background processes
- manipulated or foreign resource identifiers
- direct access across API and database boundaries

---

4. Measure Cost per User and Workflow

`PRIORITY: MEDIUM`

Total costs so far are within the expected range. Reliable attribution to individual users, agent runs, or product workflows is missing.

Business impact

Cost increases may only become visible on the total invoice. Pricing and scalability therefore remain difficult to assess.

Required before

Pricing validation and scaling

Next step

Measure at least:

- model cost per user
- model cost per primary workflow
- number and duration of agent runs
- failed and repeated executions
- infrastructure cost per active user
- cost development as usage grows

---

5. Test Recovery Reproducibly

`PRIORITY: MEDIUM`

Backups and rollback options are generally available. Full recovery was not demonstrated reproducibly within the scope of this review.

Business impact

A backup is only dependable once restoration, duration, and completeness have actually been verified.

Required before

Broad commercial rollout

Next step

- document recovery steps
- test restoration under controlled conditions
- define maximum data loss
- define a target recovery time
- assign the responsible role
- verify functionality after restoration

SECTION

The One Improvement Implemented Now

Agent Behavior Release Gate

The most important bounded product risk was not a conventional software defect.

The greater risk was that the agent could produce technically valid and plausibly written results that nevertheless failed the user's domain objective.

A reproducible test process for key agent scenarios was therefore implemented.

Why This Measure Was Selected

A medium-term objective had already been interpreted incorrectly. Before the release gate was implemented, this problem could only be found through manual inspection or user feedback.

The new process answers at least these questions before a release:

- Does the agent understand short- and medium-term goals correctly?
- Does a training plan develop coherently over time?
- Are load limits and relevant health information taken into account?
- Does the agent recognise unclear or contradictory input?
- Does a known failure reappear after a change?

Implemented Scope

- representative scenarios for short- and medium-term training objectives
- regression test for the observed misinterpretation
- scenarios with unclear or contradictory input
- defined minimum requirements for progression and load management
- criteria for critical and non-critical deviations
- repeatable execution against relevant agent versions
- documented process for adding new scenarios
- integration into the release process

Done When

The implementation package is complete when the known failure case is detected reliably and:

- a deliberately faulty agent version fails the check
- the approved version passes every defined critical scenario
- the results are documented transparently
- the product or development team can run the check again independently

Handed Over

- tested code change
- evaluation scenarios
- assessment criteria
- automated regression test
- execution instructions
- instructions for adding new failure cases
- overview of risks not yet covered

Before

Changes to the model, prompt, or agent logic could reintroduce a known failure without it being detected reliably before production use.

After

Known critical misinterpretations are tested reproducibly. An agent version that does not meet the defined minimum requirements can be stopped before release.

Not Solved by This Measure

The release gate does not prove that every possible agent output is correct. In particular, it does not replace:

- expert review of every individual recommendation
- medical validation
- monitoring of ongoing operations
- cost control
- complete data-isolation tests
- general end-to-end tests
- detection of entirely unknown classes of failure

SECTION

Next Release Stage

A broader rollout should only be considered once:

- critical background failures generate active notifications
- the primary user journey is checked before every release
- known critical agent failures are detected reproducibly
- critical user boundaries are tested fully and automatically
- cost per user and workflow is measurable
- warning or limiting mechanisms for cost are in place
- faulty releases can be stopped or rolled back
- recovery and incident response are documented and tested

SECTION

Technical Evidence

The following part documents the basis for the decision. It is intended primarily for technical owners who need to understand or continue working on the findings, evidence, and remaining risks.

SECTION

1. Review Scope

Assessed Use

The review assessed a controlled, invitation-only pilot operation with initial paying users. The verdict applies exclusively to:

- a limited and known user group
- the primary user journey described here
- the reviewed application and agent version
- a supported operation
- a direct ability to respond to problems

Primary User Journey

The review focused on this sequence:

1. The user provides goals and relevant context.
2. The application creates a training or nutrition plan.
3. The user records completed sessions, meals, and health information.
4. The agent assesses the user's progress.
5. Later recommendations are adapted using the recorded information.

