

This is the author's version of a work that is published through the following outlet:

Hendriks, D.; Meier, I.; Wiegand, H.; Benz, C.; Heinz, D.; Spitzer, P. (2026): From Pilot to Production: Microfoundations and Responsibility Patterns for Enterprise AI Projects. *Proceedings of the 2026 International Conference on Information Systems (ICIS 2026)*.

Please note: Copyright is owned by the author and/or the publisher.
Commercial use is not allowed.

From Pilot to Production: Microfoundations and Responsibility Patterns for Enterprise AI Projects

Completed Research Paper

Daniel Hendriks*

Karlsruhe Institute of Technology
Karlsruhe, Germany
daniel.hendriks@kit.edu

Ida Meier*

Karlsruhe Institute of Technology
Karlsruhe, Germany
ida.meier@kit.edu

Henrik Wiegand

Karlsruhe Institute of Technology
Karlsruhe, Germany
henrik.wiegand@kit.edu

Carina Benz

Karlsruhe Institute of Technology
Karlsruhe, Germany
carina.benz@kit.edu

Daniel Heinz

Karlsruhe Institute of Technology
Karlsruhe, Germany
daniel.heinz@kit.edu

Philipp Spitzer

Karlsruhe Institute of Technology
Karlsruhe, Germany
philipp.spitzer@kit.edu

*** These authors contributed equally to this work.**

Abstract

Enterprise AI projects in firms frequently fail to transition from pilot to production. This study characterizes this transition through 14 projects across multiple industries. Using an inductive multiple-case design and a microfoundations lens grounded in the dynamic capabilities framework, we identify 38 microfoundations and 12 meso-level capabilities spanning sensing, seizing, and transforming. While sensing and seizing activities largely resemble prior digital transformation research, the transforming dimension diverges, centering on operational AI governance, knowledge institutionalization, and AI asset evolution. We further develop a two-dimensional framework of responsibility allocation across piloting and productionizing, identifying three responsibility patterns: domain-driven, handover, and tech-driven. For theory, the transforming dimension requires a microfoundation profile distinctive to AI projects, and responsibility allocation is a phase-sensitive structural variable rather than a steady-state governance property. For practice, AI competence distribution is the key precondition shaping which responsibility patterns are sustainable – and something firms can deliberately develop.

Keywords: Enterprise AI, Microfoundations, Responsibility Allocation

Introduction

Artificial intelligence (AI) has become a strategic priority for large incumbent firms, which expect it to deliver efficiency gains, new forms of value creation, and competitive differentiation (Krakowski et al., 2023). Most firms launch AI *pilots*, limited-scope trials intended to demonstrate feasibility, before *productionizing* the resulting systems: deploying and using them at scale. Yet pilots rarely make this

transition. A 2026 Deloitte survey of 3,235 business and IT leaders across 24 countries reports that only 25% of organizations have moved 40% or more of their AI pilots into production (der Weduwe, 2026), consistent with prior IS evidence (Dzhusupova et al., 2024; Peretz-Andersson et al., 2024). The two phases place fundamentally different demands on the firm: a pilot can demonstrate feasibility with a small team and a contained scope, while embedding the resulting system into organizational workflows and infrastructure requires sustained reconfiguration – what prior research has termed the *pilot-to-production gap* (Ångström et al., 2023; Dzhusupova et al., 2024).

The gap is particularly acute for AI. Unlike conventional IT, AI is opaque, learns from data rather than from explicit specification, and produces probabilistic outputs (Benbya et al., 2021; Berente et al., 2021). Because outputs are non-deterministic, users cannot simply consume them; they must learn to interpret, verify, and at times question them, reshaping the practices the system is meant to support (van den Broek et al., 2021). Whereas conventional IT is configured once and then maintained, AI systems keep changing as data and usage shift, requiring organizational attention well beyond the initial rollout. These properties weigh heaviest on organizations, whose established routines, legacy infrastructures, and path dependencies raise the cost of reconfiguring roles, processes, and structures (Haefner et al., 2023; Raisch & Krakowski, 2021).

A growing body of AI management research seeks to explain why some firms succeed with AI while others stall. Much of this work examines organizational capabilities (Enholm et al., 2022; Mikalef & Gupta, 2021), the competencies needed to manage data dependency and model inscrutability (Weber et al., 2023), implementation challenges (Ångström et al., 2023), management risks (Hutzschenreuter et al., 2025), and adoption antecedents along technological, organizational, and environmental dimensions (Jöhnk et al., 2021). What this literature does not yet provide is a concrete account of *how* successful transitions unfold – neither which activities firms perform across the project lifecycle (Dzhusupova et al., 2024), nor how responsibility for those activities is distributed between technical and domain functions (Berente et al., 2021). We therefore ask:

RQ1: *What organizational activities and capabilities characterize AI projects that successfully transition from pilot to production?*

Among these activities, responsibility allocation deserves particular attention. Because AI projects span technical and domain work, allocation shapes who is accountable for outputs, who maintains the system, and who decides when it is producing acceptable results. Productionizing also introduces operational concerns (maintenance, monitoring, user adoption) that differ from those of piloting, raising the possibility that responsibility shifts as a project moves between phases. How responsibility is allocated within each phase, and how it shifts, has not been examined as a structural variable. We therefore ask:

RQ2: *How is responsibility for AI project work allocated between technical and domain functions, and how does this allocation shift between the piloting and productionizing phases?*

We address these questions through an interview study of 14 enterprise AI projects across multiple industries. Using inductive Gioia-style analysis, with the microfoundations lens of the dynamic capabilities framework (Teece, 2007) as an organizing scaffold, we identify 38 microfoundations, 12 meso-level capabilities, and three aggregate dimensions (sensing, seizing, transforming). We then develop a two-dimensional framework characterizing how responsibility is allocated across the piloting and productionizing phases. Mapping our cases onto this framework reveals three patterns: *domain-driven*, *handover*, and *tech-driven*, illustrated through three vignette cases. In our data, the location and depth of AI competence within the firm – the technical capacity to develop and operate AI systems and the domain capacity to frame use cases and validate their probabilistic outputs – can constrain which arrangements are sustainable: responsibility tends to settle where the underlying competence is already present.

We make three contributions. First, we identify the microfoundations and meso-level capabilities of AI productionization across sensing, seizing, and transforming, and show that while sensing and seizing capabilities largely extend those identified for digital transformation more broadly, transforming meso-level capabilities systematically diverge. Second, we develop a two-dimensional responsibility framework that captures how responsibility is allocated and shifts across phases. Third, we identify three recurring patterns and propose AI competence distribution as a candidate structural precondition shaping which pattern emerges. These contributions reframe AI productionization as a coordination problem whose solution depends on where AI competence already sits in the firm.

Foundations and Related Work

This section reviews three streams of literature that ground our study. We begin with research on the AI pilot-to-production gap, which establishes the phenomenon. We then turn to the dynamic capabilities framework and its microfoundations construct, which we use as an analytical scaffold for organizing the activities through which AI projects move from pilot to production (RQ1). Finally, we review work on responsibility allocation in AI projects, which motivates our analysis of how responsibility is distributed and shifted across project phases (RQ2).

The AI Pilot-to-Production Gap

Enterprise AI takes different forms. Following Feuerriegel et al. (2024), we distinguish AI models (trained networks), AI systems (models embedded in infrastructure, data pipelines, and interfaces), and AI applications (systems situated in a task context to solve a business problem). The projects we study span these levels: *pilots*, i.e., limited scope trials to demonstrate feasibility and build organizational support, typically center on models and early systems; *productionization* then turns them into applications embedded in business processes (deployment, scaling, and continuous updating in operational environments) so that user or business value can be sustained (Dzhusupova et al., 2024; Haefner et al., 2023). Although AI pilots often produce promising results, organizations across industries continue to struggle to convert them into production systems (der Weduwe, 2026).

