



International Journal of Knowledge Management and Leadership

Journal homepage: <https://trpub.org/journal.php?id=obXzcXzc8r>

TRPub

IJKML

International Journal of
Knowledge
Management and
Leadership

Vol. 1 Issue 1 2026

Methodology Article

Open Access

Building a Lessons-Learned Repository That Project Teams Actually Consult: A Proposed Design Procedure

Priya Fernandez^{1*}, Latoya Rajagopal², Ignacio Sultan³

¹Department of Business, Vistula Research Institute of Business, Greece

²Department of Business, Limpopo Research Institute of Business, Serbia

³Department of Business, Nile Delta Research Institute of Business, Norway

Article Info

Volume 1, Issue 1, 2026

doi: [10.62715/ijkml.2026.v1.i1.pp15-20](https://doi.org/10.62715/ijkml.2026.v1.i1.pp15-20)

Received: 14 February 2026

Accepted: 12 April 2026

Published: 31 May 2026

Abstract

Most project-based organisations collect lessons at the end of their projects, and most find that the collection is seldom opened again. The literature on project learning has explained why with some consistency: lessons are written late, by tired teams, for no particular reader, in a form that strips out the context a later team would need, and they are stored where nobody looks at the moment a decision is being made. This article proposes a procedure for designing a lessons-learned repository from the opposite end, starting with the occasions on which a project team could use a lesson and working back to what must be captured and how. The procedure has six stages: identify the points of use, define what counts as a lesson, capture at stage gates, validate and assign ownership, deliver lessons into the workflow, and review and retire them. For each stage the article states the design decision to be made, the reasoning from published work and the output to be produced. The procedure is a proposal built from the literature and from established project management guidance. It has not been applied or evaluated here, and the article sets out what an organisation adopting it would need to measure to find out whether it works.

Keywords: *Lessons learned, Project knowledge management, Knowledge reuse, Post-project review, Organisational learning, Repository design*

© 2026 Priya Fernandez 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 lessons-learned exercise is among the most widely prescribed practices in project management. The guide of the Project Management Institute gives it a place in the process for managing project knowledge and provides for a lessons learned register to be kept throughout the project and transferred to an organisational repository at its close (PMI, 2017). The international standard for knowledge management systems expects an organisation to retain and make available what it learns (ISO, 2018). Compliance with the letter of these expectations is common. A survey of practitioners by Williams (2008) found lessons-learned processes in general use and, at the same time, widespread doubt among those who ran them that the lessons reached later projects.

* Corresponding author: Priya Fernandez, Department of Business, Vistula Research Institute of Business, Greece. E-mail: priya.fernandez@vistula.edu

© 2026 Priya Fernandez 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.

That doubt is the subject of this article. A repository that is filled and not read consumes effort at the end of every project and returns nothing, and it teaches teams that the exercise is a formality. The proposition advanced here is that the usual order of design is backwards. Organisations begin with capture, by issuing a template and a database, and hope that use will follow. The procedure proposed below begins with use. It is offered as a design method assembled from published research and guidance, and it should be read as such: it has not been tried in an organisation as part of this work.

2. Why Repositories Go Unread

The research on learning between projects is unusually agreed about what goes wrong. [Keegan and Turner \(2001\)](#) found that project-based firms had review practices in place and that time pressure routinely defeated them, with reviews deferred until the team had dispersed. [von Zedtwitz \(2002\)](#) grouped the barriers to post-project review into psychological, team-based, epistemological and managerial kinds, among them reluctance to revisit failure and difficulty in generalising from a single case. [Schindler and Eppler \(2003\)](#) compared debriefing methods and concluded that learning should be gathered at intervals during the project and not left to the end, when memory has faded and attention has moved on.

A second group of findings concerns the reader. [Newell et al. \(2006\)](#) showed that documents captured through project review systems were little used by other teams, who preferred to ask someone they knew, and that what was written down tended to describe what had been delivered and to omit how the work had been done. [Markus \(2001\)](#) explained the pattern in a theory of knowledge reuse: records made by a team for its own purposes are seldom usable by strangers, because the strangers lack the context the authors took for granted, and making a record reusable is additional work for which the author is rarely rewarded. [Hartmann and Dorée \(2015\)](#) criticised the underlying picture of one project sending messages to the next and argued that learning takes hold only where it is tied to the work the receiving team is doing.

A third finding is that storage is not learning. [Milton \(2010\)](#) draws the distinction between a lesson identified and a lesson learned, the second being reserved for cases in which something has changed as a result: a procedure, a standard, a checklist, a design rule. [Levitt and March \(1988\)](#) had described organisational learning in the same terms, as the encoding of inferences from experience into routines that guide behaviour. A database entry that alters no routine has, on this view, taught the organisation nothing, however well it is written.

