SENTINELFORESIGHT

WHITEPAPERS

Proving change control in Oracle Fusion

Why sampling from the change management tool no longer convinces auditors, what evidence starts from the system itself, and how to show that every change made in Oracle Fusion was authorised.

IN BRIEF

- Change management is one of the IT controls auditors examine most closely in a financial system such as Oracle Fusion.
- The traditional test samples changes from the change management tool. That shows recorded changes were handled well. It cannot show that every change was recorded.
- In our experience, auditors and regulators increasingly want the evidence to start from the system itself: what actually changed, and an authorisation for each change.
- Oracle Fusion records its changes, but across several audit logs and at a level of detail no one can review by hand.
- Sentinel groups those events into change episodes, matches each episode to its change request, and records a decision on everything that does not match.
- How change requests are written makes a measurable difference. A few simple habits raise the match rate considerably.

Why change control is audited

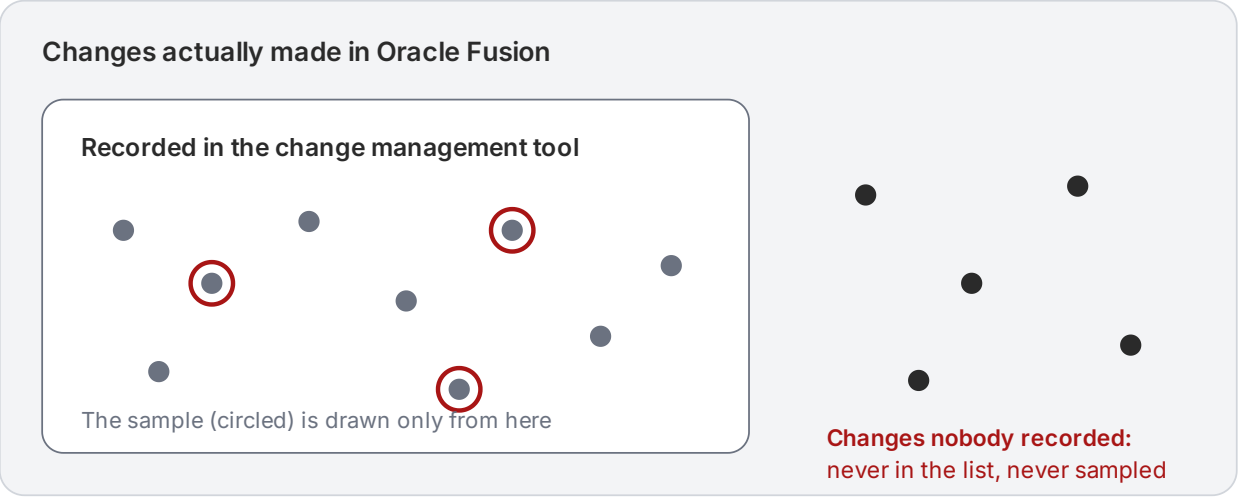
Listed companies and other large organisations must show that their financial reporting can be relied on. In the United States, the Sarbanes-Oxley Act requires management to assess internal control over financial reporting, and for many companies the external auditor must attest to it. There is no single European equivalent. Statutory auditors work to International Standards on Auditing, which expect them to understand the IT controls behind financial reporting. Several countries add their own regimes: the UK Corporate Governance Code asks boards to declare that their material controls are effective, for periods from 2026; France has the Loi de sécurité financière; Japan has J-SOX; and India requires internal financial controls under the Companies Act.

For an ERP such as Oracle Fusion, these regimes lead to the same place: the IT general controls. Change management is one of the most closely examined of these. The question it answers is simple. Was every change made to the system appropriate and authorised, so that none of them puts the accuracy of the financial statements at risk?

The gap in the traditional approach

For years, the usual test has started from the change management tool. The auditor takes the list of change requests for the period, picks a sample and checks that each sampled change was requested, approved, tested and implemented as described.

That test is sound as far as it goes. It shows that the changes recorded in the tool were handled properly. But it starts from the record of what people *meant* to change, not from the system that was changed. A change made without a request never appears in the list, so it can never be sampled.



A sample from the change management tool can only contain changes someone recorded.