The literature frames this gap as an organizational and managerial challenge, not merely a technical one. Hutzschenreuter et al. (2025) identify five management risks that hinder AI scaling; Peretz-Andersson et al. (2024) emphasize orchestrating technological, human, and organizational resources alongside governance and learning capabilities; and Sharma et al. (2026) show that moving from a generative AI pilot to productive use requires developing enterprise-wide capabilities. Practitioner-oriented and engineering literatures add barriers around workflow redesign, data sensitivity, intellectual property, legal constraints, and scale-up infrastructure (Dzhusupova et al., 2024; Fountaine et al., 2019). Across this body of work, two questions remain underspecified: which *concrete activities* firms perform across the project lifecycle, and through which *structural arrangements* responsibility for those activities is allocated and shifted.

Dynamic Capabilities and Microfoundations

Dynamic capabilities describe a firm's higher-order ability to adapt, learn, and reconfigure its resource base in response to changing environments, enabling purposeful renewal rather than routine operations (Teece, 2007; Teece et al., 1997). Their relevance is greatest under rapid and uncertain technological change (Teece, 2007) – conditions that AI clearly meets.

The *microfoundations* lens directs attention to the concrete individual, team, and organizational processes through which dynamic capabilities are enacted (Salvato & Vassolo, 2018; Teece, 2007). Rather than treating sensing, seizing, and transforming as abstract firm-level attributes, microfoundations make visible the distributed activities, knowledge flows, and interactions through which higher-order capabilities operate (Simón et al., 2024; Teece & Linden, 2017). For AI productionization, this lens is useful because the transition from pilot to production is achieved not through strategic intent alone but through coordinated activities across business units, data science teams, IT, legal, compliance, and operations.

Although dynamic capabilities have been widely applied, including in digital transformation research, their use in AI adoption and deployment is still emerging (Chirumalla, 2021; Schneider et al., 2024). Prior work has focused either on macro-level organizational routines or on micro-level managerial decision-making (Adner & Helfat, 2003; Teece, 2007). Recent work emphasizes a *meso-level* perspective on interactions among employees and teams, between the individual and the firm, as the link between managerial action and organizational transformation (Salvato & Vassolo, 2018; Simón et al., 2024). Closest to our work, Warner and Wäger (2019) identify nine microfoundations through which incumbent firms build dynamic capabilities for digital transformation, organized along Teece's (2007) sensing-seizing-transforming dimensions. Their study, however, treats digital transformation as a general phenomenon spanning a range of technologies. This leaves open which microfoundations and meso-level capabilities are distinctive to AI given its probabilistic, data-coupled, and opaque properties – and how they are enacted at the meso-level rather than the firm level.

We therefore use the microfoundations construct as an analytical scaffold for organizing our inductive findings about reconfiguration practices specific to AI. Following Salvato and Vassolo (2018), we treat the 38 first-order concepts identified in our analysis as microfoundations (the concrete individual- and team-level activities) and the 12 second-order themes as the meso-level capabilities they enact. We do not theorize AI productionization as a dynamic capability or extend dynamic capabilities theory; the microfoundations vocabulary offers a level of granularity well suited to identifying the concrete organizational activities through which AI projects move from pilot to production (RQ1).

Responsibility Allocation in AI Projects

AI projects sit at the intersection of technical work (model development, data engineering, integration, monitoring) and domain work (use case framing, validation, operational use), making them inherently cross-functional. IS work on AI governance has emphasized that responsibility is distributed across technical teams, business units, legal, and risk management, and that governing AI requires coordinated processes spanning these functions (Mäntymäki et al., 2022; Schneider et al., 2023). Building on this view, Papagiannidis et al. (2025) propose that AI governance is enacted through *structural* arrangements (boards, committees, roles), *procedural* arrangements (policies and processes), and *relational* arrangements (cross-functional collaboration and communication) – but treat the appropriate mix as broadly applicable across the AI lifecycle, without examining how it varies by phase. Practitioner and academic work has also documented how firms organize AI project work itself: centralized AI centers of excellence, business-unit-embedded teams, and hybrid models (Davenport & Ronanki, 2018; Fountaine et al., 2019). Related work has examined how firms orchestrate technical, human, and organizational resources for AI implementation (Peretz-Andersson et al., 2024). A complementary stream conceptualizes AI development at the firm level as a maturity progression along a single developmental path of staged capability (Hansen et al., 2024).

What is missing from this literature is an account of how responsibility for AI projects is allocated *within* phases and reallocated *across* them. The governance literature largely treats arrangements as firm-level and steady-state. Responsibility for specific AI projects, however, can shift mid-project as piloting gives way to productionizing, and that shift is consequential because the two phases place different demands on the firm. Prior studies treat organizational structure largely as a background variable (Dzhupova et al., 2024) or examine governance arrangements at a single point in time (Mäntymäki et al., 2022; Schneider et al., 2023), without tracing how responsibility evolves as a project moves toward operational deployment. How responsibility allocation varies across AI projects that successfully reach production is the focus of RQ2.

Research Methodology

We conduct a qualitative multiple-case study to characterize how AI projects transition from pilot to production. The multiple-case design (Eisenhardt, 1989) lets us analyze each project in depth while comparing patterns across cases – well suited to a descriptive aim that targets activities, routines, and responsibility arrangements rather than causal effects.

By *enterprise AI*, we mean customized AI systems developed and used internally by large incumbent firms – applications adapted to firm-specific data, processes, or requirements rather than off-the-shelf software, with intended users among the firm’s own employees or organizational units. An *enterprise AI project* is the organizational initiative through which such a system is developed, deployed, and used; we use ‘AI project’ as shorthand throughout the paper. We restrict the sample to this scope because productionizing customized internal AI is what generates the organizational reconfiguration challenges Section 1 motivates.

Sampling and Data Collection

We identified candidate AI projects through LinkedIn and personal networks, screening for successfully productionized AI use cases, which we define as use cases that are either fully embedded in, or currently being integrated into, their intended operational context. This definition deliberately captures deployment rather than adoption or value realization. In later data collection stages, we sampled more selectively to include projects with different responsibility arrangements, consistent with theoretical sampling in qualitative research (Corbin & Strauss, 1990; Eisenhardt, 1989). The final sample comprised 14 AI projects (Case A-N) across seven companies (Organization 1-7; revenue €3–60 bn).

We conducted 15 case interviews covering these projects (Case B involved two interviewees), and 2 contextual interviews with senior managers (I16, I17) whose remit spans broader AI initiatives at firms in our sample but who were not associated with any specific case in Table 1. Interviews ran from June 2025 to April 2026 and totaled 877 minutes (≈ 14.5 hours). Most were conducted online, in German or English; recordings were transcribed in full, and German quotations were translated into English for the paper. Table 1 summarizes the cases and interviewees.

Organization	Case	Project Description	Role of Interviewee(s)	Time
1 Software	A	Agentic Research System	AI Product Engineer (I1)	58 min
2 Energy	B	ML Custom Service System	Data Scientist (I2)	54 min
			Team Lead Process Mining (I3)	54 min
3 Wholesale	C	RAG System	VP Supply Chain AI (I4)	52 min
4 IT Consulting	D	RAG System	Consultant Data & AI (I5)	51 min
	F	Enterprise AI Platform	AI Innovation Executive (I7)	56 min
	G	Enterprise AI Platform	Senior Managing Consultant (I8)	47 min
5 Banking	E	RAG System	Controlling Specialist (I6)	40 min
6 Chemicals	I	RAG Chatbot	Full-Stack AI Developer (I10)	59 min
	J	Agentic Sales Support	Embedded Data Scientist (I11)	72 min
	K	Agentic Comm. Support	Senior Managing Consultant (I12)	60 min
	L	AI Contract Processing	Legal Manager (I13)	53 min
	M	Decision Support Agent	AI Solution Manager (I14)	48 min
7 Manufacturing	H	ML Quality Inspection	ML Engineer (I9)	54 min
	N	Predictive Testing System	Team Lead Data Science and AI (I15)	57 min
Contextual Interviews	-	-	Team Lead AI Transformation (I16)	30 min
	-	-	Industrial AI Lead (I17)	32 min
Table 1. Case and Interview Details				

Cases A-N and contextual interviews I16 (Chemicals), I17(IT Consulting) informed RQ1; the RQ2 analysis used 11 cases (E, F, G excluded).