3. Design Principles

Three principles follow from this diagnosis and govern the procedure. The first is to design from the point of use. [Weber et al. \(2001\)](#), reviewing lessons-learned systems in government and industry, observed that nearly all relied on passive dissemination, waiting for a user to search, and argued for delivering lessons inside the tools and processes where the relevant decision is taken. The second is to prefer embedding to storing. Wherever a lesson can be turned into a change in a template, a standard or a checklist, that change is the primary product and the repository entry is its record. The third is that every lesson has an owner, a named role responsible for deciding what is done with it and for retiring it when it no longer applies.

These principles describe a loop and not a pipeline. Figure 1 shows the intended circulation: lessons are captured during projects, validated, embedded or pushed to the teams that need them, and then reviewed in the light of whether they were used. The remainder of the article sets out the stages by which an organisation might build such a loop.

4. The Proposed Procedure

The procedure has six stages, listed in Table 1 with the decision each requires and the output it should leave. The first two are design work done once and revisited occasionally. The remaining four describe the repository in operation.

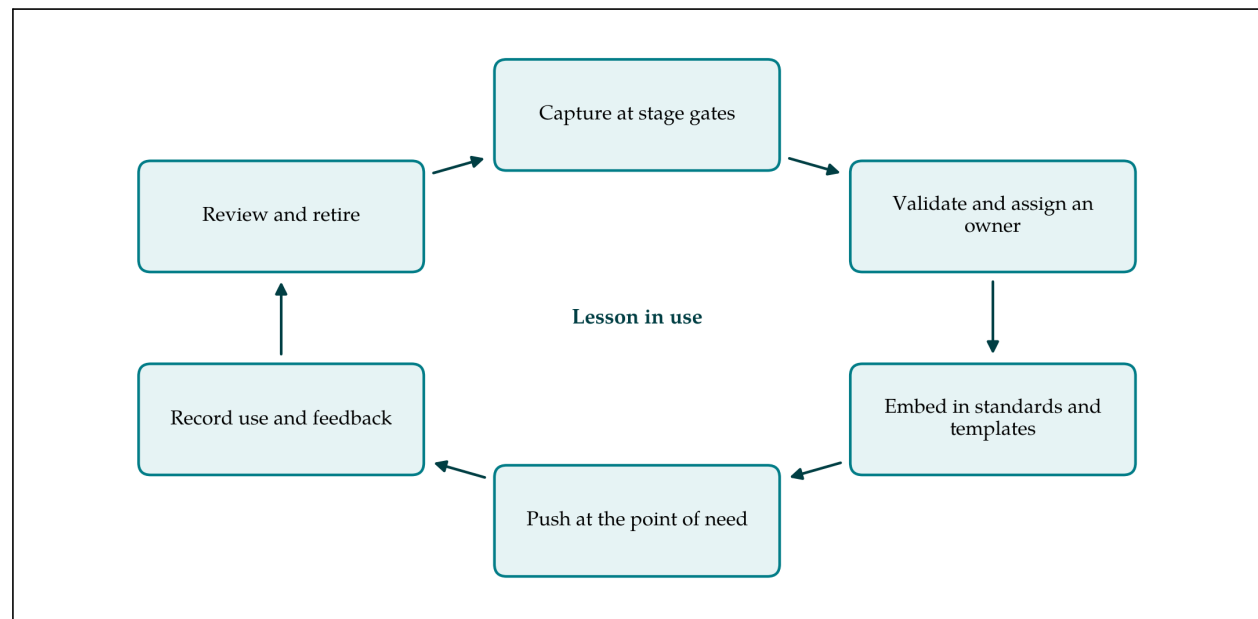


Figure 1: The intended circulation of a lesson in the proposed design, from capture to retirement

Stage	Design decision	Output
1. Identify points of use	At which moments in a project could a past lesson change a decision?	Map of decision points and the roles present
2. Define a lesson	What must an entry contain to be actionable by a stranger?	Entry format and acceptance criteria
3. Capture at stage gates	When, by whom and with what facilitation are lessons gathered?	Short reviews scheduled at each gate
4. Validate and assign	Who judges an entry and who owns the response?	Validated lessons, each with an owner and an action
5. Deliver into the workflow	How does a lesson reach a team without being searched for?	Changed templates and standards; prompts at gates
6. Review and retire	How is use recorded and when is an entry withdrawn?	Usage record and periodic cull

4.1. Stage 1: Identify the points of use

The designer begins by listing the recurring moments at which project teams make decisions that earlier experience could inform. In most project organisations these include bid or business case preparation, estimating, risk identification, selection of suppliers, design freeze, and the planning of commissioning or handover. For each, the designer records who is present, what documents or tools they use and how much time they have. This list, and not a taxonomy of subject matter, becomes the organising scheme of the repository. The classification adopted by Markus (2001) is a useful check: a team reusing its own records, practitioners in the same role elsewhere and novices seeking expertise need different things from the same entry.