The test therefore rests on an assumption: that the list of change requests is complete. If someone changed a role, a report or a setup value directly, without a request, the test will not find it. Worse, the test result can look clean precisely because the change was never recorded.

What auditors increasingly expect

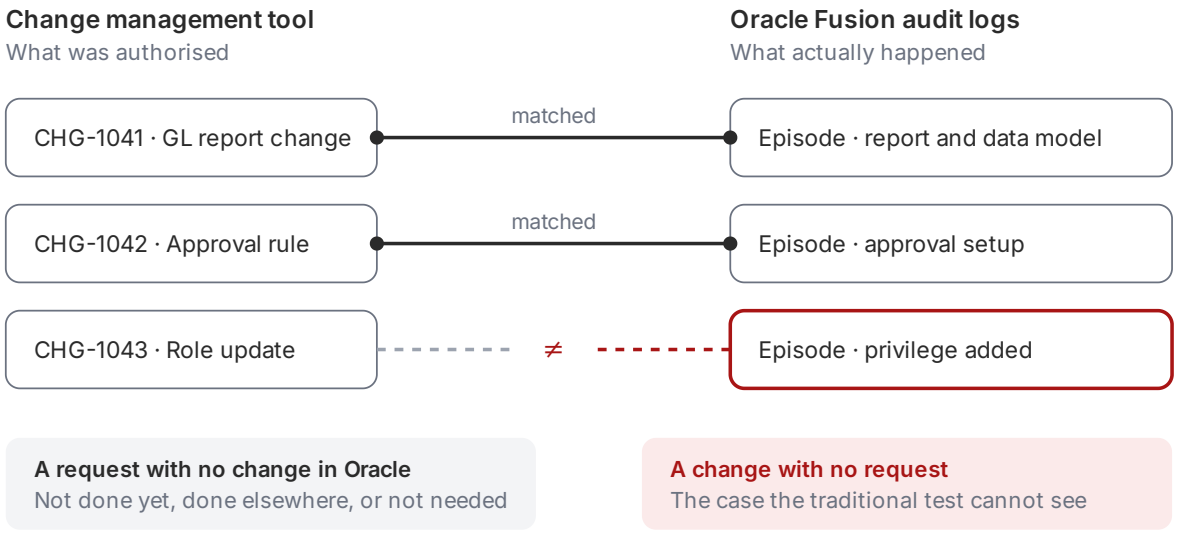
In our experience, auditors and regulators are less and less willing to accept that assumption. They increasingly ask for the evidence to start from the system under control:

1. **What actually changed**, taken from the system itself, for the whole period.
2. **An authorisation for each change**, so the company can show that every change on that list was requested and approved.

This reverses the direction of the test. Instead of asking whether the recorded changes were handled correctly, it asks whether every change that happened was authorised. The first question can be answered from the change management tool alone. The second cannot.

Seeing both sides of the picture

Answering the second question needs both sides at once: the system's own record of what changed, and the change management tool's record of what was authorised. Each side then checks the other.



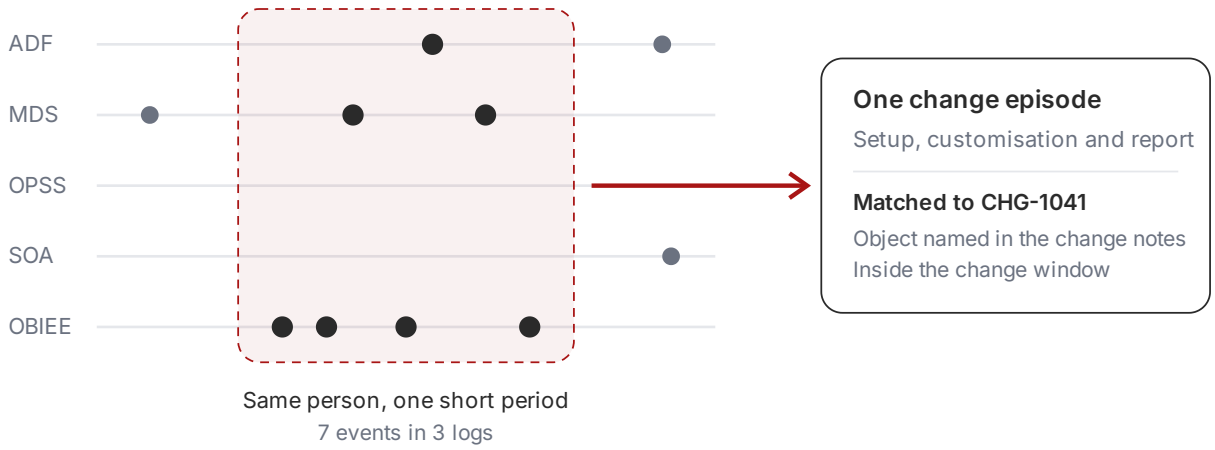
Matching both ways shows authorised changes, requests with no activity, and changes with no request.