The interview guide was adapted iteratively over the study. Early interviews covered the interviewee's role, organizational context, project origin, piloting and productionizing of the AI system, the actors involved, the responsibilities held, and perceived success factors. Subsequent interviews followed the implementation process more systematically, with attention to who held responsibility at each phase, and how that allocation was negotiated between technical and domain functions. We concluded data collection when the sample produced sufficient empirical variation for our descriptive aim; we approached theoretical saturation but do not claim to have reached it.

Data Analysis

Data were managed in MAXQDA. The analysis proceeded in two phases corresponding to the two research questions. RQ1 drew on the full material – the 15 case interviews covering 14 AI projects across seven organizations, plus the two contextual interviews. RQ2 drew only on 11 bounded use cases for which sufficient case-level evidence was available to assess responsibility during both piloting and productionizing; Cases E, F, and G were excluded because Case E reflected a user-side perspective and Cases F and G lacked sufficient use-case-specific data for comparison.

For RQ1, we coded inductively following the Gioia methodology (Gioia et al., 2013). Coding began with informant-centric first-order codes, including in-vivo codes, to stay close to interviewees' terminology. Coding was led by one author and refined through iterative discussions with co-authors. First-order codes

were then grouped into second-order themes and these were related to the aggregate dynamic capability dimensions of sensing, seizing, and transforming. Following Salvato and Vassolo (2018), we treat the 38 first-order concepts as microfoundations (the concrete individual- and team-level activities) and the 12 second-order themes as the meso-level capabilities they enact. We related themes to the dynamic capabilities framework only after the inductive coding structure had stabilized: sensing, seizing, and transforming were not initial coding categories but a subsequent organizing lens. As coding progressed, we refined and challenged earlier patterns through constant comparison (Corbin & Strauss, 1990); several first-order concepts and second-order themes were renamed or restructured as material accumulated.

For RQ2, we ran a focused case classification analysis of the 11 bounded use cases. The motivation came from RQ1 analysis: several second-order themes (particularly responsibility definition, cross-functional collaboration, and embedded AI competence) pointed to the importance of how responsibility was allocated between technical and domain actors and how that allocation shifted across phases. We coded each case along two dimensions: the locus of responsibility during piloting and the locus during productionizing. Each dimension used a five-point scale, anchored at *tech-only* and *domain-only*, with intermediate categories of *tech-led*, *balanced*, and *domain-led*. A case was coded *tech-only* (or *domain-only*) when the focal activities – use-case identification, evaluation, readiness assessment, and solution development for piloting; maintenance, governance, technical asset reuse, knowledge documentation, and enablement for productionizing – were carried out exclusively by tech (or domain) actors; *tech-led* or *domain-led* when one side held primary responsibility while the other contributed actively; and *balanced* when the two sides shared activities approximately equally.

Two researchers independently coded each case along both dimensions and identified supporting quotations. Initial agreement was 6 of 11 cases for the piloting axis and 8 of 11 cases for the productionizing axis. The remaining disagreements reflected the interpretive judgment involved in weighing relative cross-functional contributions, particularly in cases where one function formally held responsibility but exercised it operationally across both. All disagreements were resolved through a joint reading of supporting transcript passages and consensus discussion. The consolidated case positions were then compared to identify recurring clusters across the two phases, from which one vignette case per cluster was selected for detailed presentation.

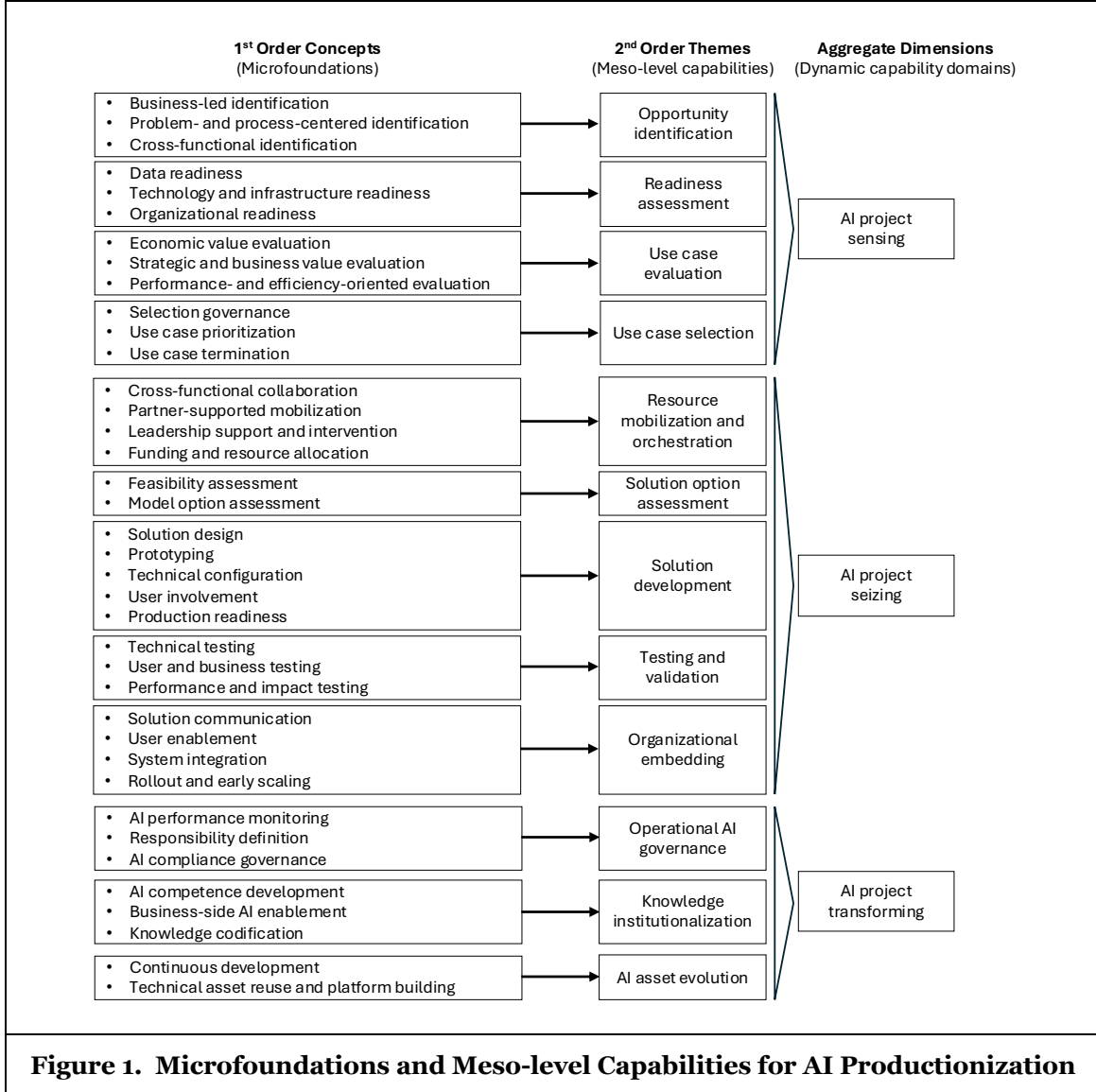
Findings: Microfoundations and Responsibility Patterns

This section is organized in three parts. We first identify the organizational activities and capabilities underlying successful AI productionization. We then examine how responsibility for these activities is allocated across technical and domain functions during piloting and productionizing. Finally, we illustrate the identified responsibility patterns and microfoundations through three vignette cases.