Not every product function was reviewed in full. Priority went to components whose failure or incorrect behaviour would directly endanger the pilot.

Review Areas

- product behaviour and primary user journey
- architecture and external integrations
- user and data separation
- agent behaviour and domain failure cases
- testing and release safeguards
- deployment and recovery
- monitoring and failure detection
- cost and scalability
- maintainability and technical handover

Outside the Scope

The review was explicitly not a complete penetration test, medical efficacy assessment, regulatory audit, complete load or scalability test, review of every product function, general production-readiness certification, data-protection impact assessment, or substitute for legal or medical advice.

SECTION

2. Method and Assessment Basis

Evidence Used

EVIDENCE TYPE	MEANING
Measured	Confirmed through automated checks, logs, or reproducible measurements
Demonstrated	Observed in the product or a controlled execution
Documented	Supported by source code, configuration, or technical documentation
Reported	Described through experience from the test operation to date
Unknown	Not established reliably with the available access

Statements based on documented or reported evidence were weighted less strongly than reproducible measurements.

Assessment Scale

RATING	MATURITY	MEANING
0	Uncontrolled	Material controls are missing or a high risk is not limited
1	Fragile	The normal case works, but important safeguards are missing
2	Controlled	Defensible for the defined use within known boundaries
3	Proven	Tested, observable, documented, and clearly owned

The ratings are not averaged. A single credible blocker can outweigh strong results elsewhere. The final verdict is therefore based on risk, impact, and available evidence rather than a mathematical overall score.

Checks Performed

The review included in particular:

- static review of relevant application components
- execution of existing automated tests
- review of selected user and agent permissions
- walkthrough of the primary user journey
- analysis of the deployment process
- assessment of known failures from the test operation
- review of existing monitoring and diagnostic capabilities
- assessment of available cost information
- selection and implementation of one bounded risk reduction

The existing automated checks passed at the time of the review. They did not yet cover the complete user journey or every critical agent scenario.

SECTION

3. Readiness Scorecard

REVIEW AREA	RATING	SUMMARY
Product and user journey	2	The primary journey exists and is used regularly
Architecture and integration	2	System boundaries and key responsibilities are identifiable
User and data separation	2*	Material controls exist, but the evidence is not yet complete
Agent behaviour	2	The known failure and prioritised scenarios are tested reproducibly
Testing and release process	1-2	Targeted tests exist, but a complete mandatory gate is missing
Deployment and recovery	1	Deployment is automated, but there is no production-like pre-release stage
Monitoring	0-1	Failures can be found but do not reliably trigger active notification
Cost control	1	Total cost is plausible, but not measurable per user and workflow
Maintainability and handover	1-2	The application is documented but retains manual operational steps

> * Controlled with conditions. The review found no actual disclosure of another user's data. The possible impact of a failure nevertheless justifies additional automated evidence.

SECTION

4. Detailed Findings

Finding 1: Agent Behaviour Was Not Reproducible Enough in Domain Terms

PRIORITY: CRITICAL EVIDENCE: REPORTED AND REPRODUCED STATUS: REDUCED BY BOUNDED IMPLEMENTATION

A medium-term objective was interpreted inadequately by the agent. The resulting training plan was technically valid, but did not develop load and progression appropriately over the planned period.

Impact

A result can appear formally correct while missing the domain objective. Users may find such failures difficult to detect because the output can be plausibly written.

Cause

There was no reproducible set of representative scenarios testing core domain expectations. Before changes to prompts, models, or agent logic, it was not possible to determine reliably whether known failures had reappeared.

Implemented measure

The known failure and additional prioritised scenarios were added to the Agent Behavior Release Gate.

Remaining risk

The test set covers known and prioritised failure classes, not every possible input or recommendation.

Finding 2: Repeated Background Failures Did Not Trigger Active Alerts

PRIORITY: HIGH EVIDENCE: DEMONSTRATED STATUS: OPEN

An enabled background process failed repeatedly. The failures were traceable in technical logs but did not trigger active notification.

Impact

A user-relevant workflow can remain disrupted for an extended period even while the application itself remains available.

