Tucker Payment Processing · Standards self-assessment

ISO/IEC 42005:2025 AI System Impact Assessment

An impact assessment of Tucker and the AI SOC Squirrel, following the standard's structure · Self-assessment · November 24, 2026

8 met **2 partly met**

About this document

This is written by Michael Cunningham, who designed and built the site. ISO/IEC 42005:2025, published in May 2025, is the companion to ISO/IEC 42001. It gives guidance on assessing how an AI system and its foreseeable uses affect individuals, groups and society, and on recording that assessment so it can be reviewed and kept current. ISO 42001 requires organizations to assess AI impacts; ISO 42005 explains how.

Unlike the other write-ups, this one is the assessment itself, set out in the elements the standard asks an assessment to cover. Each element shows whether the site's documentation covers it and where the evidence lives.

The assessment

Scope and timing — MET

The assessment covers the two AI systems on tuckerpaymentprocessing.com: Tucker, a customer-support agent, and the AI SOC Squirrel, a security analyst. It was written at launch and is reviewed whenever an AI's tools, data or purpose change. Any such change goes through GitHub, where this document lives beside the code.

Evidence: `README.md`, `docs/standards/iso-42005.md`.

System description and intended use — MET

Tucker answers questions about a fictional payment company's policies, and about a verified demo customer's own cards, transactions and orders. He is built on Google Gemini, with ten read-only lookup tools and retrieval over four policy documents. The AI SOC Squirrel reviews one attack run's events and writes an incident report. Neither system makes or recommends a decision about a person.

Evidence: `app/chat.py`, `app/tools.py`, `app/cipher/soc.py`, `policies/ai-assistant-policy.md`.

Foreseeable misuse — MET

The assessment assumes people will try to misuse the AI: prompt injection, jailbreaks, extracting data about other customers, getting refunds issued, running up costs and planting instructions for the security AI. Each misuse is answered by a control in code and is attacked live by Cipher on demand.

Evidence: `app/cipher/attacks.py`, `docs/standards/owasp-llm-top-10.md`.

Data — MET

All customer data is fictional and reset nightly. Cards are stored as the last four digits and a token only. Visitors' chat text is held in server memory for at most 30 minutes, is not stored with IP addresses and is never used to train a model. Web logs, which hold IP addresses, are kept for at most two days. Visitor IP addresses are never sent to the AI.

Evidence: `db/schema.sql`, `db/nightly_reset.sql`, `app/session.py`, `policies/data-retention.md`, `deploy/logrotate-apache2`.

Interested parties — MET

Those affected are: site visitors (privacy, honesty about the AI, not being misled); the demo customers whose fictional records Tucker reads (only their verified selves may see them); the owner (cost, reputation, security); the model supplier (Google); and, in a real deployment, cardholders, card networks and regulators.

Evidence: `site/about.html`, `policies/`.

Benefits — MET

For visitors: quick, accurate answers grounded in the company's actual policies, available at any hour, with a clear view of how each answer was produced. For the business: lower support load and a security analyst that summarizes attacks in plain English as they happen.

Evidence: `app/policies.py`, `site/assets/chat.js`.

Potential harms to individuals — MET

Assessed harms and their treatment. Privacy: data shown only after identity verification and only that person's own, minimized, with sensitive-looking output masked. Financial harm: no tool can move money or change an account. Deception: Tucker discloses he is an AI in every chat. Wrong answers: answers come from tools and policy documents, but this is reduced rather than eliminated (see the residual risk below).

Evidence: `app/tools.py`, `app/session.py`, `app/guardrails.py`, `app/chat.py`.

Potential harms to groups and society — MET

Neither AI decides anything about anyone (credit, eligibility, pricing, fraud), so there is no route to unfair treatment of a group. The site collects no protected characteristics. Tucker declines off-topic requests, which limits use of a company AI for unrelated or harmful content.

Evidence: `app/chat.py`, `db/schema.sql`.

Residual risk and human oversight — PARTLY MET

Two risks remain after controls. A fooled model could still give a wrong or misleading answer, and there is no human escalation path in the chat. In this demonstration both are accepted because no real money or customer is involved, and every conversation's guardrail events reach the owner's nightly report. A real deployment would add a visible route to a person and periodic review of a sample of conversations.

Evidence: `app/report.py`, `policies/ai-assistant-policy.md`.

Approval, monitoring and review — PARTLY MET

The owner approved the assessment, monitors outcomes through the nightly report and real-time alerts, and updates it with every change to the AI. What a real organization would add: sign-off by someone independent of the builder, and a fixed review cycle such as annually or on each model change.

Evidence: `app/report.py`, `app/alerts.py`.

Source: https://github.com/mcunningham-infosec/tucker-payment-processing/blob/main/docs/standards/iso-42005.md · Live site: https://tuckerpaymentprocessing.com
Written by Michael Cunningham with AI assistance (Claude). A self-assessment, not an audit, certification or legal opinion. Tucker Payment Processing is a fictional company and processes no real payments.