Microfoundations and Meso-Level Capabilities of Enterprise AI Productionization

To address RQ1, we draw on the 15 case interviews covering 14 AI projects across seven organizations, and 2 contextual interviews. Building on inductive Gioia-style analysis with the microfoundations lens of the dynamic capabilities framework (Teece, 2007) as an organizing scaffold, we generated 603 initial codes, consolidated into 38 first-order concepts (microfoundations) and abstracted into 12 second-order themes (meso-level capabilities). These aggregate into three capability domains: AI Project Sensing, AI Project Seizing, and AI Project Transforming.

Successful AI productionization is a capability-building process extending well beyond the initial deployment event, since a deployed AI system continues to change its behavior as data and usage evolve. Organizations sense viable AI opportunities against business relevance and expected value, with two AI-specific preconditions: data readiness and technology readiness. They seize selected opportunities by mobilizing resources, assessing model and solution options, developing and validating AI solutions with domain users since output validity cannot be assessed by technical testing alone, and embedding them into workflows, systems, and rollout structures. They transform project-level outcomes into durable organizational capabilities through operational AI governance, knowledge institutionalization, and AI asset evolution. This involves especially ongoing monitoring of AI performance and defining responsibilities. Figure 1 summarizes the three domains and their underlying meso-level capabilities and microfoundations; the following subsections unpack each domain.



AI Project Sensing

AI Project Sensing comprises four meso-level capabilities through which organizations identify and filter potential projects: opportunity identification, readiness assessment, use case evaluation, and use case selection (Figure 1). Each is enacted through more specific microfoundations.

Opportunity identification refers to the activities through which organizations recognize potential AI projects and connect them to business needs. In our data, this activity rarely started from technology. Instead, participants described *business-led* searches for value: “I have a business target – productivity, revenue, or whatever. Then we ask how we can achieve that with technology” [I17]. In other cases, opportunities arose from existing domain needs: “the target area was already extremely excited” because it was something “they had always wanted a solution for” [I2]. Other participants emphasized *problem- and process-centered identification*: “We know our business processes pretty well. [...] We can look at the existing business processes where we have some gaps or some challenges” [I4]. *Identification* was also *cross-functional*, involving business stakeholders, technical experts, domain owners, and external partners. One use case emerged from a “hackathon [...] with a mixed audience” and was selected for its “really high business relevance” [I11].

Readiness assessment captures the feasibility check behind AI opportunities. One participant linked implementation speed to “accessible data, technology, team capacity” and to questions such as: “Is there a project sponsor? Is the data ready? Are the tools available?” [I11]. A central microfoundation was *data readiness* assessment. Participants repeatedly described data as the central implementation condition – “The big topic is always the data” [I8] – and as the first feasibility check: “We always look first, like, do we have data to build this?” [I11]. Readiness also involved *technology and infrastructure readiness* assessment. Some AI technologies were described as “not mature enough yet,” [I10] requiring later refactoring, while in other cases existing tools made implementation almost “plug and play” [I12]. Finally, *organizational readiness* assessment included whether the required people were available – “are the people available?” [I2] – alongside sponsorship and team capacity.

Use case evaluation assesses whether an AI opportunity is worth pursuing. *Economic value evaluation* appeared when participants linked expected benefits to financial effects: “in the end, it is always about euros,” whether through time savings, efficiency, sales impact, or cost reduction [I2]. Where ROI was not directly quantified, organizations translated expected benefits into *strategic and business value evaluation*, using “business value points” [I2] or scoring categories such as “sales impact,” “strategic impact,” and “enabling impact” [I3]. Implementation effort was also evaluated, as pitched ideas had to specify “which teams” were needed alongside expected business value [I2]. *Performance- and efficiency-oriented evaluation* considered whether a use case reduced risks, improved user experience, generated revenue, or enabled cost reductions [I8]. These criteria were combined with technical feasibility evaluation, including whether a use case worked “with our stack” [I8]. Also, speed and reusability evaluation mattered when initiatives could be implemented quickly and create foundations for future use cases: “Speed is really, really core. And we also look at these foundational enabler elements, such as reusability” [I4].

Use case selection concerns how organizations decide which AI opportunities to pursue. A central microfoundation was *selection governance*: technical experts assessed feasibility and acted as consultants, while the final decision remained with the business. As one participant put it, “we can act as a technical consultant,” but “acceptance and choice [...] are given to the business” [I11]. Similarly, the domain owner decides “whether they want to develop the use case” [I16]. In some cases, “leadership also made the choice in the use cases” [I11], while another organization described “a kind of governance board” that qualifies incoming use cases [I8]. The decision process itself could be consequential, with one participant describing the “decision for this use case” as “an important factor” in shaping how the project unfolded [I12]. *Use case prioritization* appeared when organizations compared top candidates and asked “where is the greatest potential?” [I3]. Finally, *use case termination* was part of selection: some ideas initially seemed promising, but after reviewing the value case, “you stop it directly there” [I17].

Sensing activities therefore show that successful AI productionization begins before technical development. Organizations identify AI opportunities from business targets, process gaps, and domain needs, and then narrow them by combining business relevance, data and technology readiness, and organizational fit.

AI Project Seizing

AI Project Seizing comprises five meso-level capabilities through which organizations move from selected AI opportunities to rollout: resource mobilization and orchestration, solution option assessment, solution development, testing and validation, and organizational embedding.

Resource mobilization and orchestration describes how organizations assemble the authority, expertise, and resources needed to implement AI initiatives. *Cross-functional collaboration* appeared where business and digital teams were structurally integrated – “the business and digital teams are together [...] literally in one team reporting to the same person” [I4]. Other cases used formal coordination roles, such as an Epic Owner responsible for bringing “the people who are needed to the table” [I3]. *Partner-supported mobilization* addressed needs for additional expertise or capacity: one partner’s involvement reduced “engineering effort” and ensured “scalability” [I11], while another participant explained that “sometimes you want to bring in a consultant so that they simply create pressure” [I17]. *Leadership support and intervention* ranged from active resource mobilization – CIO support that created “a complete development team” [I1] – to passive accountability, with executives staying “in the loop” and creating “a certain seriousness among project participants” through KPI reporting and status meetings [I15]. *Funding and resource allocation* appeared through dedicated financial and personnel resources: one project was

financed through “the innovation budget” [I14], while in another, implementation depended on “the corresponding resources being available” and the team worked to secure “co-funding” [I12].

Solution option assessment concerns how organizations evaluate possible solutions and model approaches before committing to an implementation path. Technical *feasibility assessment* involved clarifying “how it is technically feasible at all” [I5]. In AI projects, this also involved *assessing model options* before selecting the better-performing one. This also includes build-or-buy decisions, with one participant emphasizing that “the build-or-buy decision naturally has to be made early” [I5]. Solution options were often assessed jointly with business stakeholders; one team “reviewed this list [...] to prioritize pilots,” with the business side already “mature in assessing feasibility” [I14].

Solution development captures how organizations translate selected AI opportunities into AI solutions. *Solution design* translated use cases into concrete personas, user stories, and backlogs. In one case, the team decided, based on “personas and use cases,” which “communication artifacts” and search content to include [I12]. In another, teams used “deep dive workshops to create the first version of the backlog” and refined user stories weekly: “We worked in an agile way [...] [the business service unit] drafted user stories, and we refined them together weekly” [I14]. Backlogs also structured prototype prioritization, as some topics were “deprioritized” during prototype development [I12]. *Prototyping* was used to test feasibility without treating early work as disposable: “First, we always do PoCs. [...] They are not wasting money. [...] It is already there, reusable” [I17]. Some teams followed a fast-fail logic, building several options in parallel and continuing only with those that worked: “You start to build 10 things. [...] Eight or six work. [...] You drop the remaining four” [I14]. *Technical configuration* involved adapting prompts, models, and technical setups to the use case, including learning “how to tell the AI tool what you want,” [I13] testing “a range of prompts,” and selecting the “best of breed” [I1]. *User involvement* took place through workshops and early stakeholder participation; customer-internal workshops were described as a “main success driver” because they created “much more clarity and understanding” and showed “how users really interact with it” [I5]. Finally, *production readiness* covered integration, infrastructure, scalability, deployment, and security work – “integration into the existing CRM platform” [I11], “setting up the infrastructure on the Azure side” [I12], and “authentication,” “security,” and “code hardening” [I5].

