



Setting Up Continuous Auditing With Process Mining for a Mid-Sized Manufacturer's Procure-to-Pay Cycle: A Methodological Guide

Jack Wojcik^{1*}, Camille Cote², Aiko Osborne³

¹Department of Accounting, Sharavati Research Institute of Accounting, Finland

²Department of Accounting, Ural Institute for Advanced Accounting, Jordan

³Department of Accounting, Pearl River Academy of Accounting, Armenia

Article Info

Volume 1, Issue 1, 2026

doi: [10.62715/ijara.2026.v1.i1.pp14-18](https://doi.org/10.62715/ijara.2026.v1.i1.pp14-18)

Received: 24 January 2026

Accepted: 09 April 2026

Published: 08 June 2026

Abstract

Continuous auditing has been discussed for more than thirty years and is still rare outside large groups. Process mining, which reconstructs how transactions actually moved through a system from the traces they leave, has made it more attainable, because the same event log can show the real process, test it against the intended one and be refreshed on a schedule. This article sets out a method for establishing continuous auditing of the procure-to-pay cycle in a mid-sized manufacturing company using process mining. It is a guide to the method and reports no implementation. Eight stages are described: fixing scope and control objectives, building the event log, checking data quality, discovering the process as it runs, defining the reference model and rules, conformance checking, scheduling and alerting, and follow-up with feedback into the rules. For each control objective the article gives the deviation to look for and the event-log test that detects it. The discussion covers the choices a mid-sized firm faces on tooling, frequency and ownership, and the limits of what an event log can show.

Keywords: Continuous auditing, Process mining, Procure-to-pay, Conformance checking, Internal control, Event log

© 2026 Jack Wojcik et al. This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

1. Introduction

The idea of auditing transactions as they occur, and not months later from a sample, goes back at least to Vasarhelyi and Halper (1991). Pilot systems followed in large organisations, the best documented being the continuous monitoring of business process controls at Siemens (Alles et al., 2006). Three decades on, continuous auditing remains unusual in companies of ordinary size. The obstacles are well known: rules have to be written for each control, the systems that hold the data were not built to be audited, and somebody has to deal with the exceptions (Chan and Vasarhelyi, 2011).

Process mining changes the first of those obstacles. Every purchase order created, approved, received and paid leaves a time-stamped record in the enterprise system. Collected into an event log, those records

* Corresponding author: Jack Wojcik, Department of Accounting, Sharavati Research Institute of Accounting, Finland. E-mail: jack.wojcik@sharavati.edu

© 2026 Jack Wojcik et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

allow the path of each transaction to be reconstructed and compared with the path it was supposed to take (van der Aalst, 2016). Jans et al. (2013) argued that this gives auditors something no sample can: evidence on the whole population, generated by the system and not by the auditee. A field study at a European bank then showed that mining the procurement event log surfaced control failures a conventional audit had not found (Jans et al., 2014).

This article turns that body of work into a method for one common setting: the procure-to-pay cycle of a mid-sized manufacturer. Purchasing is where a manufacturer spends most of its money, the process is standardised enough to model, and the classic control failures are easy to state. The article does not describe an implementation at a particular company. It sets out what would need to be done, in what order, and what could go wrong.

2. Background

Two ideas are combined here, and it helps to be exact about each.

2.1. Continuous auditing

Continuous auditing produces assurance on a subject matter at short intervals, using automated procedures over complete data. It is usually distinguished from continuous monitoring, which is management's own check on its controls; professional guidance treats the two as complementary and stresses that the auditor must stay independent of the process being monitored (IIA, 2015). Kogan et al. (2014) designed and evaluated a continuous system working at the level of individual transactions, and showed that most of the value lies in the choice of what counts as an exception. Too tight, and real problems pass; too loose, and reviewers drown.