4.2. Stage 2: Define what counts as a lesson

An entry format is then fixed. Following the criteria that Weber et al. (2001) derive from the definitions in use, a lesson should be grounded in an actual event, should be valid in the sense that the facts are right, and should be applicable in that it names a decision or process it would change. A workable format has

four parts: the situation in enough detail for an outsider to judge similarity, what happened, the recommendation phrased as an action, and the point of use from Stage 1 to which it applies. Entries that amount to complaints or to restatements of existing procedure are declined. A contact name is included, because the evidence that people prefer to ask a person is strong and the repository should serve that preference and not compete with it.

4.3. Stage 3: Capture at stage gates

Capture is attached to the stage gates the project already observes, so that it happens while events are fresh and before the team disperses. Each session is short and facilitated by someone outside the team. The after action review offers a tested pattern of questions: what was intended, what occurred, why the two differed and what will be done differently. [Darling et al. \(2005\)](#), describing its use in military training, stress that the review is paired with a briefing before the next action in which earlier lessons are applied. [Kerth \(2001\)](#), writing on retrospectives in software projects, adds the ground rule that a retrospective looks for causes in the situation and not for culprits. Without that rule teams record only what is safe to say.

4.4. Stage 4: Validate and assign ownership

Raw entries pass to a reviewer with authority in the relevant discipline, who checks them against the format, merges duplicates and decides the response. The preferred response is to embed: to amend the estimating norm, the specification clause, the risk checklist or the contract template so that the next team meets the lesson as part of its ordinary work. Where a lesson cannot be embedded, because it calls for judgement in particular circumstances, it is retained as an entry linked to a point of use. In either case a named role owns it. [Prencipe and Tell \(2001\)](#) distinguish learning by accumulated experience, by articulation and by codification, and this stage is where the organisation chooses how far along that sequence a given lesson should be taken.

4.5. Stage 5: Deliver lessons into the workflow

Entries that remain in the repository are delivered at the points of use and are not left to be found. The simplest mechanism is procedural: the gate checklist for each stage requires the team to review the lessons tagged to that stage and to record which it has acted on. Where project tools allow, the lessons for a stage can be shown inside the risk register or the estimating workbook when that document is opened. A start-up briefing for each new project, in which a facilitator presents the handful of lessons most relevant to its type, serves the same purpose by conversation. The number shown at any one time should be small. A team presented with every lesson on record will read none of them.

4.6. Stage 6: Review and retire

The repository records each occasion on which a lesson is shown and whether the team marked it as applied, not applicable or disputed. Owners review their lessons at a fixed interval. Entries that have been embedded in a standard are closed with a reference to the change. Entries that are never applied are examined and either rewritten or withdrawn, and disputed entries are returned to the reviewer. The review stage thus applies to the repository the same cycle of planning, doing, checking and acting that it asks of projects.

5. Fitting the Procedure to the Organisation

The procedure assumes a degree of repetition between projects and some central capacity to act as reviewer. Organisations differ in both. [Hansen et al. \(1999\)](#) distinguished firms that compete by reusing codified knowledge from those that solve unique problems through expert contact, and the balance between embedded lessons and entries that point to a person should shift accordingly. A small organisation may combine reviewer and owner in one senior role and run the repository as a shared document tied to its gate checklists. A large one will need the roles separated by discipline. [Carrillo et al. \(2013\)](#), examining practice among construction contractors, found that practices varied widely and that the people involved mattered more than the technology, which suggests starting with the reviews and the

ownership rule and adding software later.

Culture sets a limit that design cannot remove. Duffield and Whitty (2015) model lessons learned as passing through layers of people and systems, any one of which can block them. A procedure can make use easier and more visible. It cannot make a senior manager welcome an entry that reflects badly on a decision of the manager's own.

6. Limitations and What Remains to Be Shown

The procedure is a proposal. Its stages are each supported by published argument or observation, but their combination has not been tested here, and it is possible that the stages interact in ways the sources do not anticipate. An organisation adopting it should decide in advance how it will judge the result. Suitable indicators include the proportion of validated lessons that end in a changed standard, the proportion of gate reviews at which a lesson is marked as applied, and whether a problem recorded as a lesson recurs in later projects. The last is the most meaningful and the slowest to observe.