Testing and validation captures how organizations assess whether AI solutions perform technically, work for users, and create measurable impact. *Technical testing* included unit tests in which teams gave the agent inputs and observed whether it behaved as intended [I14]. In generative AI cases, technical testing also addressed response consistency and hallucination risks before business stakeholders approved outputs or requested modifications [I11]. *User and business testing* involved domain users in validating that the solution worked in the intended process context. During productionization, testing became more formalized through a user council in which “users from all possible areas” tested the tool [I1]. As one participant summarized, “business users have to test and contribute” [I17]. *Performance and impact testing* involved tracking usage and business impact after implementation – for example, assessing the solution “on the impact” from the commercial side [I11]. Unlike conventional software testing, validation in these AI projects extended beyond technical correctness to output plausibility, response consistency, hallucination risks, and user trust.

Organizational embedding captures how organizations move AI solutions into everyday use. *Solution communication* included internal formats such as “focus sessions” with the pilot group “where the solution was presented,” as well as external visibility through “LinkedIn” and “press” [I12]. *User enablement* was supported through “release notes,” “training topics” [I3], programs tailored to specific user groups [I14], and rollout materials prepared by business colleagues [I11]. *System integration* occurred when solutions were embedded into existing technical environments, such as a CRM platform or other internal systems [I11]. Finally, *rollout and early scaling* involved structured planning and gradual expansion: business colleagues “prepared a structural rollout plan, mapped which countries would use the tool, ensured stakeholder adoption, and designed training materials” [I11], while other cases moved from “five different users” to “100” and then “1,000 users” [I5], or rolled out “area by area” [I6].

Seizing activities show that the transition from pilot to production depends on more than building a technically functioning model. Organizations had to orchestrate cross-functional resources, assess solution options, iteratively develop and validate AI solutions with users, and prepare them for production through integration, infrastructure, security, workflow embedding, and structured rollout.

AI Project Transforming

AI Project Transforming comprises three meso-level capabilities converting project work into durable capabilities: operational AI governance, knowledge institutionalization, and AI asset evolution.

Operational AI governance concerns the structures and routines needed to operate AI systems after development. A key microfoundation was *AI performance monitoring*, including model quality, usage, and user trust over time. In one case, weekly indicators were used to “measure the health of the AI model” and check whether operation had become “disturbed” [I9]. Other cases monitored user behavior (for example, reports showing “unique users” and “types of questions asked” [I10]) or used formal evaluation setups, including a “multi-stage evaluation framework” with “15 different metrics” [I4]. One team tracked live KPIs including “how many predictions materialized per day” and “whether the prediction had been generated in time”; when performance fell below threshold, the team investigated root causes — in extreme cases concluding “good attempt, but let’s shut it down” [I2]. *Responsibility definition* concerned accountability for AI outputs, model operation, maintenance, and related data flows. In one case, output accountability remained with users because “you never have 100% certainty” and the AI module was intended to “extend the process and not completely automate it” [I1]. In another, responsibility first had to be negotiated: was it even “desired that we [...] make ML predictions or whether that should sit somewhere else entirely?” Once clarified, model operation remained with the AI team: “the model operation stays with us” [I2]. Maintenance responsibility extended beyond the model itself to “the data flows to the left and right” [I8] with data drift and model drift explicitly assigned to the AI team [I9]. Effective maintenance required the responsible team to have “an idea of how the case is built” [I8]. *AI compliance governance* involved security, legal, and data-related checks before or during operation: “every project goes through a governance and security check” [I10]. Another organization had a “Data and AI Governance Board” of legal and domain experts that evaluates use cases [I8]. Compliance could also be tool-specific, with prior legal alignment and dedicated assessments when data were stored externally rather than in-house [I2].

Knowledge institutionalization captures how organizations make AI-related knowledge available beyond the initial project team. *AI competence development* appeared in dedicated structures such as “an extra innovation team,” “an internal delivery team,” and “a data governance team,” alongside steering mechanisms defining “needed capabilities” [I17]. Central teams helped spread “know-how across business domains” [I4], while some organizations built joint Centers of Excellence or an “AI factory” with internal and partner-side data scientists and engineers [I8]. *Business-side AI enablement* addressed domain stakeholders’ knowledge gaps. In one case, stakeholders “didn’t understand AI capabilities” and assumed the AI team “could do anything” [I10]. Enablement involved expectation-setting, education on “how AI works” [I10], and learning to instruct AI tools, such as “teaching how to read a contract” [I13]. Organizations supported this through training, communities, central wikis, Q&A formats, “facts and features,” and guidance on moving “a use case from beginning to end” [I1; I6]. *Knowledge codification* took the form of playbooks, wikis, and handover artifacts. One case transferred “extensive project documentation” to a new team [I11], while another developed a “playbook” for future AI-enabled manufacturing use cases [I9]. Codification was sometimes incomplete: beyond the wiki, knowledge remained partly informal and was shared through one-to-one exchange and “connecting the dots” between teams [I1].

AI asset evolution captures how organizations turn individual AI projects into reusable and continuously evolving technical foundations. *Continuous development* appeared where AI tools were treated as evolving systems, not finished products. One team established “a system for testing models against each other, optimizing parameters, and retraining them over time” [I2]. Participants described AI solutions as “not a static environment” [I4], with teams “rolling out new features” weekly [I4]: “at every step, something has happened and we’ve added more — we started with an AI tool that just read a contract, but we’ve grown the tool into what we have today” [I13] developing solutions through shared backlogs [I12], and refactoring tools when technologies changed — for example, moving toward a low-code architecture [I8]. *Technical asset reuse and platform building* emerged as organizations created reusable components, platforms, and shared infrastructure for future AI use cases. One participant described building “a dedicated platform [...] first for GenAI and now also for agentic AI” [I17], while another organization had a centralized unit responsible for AI infrastructure, with business use cases remaining in the domains [I4]. Reuse also occurred at the component level: one case reused a research-oriented tech stack by adapting prompts and surrounding components [I1], and another used a generic approach designed to be “capable of more use cases” [I13]. In RAG-related projects, reusable assets included upload interfaces, vector storage, processing

components, connectors, and chat interfaces, with improvements flowing “back into the asset” [15]. One participant estimated that “75 percent of the tech stack” could be reused [15]; others pointed to “reusable scripts” [19] and internal AI platforms that provided “more flexibility” for new solutions [11].

Transforming activities extend productionization beyond deployment. Successful projects created routines for operating and governing AI systems, developed business-side AI competence, institutionalized AI knowledge, and converted project-level technical work into reusable assets and platforms for future initiatives. In sum, these RQ1 findings identify the activities and capabilities through which AI productionization is enacted, but they do not yet specify how organizations distribute responsibility for them. Because many activities cut across technical and domain expertise, we next examine how responsibility is allocated across piloting and productionizing.

Responsibility Arrangements for Enterprise AI Projects

To address RQ2, we develop a two-dimensional framework characterizing the allocation of responsibility across the piloting and productionizing phases (Figure 2). Each axis ranges from *tech-only* (responsibility held entirely by technical functions such as central IT, data science, AI/ML engineering, or AI center-of-excellence units) to *domain-only*, where responsibility is held entirely by business or domain functions such as process owners, end-user departments, and functional areas including legal, controlling, or supply chain. Intermediate positions reflect *tech-led*, *balanced*, and *domain-led* arrangements (see Section 3 for the full coding rubric). Embedded AI experts within business functions were treated as domain-side actors because their formal accountability was anchored in their respective functions. Capturing piloting and productionizing on separate axes lets us see whether responsibility shifts between phases. Piloting centers on identifying, building, and evaluating a candidate solution; productionizing centers on operating, sustaining, and scaling it.