This creates the risk that:

- problems are only noticed after user complaints
- data or recommendations are processed late
- causes become harder to reconstruct later
- operations appear stable while subprocesses have failed

Required measure

- detect repeated failures automatically
- alert after a defined threshold
- identify the affected workflow with minimal data
- document response and recovery
- verify alerting under controlled conditions

Assessment

Required before materially expanding the user group.

Finding 3: User and Data Separation Has a Strong Foundation but Incomplete Evidence

`PRIORITY: HIGH` `EVIDENCE: DOCUMENTED AND PARTLY MEASURED` `STATUS: PARTLY CONTROLLED`

The application uses user-bound access, limited agent permissions, and targeted isolation tests.

No actual disclosure of another user's data was found.

Remaining gap

Not all critical access paths in the primary user journey are tested automatically across multiple users, roles, and failure cases.

Impact

A failure in this area would have a high potential impact despite its limited apparent likelihood.

Required measure

Automated isolation tests for:

- another user's training data
- another user's meals and health information
- agent execution in the wrong user context
- background processing with incorrect user assignment
- manipulated identifiers
- direct API and database access

Assessment

Defensible for the limited pilot. Evidence must be extended before public registration.

Finding 4: Changes Reach Production Without an Adequate Pre-Release Stage

PRIORITY: HIGH EVIDENCE: DEMONSTRATED STATUS: OPEN

Changes to the main branch are deployed automatically. There is no separate production-like staging environment and no complete mandatory release gate.

Impact

Failures may first become visible in the live pilot system. This concerns:

- conventional software defects
- new prompts
- model changes
- adapted agent tools
- new data formats
- changed external integrations

Required measure

Before production deployment, require at least:

- build
- static code checks
- automated tests
- the primary end-to-end user journey
- critical isolation tests
- agent evaluations
- a controlled smoke test

Assessment

Temporarily defensible for infrequent, tightly controlled changes during the pilot. Not sufficient for frequent releases or a larger user group.

Finding 5: Cost Is Plausible but Not Demonstrably Controllable

PRIORITY: MEDIUM EVIDENCE: REPORTED STATUS: OPEN

Total costs so far are within the expected range. Reliable attribution to individual users, agent runs, or product workflows does not exist.

Impact

Cost increases may only become visible on the total invoice. It remains unclear:

- which users create particularly high cost
- which agent workflows are expensive
- whether failures create unnecessary model calls
- what cost will arise as usage grows
- whether the planned pricing remains economically viable

Required measure

Measure at least:

- model cost per user
- model cost per workflow
- number and duration of agent runs
- failed or repeated executions
- infrastructure cost per active user
- cost development as usage increases

Assessment

No immediate blocker for a small pilot group. Required before pricing validation and scaling.

Finding 6: Recovery Is Possible but Not Operationally Proven

PRIORITY: MEDIUM EVIDENCE: REPORTED AND DOCUMENTED STATUS: OPEN

Backups and rollback options are generally available. Recovery was not fully demonstrated reproducibly within the available review scope.

Impact

A backup is dependable only once it is clear what data it contains and how long recovery takes.

- who can perform the recovery
- what data may be lost
- how functionality is verified afterwards

Required measure

- document recovery steps
- test restoration under controlled conditions
- define target values for data loss and recovery duration
- assign the responsible role
- run a defined smoke test after restoration

Finding 7: Maintainability Is Sufficient for the Pilot but Person-Dependent

`PRIORITY: MEDIUM` `EVIDENCE: DOCUMENTED` `STATUS: PARTLY CONTROLLED`

The application has a traceable structure and technical documentation. Some operational and diagnostic procedures still depend heavily on individual knowledge and manual steps.

Impact

If the main responsible person is unavailable, diagnosis and recovery may take longer.

Required measure

Short runbooks for:

- failed background processing
- faulty deployment
- model or provider outage
- unexpected cost increase
- recovery from backup
- suspending or removing a user

SECTION

5. Technical Evidence of Implementation

A complete client edition should reference the following evidence specifically.