Two risks deserve mention. Mandatory review at gates can decay into ticking a box, which would reproduce the original failure in a new place; periodic sampling of how teams responded is a partial defence. And embedding lessons in standards makes the standards longer, so that the cull in Stage 6 must extend to rules that past lessons introduced. Comparative study of organisations that organise their repositories by subject and by point of use would show whether the central claim of this article holds.

7. Conclusion

Lessons-learned repositories fail for reasons that are well documented: late capture, entries written for no reader, passive storage and the absence of anyone responsible for acting. The procedure proposed here addresses each by designing from the moments at which teams could use a lesson, by treating a changed routine as the proper form of a learned lesson, and by giving each lesson an owner and a review date. Whether the design delivers more consultation and fewer repeated mistakes is a question for organisations that try it and measure the outcome.

Conflict of interest

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

Funding

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

References

- Carrillo, P., Ruikar, K. and Fuller, P. (2013). [When will we learn? Improving lessons learned practice in construction.](#) *International Journal of Project Management*, 31(4): 567-578.
<https://doi.org/10.1016/j.ijproman.2012.10.005>
- Darling, M., Parry, C. and Moore, J. (2005). [Learning in the thick of it.](#) *Harvard Business Review*, 83(7): 84-92.
- Duffield, S. and Whitty, S.J. (2015). [Developing a systemic lessons learned knowledge model for organisational learning through projects.](#) *International Journal of Project Management*, 33(2): 311-324.
<https://doi.org/10.1016/j.ijproman.2014.07.004>
- Hansen, M.T., Nohria, N. and Tierney, T. (1999). [What's your strategy for managing knowledge?](#) *Harvard Business Review*, 77(2): 106-116.
- Hartmann, A. and Dorée, A. (2015). [Learning between projects: More than sending messages in bottles.](#) *International Journal of Project Management*, 33(2): 341-351.
- ISO (2018). [ISO 30401:2018 Knowledge management systems: Requirements.](#) International Organization for Standardization, Geneva.

- Keegan, A. and Turner, J.R. (2001). [Quantity versus quality in project-based learning practices. *Management Learning*, 32\(1\): 77-98.](#)
- Kerth, N.L. (2001). [Project Retrospectives: A Handbook for Team Reviews.](#) Dorset House, New York.
- Levitt, B. and March, J.G. (1988). [Organizational learning. *Annual Review of Sociology*, 14: 319-340.](#)
<https://doi.org/10.1146/annurev.so.14.080188.001535>
- Markus, M.L. (2001). [Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *Journal of Management Information Systems*, 18\(1\): 57-93.](#)
- Milton, N. (2010). [The Lessons Learned Handbook: Practical Approaches to Learning from Experience.](#) Chandos Publishing, Oxford.
- Newell, S., Bresnen, M., Edelman, L., Scarbrough, H. and Swan, J. (2006). [Sharing knowledge across projects: Limits to ICT-led project review practices. *Management Learning*, 37\(2\): 167-185.](#)
<https://doi.org/10.1177/1350507606063441>
- PMI (2017). [A Guide to the Project Management Body of Knowledge \(PMBOK Guide\).](#) 6th edn. Project Management Institute, Newtown Square, PA.
- Prencipe, A. and Tell, F. (2001). [Inter-project learning: Processes and outcomes of knowledge codification in project-based firms. *Research Policy*, 30\(9\): 1373-1394.](#)
[https://doi.org/10.1016/S0048-7333\(01\)00157-3](https://doi.org/10.1016/S0048-7333(01)00157-3)
- Schindler, M. and Eppler, M.J. (2003). [Harvesting project knowledge: A review of project learning methods and success factors. *International Journal of Project Management*, 21\(3\): 219-228.](#)
[https://doi.org/10.1016/S0263-7863\(02\)00096-0](https://doi.org/10.1016/S0263-7863(02)00096-0)
- von Zedtwitz, M. (2002). [Organizational learning through post-project reviews in R&D. *R&D Management*, 32\(3\): 255-268.](#)
- Weber, R., Aha, D.W. and Becerra-Fernandez, I. (2001). [Intelligent lessons learned systems. *Expert Systems with Applications*, 20\(1\): 17-34.](#) [https://doi.org/10.1016/S0957-4174\(00\)00046-4](https://doi.org/10.1016/S0957-4174(00)00046-4)
- Williams, T. (2008). [How do organizations learn lessons from projects - and do they? *IEEE Transactions on Engineering Management*, 55\(2\): 248-266.](#) <https://doi.org/10.1109/TEM.2007.912920>