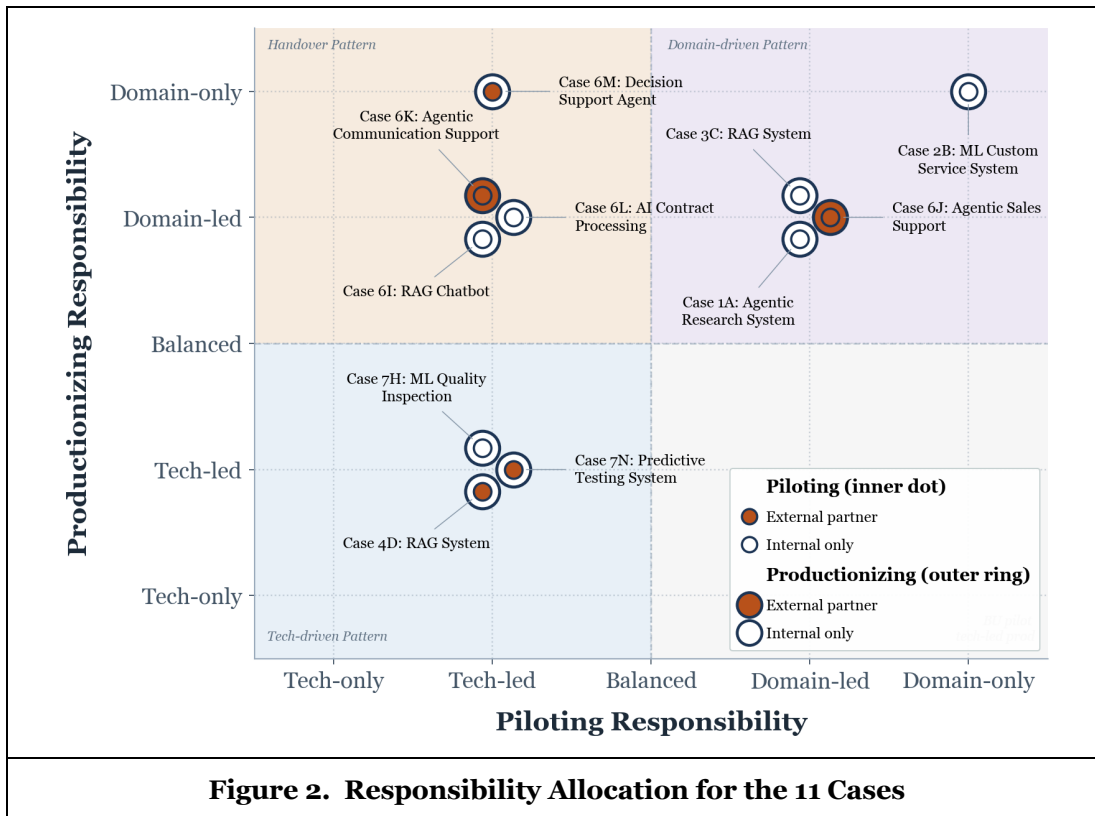


Figure 2 maps the 11 cases onto the framework. The figure additionally encodes external partner involvement during piloting (inner dot) and productionizing (outer ring), with orange indicating partner involvement and white indicating internal development. Full classification rationale for all 11 cases is available at GitHub¹. Three regions hold cases:

- **Domain-driven** (*upper-right*): domain functions held primary responsibility in both phases. Cases 1A (Agentic Research System), 2B (ML Custom Service System), 3C (RAG System), and 6J (Agentic Sales Support).
- **Handover** (*upper-left*): tech functions led the piloting phase, with responsibility shifting to domain in productionizing. Cases 6I (RAG Chatbot), 6K (Agentic Communication Support), 6L (AI Contract Processing), and 6M (Decision Support Agent).
- **Tech-driven** (*lower-left*): tech functions held primary responsibility in both phases. Cases 4D (RAG System), 7H (ML Quality Inspection), and 7N (Predictive Testing System).

The lower-right quadrant (domain-led piloting followed by tech-led productionizing) contains no cases in our sample. Partner-supported and internally developed projects appear in all three regions that hold cases. In the domain-driven quadrant, for example, four projects are situated, of which only one (Case 6J) involved an external partner; the other three reached production without partner support. External partner involvement therefore does not align with any single responsibility pattern, suggesting that the decision to engage external partners is at least partly decoupled from how internal responsibility is allocated between technical and domain functions.

Vignette Cases

To illustrate how the three responsibility patterns operate across piloting and productionizing, we selected one vignette case per pattern. The vignettes are not the basis on which the patterns were derived; they are designed to show how responsibility allocation, phase transitions, and selected microfoundations and meso-level capabilities become visible in concrete project contexts. Vignette selection followed two criteria: each case had to be clearly positioned within its area of the responsibility framework and provide sufficiently rich interview material across both phases.

Vignette 1. Domain-driven Pattern – Case 1A: Agentic Research System

Project idea. The customer briefing agent generates structured profiles of external counterparts ahead of executive customer meetings, consolidating publicly available information about a person, organization, role, and recent events for use by customer-facing managers and account executives.

Piloting. A customer-facing manager approached the embedded AI team – “is there anything where we could use AI?” – under the constraint of “fewer people, [same] work.” The AI team built a first version “quick and dirty,” and by the end of the pilot it was already in active operational use, reducing an eight-hour task to “15 seconds” with output judged “at least as good” as the account teams’ manual work.

Productionizing. Executive backing made the case for scaling – “this has to be standardized” – and the prototype was transferred to IT, where a complete engineering team rebuilt it as the AI backend of an existing internal platform. Output accountability remained explicit on the domain side: “the account teams themselves are responsible for what this AI module produces.”

We classify Case 1A as *domain-driven* because responsibility for the use case remained anchored on the domain side across both phases. During piloting, the domain initiated the opportunity, framed the business problem, and evaluated the prototype’s usefulness, while the AI team, embedded in the domain, supported rapid prototyping. During productionizing, IT took over technical rebuilding and integration, but domain actors continued to validate the solution, shape its use, and retain accountability for AI-generated outputs. A technical handover to IT does not necessarily imply a shift in overall use case ownership.

¹ <https://github.com/Ida830/From-Pilot-to-Production-Microfoundations-and-Responsibility-Patterns-for-Enterprise-AI-Projects.git>

Vignette 2. Handover Pattern – Case 6M: Decision Support Agent

Project idea. The agent automates the reasoning step in customer requests to extend invoice due dates, evaluating each request against predefined acceptance criteria within a business services unit.

Piloting. The use case was prioritized through a shared backlog, with the business side described as “mature in assessing feasibility,” but project leadership and technical responsibility remained with central IT – “the lead was always on the [central IT] side, typically the most senior agentic engineer.” The business services unit drafted user stories that were “refined together weekly,” while the product owner retained prioritization authority. Resource mobilization included an external partner architect, and validation combined technical, business, and user testing.

Productionizing. Operational ownership shifted to the business services unit, with central IT becoming “a kind of third-level support.” User training was delivered by the domain itself; reusable components were developed for future agent use cases. The embedded engineering model was described as critical for sustainability of the AI system: “if we find an embedded team from the beginning, that’s the best scenario because we know the solution will stay there.” (I14)

We classify Case 6M as a *handover* pattern because responsibility shifted from technical to domain actors across the two phases. During piloting, the domain contributed prioritization, process knowledge, user stories, and validation, while central IT retained project ownership and build responsibility. During productionizing, the business services unit assumed responsibility for rollout, user enablement, day-to-day operation, and first-line ownership, with IT remaining as third-level support and platform provider.

Vignette 3. Tech-driven Pattern – Case 7N: Predictive Testing System

Project idea. The system predicts the outcomes of physical component tests, enabling low-potential materials or product variants to be deprioritized and reducing testing effort, cost, and time-to-market.

Piloting. The opportunity originated in the business context – from an operational sales colleague’s personal interest in AI – but project management and build responsibility were consolidated in the central AI and Data Science team after an external innovation challenge: “we had project management and therefore responsibility largely on the IT side.” Business pull was initially limited (“no real noticeable pull from the business”), and cross-functional collaboration emerged only gradually.