Code Change

- repository or handover location
- pull request or commit
- affected components
- short technical description
- review status

Automated Evidence

- number of included scenarios
- agent version under test
- expected and actual results
- result of the known regression test
- result of the deliberately faulty variant

Release Integration

- time of execution
- triggering branch or workflow
- behaviour on critical failure
- documented exception or approval rules

Handover

- responsible role
- instructions for rerunning the checks
- instructions for adding new failure cases
- known limitations
- recommended next extension

A public edition may abstract sensitive repository and client details. It should still contain concrete before-and-after evidence.

SECTION

6. Remaining Risks and Transitional Controls

Background Failures Are Not Yet Alerted Reliably

Transitional control

- regular manual review of critical processes
- direct support for pilot users
- clear escalation route for problems

Not Every Agent Failure Is Covered by Evaluations

Transitional control

- limited pilot group
- fixed agent version
- manual review of unusual results
- direct escalation path

User Separation Is Not Fully Demonstrated Automatically

Transitional control

- no public registration
- controlled user provisioning
- limited roles and agent permissions

Faulty Releases Can Still Reach the Pilot

Transitional control

- small releases
- clearly assigned release responsibility
- documented rollback option
- limited change frequency

Cost Has No Warning or Limiting Mechanism Yet

Transitional control

- limited user count
- regular manual review of total cost
- no automatic scaling without approval

SECTION

7. Roadmap by Decision Gate

Before the Paid Pilot Begins

- complete and verify the Agent Behavior Release Gate
- define the reviewed agent and application version
- clearly limit the pilot group and access
- name the responsible person for incidents
- document the manual control and support process

Before Expanding the Pilot Group

- introduce monitoring for key background processes
- test active alerting with a controlled failure
- make tests and build mandatory before releases
- measure cost per user and primary workflow
- document an incident and recovery runbook

Before Public Registration

- test critical user and data boundaries comprehensively and automatically
- introduce a production-like staging environment
- run the complete primary user journey as a release gate
- stop faulty releases automatically or roll them back reliably
- define roles, responsibilities, and escalation routes

Before Broad Commercial Rollout

- define cost limits and warning thresholds
- evaluate operational data from the pilot
- extend agent evaluations with real pilot failures
- test recovery under controlled conditions
- review load and scaling assumptions
- make a new formal go-live decision

SECTION

8. Supplementary 30/60/90-Day Plan

The timeline is indicative only. The applicable release conditions remain decisive.

Within 30 Days

- introduce monitoring and alerting for critical background processes
- run build, tests, and agent evaluations as a release gate
- measure cost per user and primary workflow
- assign incident and recovery responsibility

Within 60 Days

- introduce a production-like staging environment
- broaden automated checks of critical user access
- extend agent evaluations with failure cases from the pilot
- test recovery under controlled conditions

Within 90 Days

- define warning thresholds and cost limits
- complete deployment and recovery runbooks
- review scaling and load assumptions
- make a new go-live decision based on real operational data

SECTION

9. Criteria for the Next Release Decision

A broader rollout should only be approved when:

- critical failures generate active notifications in time
- the primary user journey is checked before every release
- known critical agent failures are detected reproducibly
- changes to agent logic and models are approved under control
- cost per user and workflow is measurable
- warning or limiting mechanisms for cost are in place
- critical user boundaries are tested automatically
- a faulty release can be stopped or rolled back
- recovery and incident response are documented and tested

SECTION

10. Report Limitations and Validity

This case analysis documents the technical assessment of an application in a controlled pilot phase.

It is:

- not a client reference report
- not a complete security or penetration test
- not a medical assessment
- not a regulatory certification
- not a guarantee of error-free operation
- not a general approval for every user count or use case

The verdict applies exclusively to:

- the primary user journey described here
- the reviewed application version
- the reviewed agent version
- the controlled user group
- the stated operating conditions

Some statements are based on information from operations to date. This particularly includes daily usage, current cost, existing backups, and previous rollbacks.

Where reproducible evidence was unavailable, this was considered in the assessment.

Changes to architecture, agent logic, model, data access, deployment, or usage scenario may require a new assessment.