2.2. Process mining as audit evidence

Process mining has three main tasks (van der Aalst et al., 2012). Discovery builds a model of the process from the log alone. Conformance checking compares the log with a model and reports where they disagree. Enhancement adds information such as waiting times or the people involved. For an auditor, discovery supports understanding the process, and conformance checking is a test of controls. Chiu and Jans (2019) used it to evaluate internal control effectiveness, and Werner et al. (2021) describe how it can be embedded in a financial statement audit. Open source tools now cover the full range of techniques (Berti et al., 2019).

3. The Method

The method has eight stages. Figure 1 shows the flow of data and work, and the stages are described below. Table 1, which belongs to the fifth stage, links each control objective to a test.

3.1. Stage 1: Fix the scope and the control objectives

Start with what the controls are meant to achieve, not with what the data happen to contain. For procure-to-pay the objectives are few: purchases are authorised before commitment, goods are received before payment, invoices agree with orders and receipts, duties are separated, and vendors are genuine. The control framework the company already uses, typically the COSO framework (COSO, 2013), supplies the vocabulary. Scope is then set by materiality and risk: which company codes, which purchasing organisations, which document types.

3.2. Stage 2: Build the event log

An event log needs three things for every event: a case identifier, an activity name and a timestamp. The choice of case is the most consequential decision in the whole method. For procurement the purchase order line item is usually the right one, because quantities are received and invoiced by line. Activities are then read from the system's records: requisition created, order created, order approved, order changed, goods received, invoice received, invoice cleared for payment, payment made. In an SAP system these come from the purchasing document tables, the purchase order history, the invoice and accounting

documents, and the change log. Attributes worth carrying are the user, vendor, amount, plant and material group.

3.3. Stage 3: Check the quality of the log

Before any mining, the log is tested like any other audit evidence. Totals of ordered and invoiced value are agreed to the ledger. Cases that began before the extraction window or were unfinished at its end are identified, since an order with no payment is only an exception if the payment is overdue. Timestamps are examined for batch postings recorded at midnight and for clock differences between modules. Shared or generic user accounts are listed, because they defeat any test of who did what.

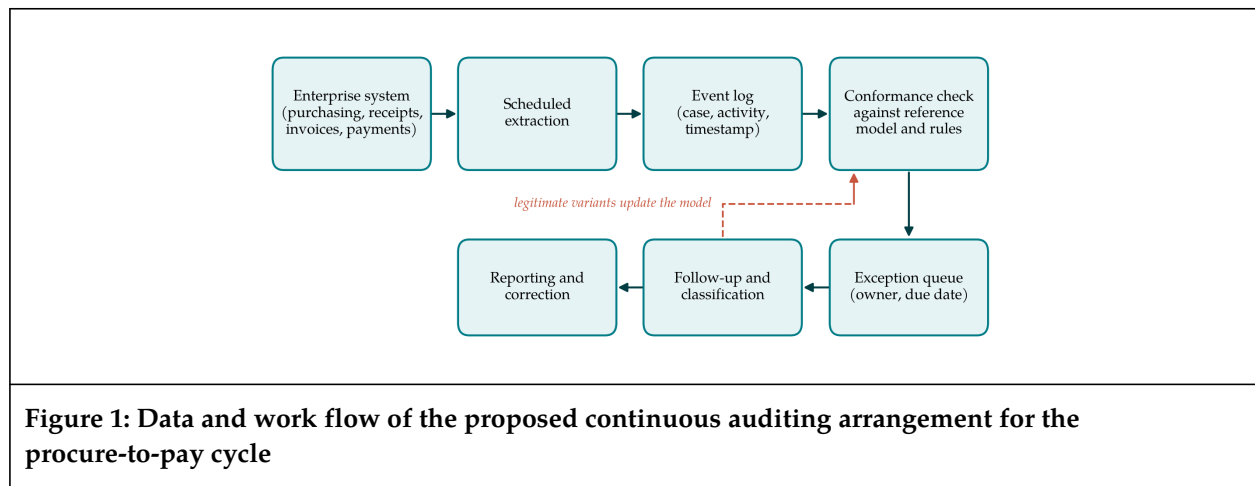


Figure 1: Data and work flow of the proposed continuous auditing arrangement for the procure-to-pay cycle

3.4. Stage 4: Discover the process as it runs

A discovery algorithm is run over the log to draw the process actually followed. Methods that guarantee a sound, readable model, such as the inductive miner (Leemans et al., 2013), suit audit use. The auditor lists the variants by frequency: commonly a small number of paths accounts for most cases and a long tail for the rest. The tail is where the audit interest lies, but the main paths must be understood first, with the people who run the process, so that legitimate variations are not mistaken for failures.