This is the gap Sentinel Foresight was built to bridge. Oracle Fusion is the first system it supports.

From log events to change episodes

Oracle Fusion records changes in its own audit logs. By default Sentinel reads five of them, covering application setup and data (ADF), customisations and metadata (MDS), security (OPSS), integrations and workflow (SOA) and reporting (OBIEE). Together they are the system's own account of what changed, by whom and when.

They are not easy to read. A single change can produce many events, and often in more than one log. Changing a report might delete and re-save the report, update its data model and touch a prompt. Adding a privilege to a role writes security events and may change setup as well. Read event by event, one piece of work looks like a dozen unrelated changes.



Related events by the same person become one episode, which is matched to the change request behind it.

Sentinel groups related events by the same person into a **change episode**: one unit of work, such as “updated this report and its data model”. Episodes are scored for risk, so the changes that matter most come first. Then each episode is matched to the change request, and where there is one the deploy task, that best explains it.

Every match comes with its reasons. Sentinel looks for the changed objects named in the change request, for a matching external reference, for the change window (actual times where recorded, planned times otherwise) and for whether the deploy task was carried out by the person who made the change. A reviewer sees why each match was suggested and confirms, corrects or rejects it.

The matching also runs the other way. Starting from each change request, Sentinel shows the Oracle activity linked to it, so a request with no corresponding change stands out as clearly as a change with no request.

Change records that match well

How change requests are written makes a measurable difference. We have seen organisations raise their match rate considerably by applying a few consistent habits:

Practice	Why it matters
Name the changed object exactly: the report path, role name or setup task, not “fix to GL report” or “security update”.	Object names in the change notes are one of the strongest signals for a match. Generic wording matches nothing.
Use the change reference when deploying a sandbox: give the sandbox the change number or external reference.	The audit trail cannot see inside a sandbox. The name is what links the published sandbox to its request.
Use change categories consistently, one category for one kind of change.	If one category covers several kinds of change, events that belong to it may not be recognised as relevant, and will not be matched.
Record the actual start and end of the work, not only the planned window.	Actual times narrow the match. Planned times are used when actual ones are missing.
Record who deployed the change in the deploy task.	When the deploy task names the person who made the change, the match is stronger.
Close the change request when the work is done.	A change request that is not completed does not authorise the change.

None of this needs new tools or extra approvals. It is a matter of writing the change record so that someone, or something, can tie it to the system later.

What happens to what does not match

Some episodes will not match a change request. Each one is reviewed and given a recorded decision: the suggested match is approved, another change request is chosen, the episode is marked as having no valid change, or it is sent for further review. Every decision records the reviewer, the time and the reason.

From the other side, a change request with no activity in Oracle can be approved as needing none, with a comment explaining why: for example, a change made in another system, or one that was cancelled.

Where an episode points to something more serious, such as a change nobody can explain, it can be opened as a case and worked through to a reviewed conclusion. That record is the evidence that exceptions were found and dealt with, not just found.

What this gives you

- **A complete population.** Every change Oracle recorded in the period, grouped into episodes that people can understand.
- **An authorisation for each change**, or a recorded decision explaining why there is none.
- **Both directions covered**: changes without a request, and requests without a change.
- **A trail of review decisions**, with names, dates and reasons.
- **A measure of improvement.** As change records become more consistent, more episodes match without manual effort, and the improvement can be shown period by period.

The result is evidence that starts where auditors increasingly want it to start: with the system itself.

This paper describes general practice and our own experience. It is not legal or audit advice; your auditors will confirm what they expect for your organisation.