Productionizing. A working pilot did not automatically lead to use; the team recalled asking, “why is no one using it?” Stronger operational governance followed, with KPIs for prediction quality and a project manager appointed on the business side. Enablement shifted toward the domain – “the business has to take over enablement” – but the live system remained classified as a “Run and Maintain” project under continued IT ownership. (I15)

We classify Case 7N as *tech-driven* because responsibility did not substantially shift from technical to domain actors across the two phases. During piloting, the business context provided the initial opportunity, but project management, development direction, and build responsibility were concentrated in the central AI and Data Science team. During productionizing, the domain became more involved in adoption, enablement, and role clarification, yet technical actors retained responsibility for monitoring, maintaining, and continuously developing the live AI system.

Discussion and Conclusion

Our study set out to characterize the organizational practice of AI projects that successfully transition from pilot to production, asking which activities and capabilities recur across the project lifecycle (RQ1) and how responsibility for those activities is allocated and shifted across phases (RQ2). We answer RQ1 by identifying 38 microfoundations and 12 meso-level capabilities across sensing, seizing, and transforming. Some of these resonate with general digital transformation microfoundations (Warner & Wäger, 2019); others, especially those concentrated in the transforming dimension, appear distinctive to AI and reflect the operational demands of probabilistic outputs, data-coupled behavior, and reusable infrastructure across use cases. We answer RQ2 by mapping our cases onto a two-dimensional framework that captures how responsibility is allocated separately during piloting and productionizing. Three patterns emerge – *domain-driven*, *handover*, and *tech-driven* – each a different structural arrangement in which

responsibility either remains stable across phases (domain-driven, tech-driven) or shifts mid-project (handover). The location and depth of AI competence within the firm appear to constrain which arrangement is sustainable.

Our findings refine three streams of work. AI governance research has characterized responsibility arrangements as broadly applicable across the AI lifecycle (Mäntymäki et al., 2022; Papagiannidis et al., 2025; Schneider et al., 2023); we surface phase as a structural dimension, showing that arrangements can change mid-project and that whether they do is itself part of the design. Organizational design typologies distinguish centralized, embedded, and hybrid models of AI work (Davenport & Ronanki, 2018; Fountaine et al., 2019); our patterns connect to these forms but operate at the project level. Dynamic capabilities work on digital transformation has theorized at the firm level (Warner & Wäger, 2019); our microfoundations sit at the meso-level, capturing the cross-functional team-level coordination through which AI productionization is enacted.

Theoretical Implications

Three implications follow. First, AI productionization requires a distinct microfoundation profile in the transforming dimension. On the one hand, sensing and seizing activities largely echo general digital transformation microfoundations. For instance, our use case selection and readiness assessment microfoundations resonate with Warner and Wäger's (2019) digital scouting and digital scenario planning, and our solution development resonates with their rapid prototyping. This similarity is itself informative: rather than indicating a lack of AI specificity, it suggests that firms with established processes for sensing and seizing digital opportunities are well positioned to do the same for AI, and it locates the distinctive break after deployment rather than at the front end of the project. Even so, we argue that readiness-related microfoundations carry more weight in AI projects: data readiness, for example, is a precondition for prototyping because AI must learn from existing data, whereas conventional IT systems often generate their data only during use (e.g., transactions in an ERP system).

On the other hand, transforming activities diverge systematically. Where Warner and Wäger (2019) name navigating innovation ecosystems, redesigning internal structures, and improving digital maturity as the transforming microfoundations of incumbent digital transformation, our cases cluster around operational AI governance, knowledge institutionalization, and AI asset evolution. These capabilities respond to AI's distinctive properties (Benbya et al., 2021; Berente et al., 2021): probabilistic outputs that require ongoing performance monitoring beyond one-off testing, data-coupled behavior that drifts and requires continuous maintenance (Weber et al., 2023), and reusable AI components that warrant handover-enabling investments (such as development platforms) as firms productionize multiple use cases. Specifically, two microfoundations make this concrete: performance monitoring differs in kind from conventional IT operations, as an AI system's performance shifts with concept drift and with how competently users interact with it (Fügener et al., 2021; Ren & Clement, 2024), whereas deterministic IT systems do not change their behavior in response to the data and input they receive; AI competence on the business side directly affects how well a productionized system performs because user interaction feeds back into performance.

Second, responsibility allocation is a phase-sensitive structural variable, not a steady-state governance property. This refines AI governance accounts that treat structural responsibility arrangements as broadly applicable across the AI project lifecycle (Papagiannidis et al., 2025). Arrangements can remain stable, shift, or invert as a project matures, and the configuration a project starts with likely constrains the transitions later available. The handover pattern is the clearest case: a deliberately sequenced shift in which technical functions lead piloting precisely to enable a domain-led operational phase. This shift does not occur automatically; it appears to require anticipatory work during piloting (e.g., platform-building, documentation, embedded team design) specifically to enable a viable handover.

Third, responsibility allocation is a project-level choice conditioned by AI competence distribution. Our analysis shows that the same firm may run multiple patterns in parallel; the centralized, embedded, and hybrid models that firm-level taxonomies describe (Davenport & Ronanki, 2018; Fountaine et al., 2019) can co-exist within one firm across different projects. Which of those patterns are sustainable depends on how AI competence is distributed: domain-driven projects feature competence on the business side, tech-driven projects rely on centralized technical functions with limited business-side competence, and handover projects need platform-building competence on the technical side and operational competence on the domain side. This complements firm-level AI maturity model accounts (Hansen et al., 2024) by

operating at a different unit of analysis: at the project level, one firm in our sample hosted two patterns across different projects, suggesting that pattern choice depends on where AI competence sits across functions rather than on the firm's overall maturity stage.

Managerial Implications

Our results suggest three forms of guidance for practitioners, structured around questions teams should ask at each stage. Successful projects in our cases consistently originated from articulated business needs, process gaps, or domain pain points; managers should anchor AI opportunity identification in such problems, confirm data and technology readiness, and involve technical functions in feasibility assessment, rather than acquiring AI capabilities first and searching for use cases afterward. Where a handover is planned, the platform-building work the technical side must do during piloting deserves explicit attention: post-handover operation is feasible only when reusable components, codified documentation, and embedded team arrangements are built before rollout. Patterns operate at the project level rather than the firm level: the same firm may run multiple patterns across its AI portfolio, making uniform governance templates inappropriate. Managers may benefit from asking the following questions to facilitate transition of AI projects from pilot to production: Does the AI project address a specific business problem or process gap? Is there someone in the domain who understands the solution and can take over responsibility? And are reusable components and documentation in place that facilitate responsibility handover?

Limitations and Future Research

Our findings must be read against several limitations. First, all cases reached or approached production, and candidates were partly identified through their public communication, which favors projects firms are willing to showcase. The resulting survivorship bias means that the microfoundations we identify may also occur in failed projects; our design cannot isolate which of them differentiate success from failure. And because our research question concerns the transition from pilot to production, the design deliberately conditions on a pilot having existed; whether a given pilot should have been approved in the first place is an upstream question our data cannot answer. Future research should therefore include failed or abandoned AI projects – including pilots that were stopped or never approved – to identify the microfoundations that separate successful from failed productionization.

Second, we identify microfoundations across cases without examining whether they are enacted differently across responsibility arrangements. Domain-driven, handover, and tech-driven projects may perform activities such as readiness assessment, solution development, or AI compliance governance in systematically different ways, and some microfoundations may be specific to particular arrangements while others are common across all three. With three to four cases per arrangement, our sample is too small to test such linkages systematically, so the connection between our two contributions remains exploratory. Future research with a larger or more targeted sample that compares microfoundations across the three patterns would directly link our two contributions and test whether the microfoundations we identify are general or pattern-specific.

Third, we treat AI competence distribution as a candidate structural precondition without tracing how business-side AI competence accrues. Whether it is hired in, transferred from technical functions, or accumulated through prior projects is unclear, and business-side actors must do more than be generally AI-literate: they must assess feasibility, judge data suitability, and validate AI outputs against process requirements. Future work should investigate both how competence accrues and which mechanisms (e.g., embedded technical staff, structured training programs, or rotation between technical and domain functions) enable substantive engagement beyond surface-level familiarity.