3.5. Stage 5: Define the reference model and the rules

The reference model states the permitted paths. It is derived from policy, adjusted for the legitimate variants confirmed in stage 4, and agreed with management. Alongside it sit rules that a path model cannot express, such as limits on who may perform two activities on the same case. Table 1 sets out the core set for a manufacturer.

3.6. Stage 6: Check conformance

The log is replayed against the reference model and each rule is evaluated. The output is a list of cases with the reason each failed. Two measures are worth recording for every run: the share of cases that conform, and the value they represent. A process in which nine cases in ten conform but the tenth carries most of the spending is not under control.

3.7. Stage 7: Schedule and alert

What makes the arrangement continuous is repetition. The extraction, log construction and conformance check are scripted and run on a schedule, with each run adding the events since the last. For a mid-sized manufacturer a weekly or monthly cycle is usually enough; daily runs matter only for the few rules, such as changes to vendor bank details, where delay increases the loss. Exceptions go to a queue with an owner and a due date.

3.8. Stage 8: Follow up and feed back

Each exception is examined and classified: a control failure, a legitimate variant not yet in the model, or a

data artefact. Failures are reported and tracked to correction. Legitimate variants lead to a change in the reference model, approved and dated, so that the same pattern is not raised again. Data artefacts lead to a fix in the extraction. This loop is what stops the exception count from growing until nobody looks at it.

Table 1: Control objectives for procure-to-pay, the deviation that signals failure and the event-log test for each

Control objective	Deviation to detect	Event-log test
Purchases authorised before commitment	Invoice dated before the order; order sent without approval	Invoice receipt precedes order creation; no approval event before goods receipt
Goods received before payment	Payment with no goods receipt	Payment event with no earlier goods receipt on the case
Invoice agrees with order and receipt	Price or quantity changed after approval	Order change event after approval without a second approval
Approval limits respected	One purchase split into several orders below the limit	Same vendor, requester and day; individual values under the limit, sum above it
Segregation of duties	One person creates and approves, or receives and pays	Same user on two incompatible activities within one case
Vendors are genuine	Vendor bank details changed shortly before payment	Master data change event within a set number of days before payment
No duplicate payments	Same invoice paid twice	Two payment events with matching vendor, amount and reference

4. Fitting the Method to a Mid-Sized Manufacturer

Large groups buy commercial process mining platforms and staff a team to run them. A mid-sized company has neither the budget nor the people, and three choices follow. On tooling, open source libraries are adequate for a single process and avoid licence cost, at the price of needing someone who can write and maintain scripts. On ownership, the arrangement should sit with internal audit where there is one, and otherwise with a finance function independent of purchasing; the external auditor can use the output but should not operate a control on the client's behalf. On ambition, it is better to run seven rules reliably than forty badly. The rules in Table 1 cover the classic procurement control failures.

5. Limitations

The method is presented without an implementation, so nothing here shows how many exceptions it would raise in a given company or how many would prove to be real. That is the first thing an application would need to establish. Beyond that, an event log shows only what passed through the system. Orders placed by telephone and entered afterwards, approvals given by email, and collusion between two people who each perform their own step correctly are invisible to it. Timestamps record when something was entered, which is not always when it happened. And logs contain personal data about employees, so their use has to respect local data protection and employment rules. Process mining strengthens a control environment; it is not a substitute for one.

6. Conclusion

Continuous auditing of procure-to-pay is within reach of a mid-sized manufacturer if it is built on the event log the company already produces. The eight stages set out here move from control objectives to a scheduled conformance check with a managed exception queue, and the table of tests gives a starting set of rules. Whether the effort pays for itself depends on findings that only an implementation can supply, and reporting such implementations, with their false alarms as well as their successes, would be a useful contribution.

Conflict of interest

To be declared by the author(s) before publication.

Funding

To be declared by the author(s) before publication.

References

- Alles, M., Brennan, G., Kogan, A. and Vasarhelyi, M.A. (2006). Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. *International Journal of Accounting Information Systems*, 7(2): 137-161. <https://doi.org/10.1016/j.accinf.2005.10.004>
- Berti, A., van Zelst, S.J. and van der Aalst, W.M.P. (2019). Process mining for Python (PM4Py): Bridging the gap between process and data science. arXiv preprint arXiv:1905.06169.
- Chan, D.Y. and Vasarhelyi, M.A. (2011). Innovation and practice of continuous auditing. *International Journal of Accounting Information Systems*, 12(2): 152-160. <https://doi.org/10.1016/j.accinf.2011.01.001>
- Chiu, T. and Jans, M. (2019). Process mining of event logs: A case study evaluating internal control effectiveness. *Accounting Horizons*, 33(3): 141-156. <https://doi.org/10.2308/acch-52458>
- COSO (2013). *Internal Control - Integrated Framework*. Committee of Sponsoring Organizations of the Treadway Commission, New York.
- IIA (2015). *Global Technology Audit Guide 3: Continuous Auditing: Coordinating Continuous Auditing and Monitoring to Provide Continuous Assurance (2nd ed.)*. The Institute of Internal Auditors, Altamonte Springs, FL.
- Jans, M., Alles, M. and Vasarhelyi, M. (2013). The case for process mining in auditing: Sources of value added and areas of application. *International Journal of Accounting Information Systems*, 14(1): 1-20. <https://doi.org/10.1016/j.accinf.2012.06.015>
- Jans, M., Alles, M.G. and Vasarhelyi, M.A. (2014). A field study on the use of process mining of event logs as an analytical procedure in auditing. *The Accounting Review*, 89(5): 1751-1773. <https://doi.org/10.2308/accr-50807>
- Kogan, A., Alles, M.G., Vasarhelyi, M.A. and Wu, J. (2014). Design and evaluation of a continuous data level auditing system. *Auditing: A Journal of Practice & Theory*, 33(4): 221-245. <https://doi.org/10.2308/ajpt-50844>
- Leemans, S.J.J., Fahland, D. and van der Aalst, W.M.P. (2013). Discovering block-structured process models from event logs: A constructive approach. *Application and Theory of Petri Nets and Concurrency, Lecture Notes in Computer Science*, 7927: 311-329. https://doi.org/10.1007/978-3-642-38697-8_17
- van der Aalst, W.M.P. (2016). *Process Mining: Data Science in Action (2nd ed.)*. Springer, Berlin. <https://doi.org/10.1007/978-3-662-49851-4>
- van der Aalst, W.M.P., Adriansyah, A., de Medeiros, A.K.A., et al. (2012). Process mining manifesto. *Business Process Management Workshops, Lecture Notes in Business Information Processing*, 99: 169-194. https://doi.org/10.1007/978-3-642-28108-2_19
- Vasarhelyi, M.A. and Halper, F.B. (1991). The continuous audit of online systems. *Auditing: A Journal of Practice & Theory*, 10(1): 110-125.
- Werner, M., Wiese, M. and Maas, A. (2021). Embedding process mining into financial statement audits. *International Journal of Accounting Information Systems*, 41: 100514. <https://doi.org/10.1016/j.accinf.2021.100514>

Cite this article as: Wojcik J., Cote C. and Osborne A. (2026). Setting Up Continuous Auditing With Process Mining for a Mid-Sized Manufacturer's Procure-to-Pay Cycle: A Methodological Guide. *International Journal of Accounting Research and Auditing*. 1(1), 14-18. doi: 10.62715/ijara.2026.v1.i1.pp14-18