Fourth, our patterns rest on a small number of cross-sectional cases concentrated in large incumbent firms; with three to four cases per pattern, we cannot establish whether the patterns generalize or whether additional configurations exist that our sample did not surface. We aim for analytical rather than statistical generalization, extending our findings to theoretical propositions about the mechanisms of AI productionization instead of to a population of AI projects. Most cases also rely on a single informant, so responsibility arrangements are reconstructed from one vantage point per project. Larger samples would test generalizability, while longitudinal designs would show whether responsibility arrangements remain stable over a project's lifetime.

Finally, our microfoundations capture current organizational practices in a rapidly evolving field. Spanning multiple AI types at different levels of technological maturity, they respond to structural properties of AI systems, such as probabilistic outputs, data-coupled behavior, and opacity (Benbya et al., 2021; Berente et al., 2021), rather than to specific technologies, suggesting robustness to technological change. How these microfoundations evolve as organizations and AI technology mature, and whether those tied to current limitations will remain as central, is an open question for future research.

References

- Adner, R., & Helfat, C. E. (2003). Corporate effects and dynamic managerial capabilities. *Strategic Management Journal*, 24(10), 1011–1025. <https://doi.org/10.1002/smj.331>
- Ångström, R. C., Björn, M., Dahlander, L., Mähring, M., & Wallin, M. W. (2023). Getting AI Implementation Right: Insights from a Global Survey. *California Management Review*, 66(1), 5–22. <https://doi.org/10.1177/00081256231190430>
- Benbya, H., Pachidi, S., & Jarvenpaa, S. L. (2021). Special Issue Editorial: Artificial Intelligence in Organizations: Implications for Information Systems Research. *Journal of the Association for Information Systems*, 22(2), 281–303. <https://doi.org/10.17705/1jais.00662>
- Berente, N., Gu, B., Recker, J., & Santhanam, R. (2021). Managing Artificial Intelligence. *MIS Quarterly*, 45(3), 1433–1450. <https://doi.org/10.25300/MISQ/2021/16274>
- Chirumalla, K. (2021). Building digitally-enabled process innovation in the process industries: A dynamic capabilities approach. *Technovation*, 105, Article 102256. <https://doi.org/10.1016/j.technovation.2021.102256>
- Corbin, J., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13, 3–21. <https://doi.org/10.1007/BF00988593>
- Davenport, T. H., & Ronanki, R. (2018). Artificial Intelligence for the Real World. *Harvard Business Review*, 96(1), 108–116.
- der Weduwe, R. (2026). Deloitte report: AI is shifting from pilot and experimentation to scaling. <https://www.deloitte.com/nl/en/about/press-room/state-of-ai-in-the-enterprise-2026.html>
- Dzhusupova, R., Bosch, J., & Olsson, H. H. (2024). Choosing the right path for AI integration in engineering companies: A strategic guide. *Journal of Systems and Software*, 210, Article 111945. <https://doi.org/10.1016/j.jss.2023.111945>
- Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management Review*, 14(4), 532–550.
- Enholm, I. M., Papagiannidis, E., Mikalef, P., & Krogstie, J. (2022). Artificial Intelligence and Business Value: A Literature Review. *Information Systems Frontiers*, 24(5), 1709–1734. <https://doi.org/10.1007/s10796-021-10186-w>
- Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, 66(1), 111–126. <https://doi.org/10.1007/s12599-023-00834-7>
- Fountain, T., McCarthy, B., & Saleh, T. (2019). Building the AI-Powered Organization. *Harvard Business Review*, 97(4), 62–73.
- Fügener, A., Grahl, J., Gupta, A., & Ketter, W. (2021). Will Humans-in-the-Loop Become Borgs? Merits and Pitfalls of Working with AI. *MIS Quarterly*, 45(3), 1527–1556. <https://doi.org/10.25300/MISQ/2021/16553>
- Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking Qualitative Rigor in Inductive Research: Notes on the Gioia Methodology. *Organizational Research Methods*, 16(1), 15–31. <https://doi.org/10.1177/1094428112452151>
- Haefner, N., Parida, V., Gassmann, O., & Wincent, J. (2023). Implementing and scaling artificial intelligence: A review, framework, and research agenda. *Technological Forecasting and Social Change*, 197, Article 122878. <https://doi.org/10.1016/j.techfore.2023.122878>
- Hansen, H. F., Lillesund, E., Mikalef, P., & Altwaijry, N. (2024). Understanding Artificial Intelligence Diffusion through an AI Capability Maturity Model. *Information Systems Frontiers*, 26(6), 2147–2163. <https://doi.org/10.1007/s10796-024-10528-4>
- Hutzschenreuter, T., Lämmermann, T., Sake, A., & Ludwig, H. (2025). Successfully Mitigating AI Management Risks to Scale AI Globally. *MIS Quarterly Executive*, 24(3), 221–238. <https://doi.org/10.17705/2msqe.00117>

- Jöhnk, J., Weißert, M., & Wyrski, K. (2021). Ready or Not, AI Comes—An Interview Study of Organizational AI Readiness Factors. *Business & Information Systems Engineering*, 63(1), 5–20. <https://doi.org/10.1007/s12599-020-00676-7>
- Krakowski, S., Luger, J., & Raisch, S. (2023). Artificial intelligence and the changing sources of competitive advantage. *Strategic Management Journal*, 44(6), 1425–1452. <https://doi.org/10.1002/smj.3387>
- Mäntymäki, M., Minkinen, M., Birkstedt, T., & Viljanen, M. (2022). Defining organizational AI governance. *AI and Ethics*, 2(4), 603–609. <https://doi.org/10.1007/s43681-022-00143-x>
- Mikalef, P., & Gupta, M. (2021). Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. *Information & Management*, 58(3), 103434. <https://doi.org/10.1016/j.im.2021.103434>
- Papagiannidis, E., Mikalef, P., & Conboy, K. (2025). Responsible artificial intelligence governance: A review and research framework. *The Journal of Strategic Information Systems*, 34(2), 101885. <https://doi.org/10.1016/j.jsis.2024.101885>
- Peretz-Andersson, E., Tabares, S., Mikalef, P., & Parida, V. (2024). Artificial intelligence implementation in manufacturing SMEs: A resource orchestration approach. *International Journal of Information Management*, 77, 102781. <https://doi.org/10.1016/j.ijinfomgt.2024.102781>
- Raisch, S., & Krakowski, S. (2021). Artificial Intelligence and Management: The Automation–Augmentation Paradox. *Academy of Management Review*, 46(1), 192–210. <https://doi.org/10.5465/amr.2018.0072>
- Ren, Y., & Clement, J. (2024). Augmenting Human Teams with Robots in Knowledge Work Settings: Insights from the Literature. *J. Hum.-Robot Interact.*, 13(2), 20:1–20:34. <https://doi.org/10.1145/3649884>
- Salvato, C., & Vassolo, R. (2018). The sources of dynamism in dynamic capabilities. *Strategic Management Journal*, 39(6), 1728–1752. <https://doi.org/10.1002/smj.2703>
- Schneider, J., Abraham, R., Meske, C., & vom Brocke, J. (2023). Artificial Intelligence Governance For Businesses. *Information Systems Management*, 40(3), 229–249. <https://doi.org/10.1080/10580530.2022.2085825>
- Schneider, M. H. G., Kanbach, D., Kraus, S., & Dabić, M. (2024). Transform Me If You Can: Leveraging Dynamic Capabilities to Manage Digital Transformation. *IEEE Transactions on Engineering Management*, 71, 9094–9108. <https://doi.org/10.1109/TEM.2023.3319406>
- Sharma, A. J., Rathore, B., Biswas, B., Xiao, X., & Gupta, R. (2026). Generative AI Implementation in Enterprises: Lessons From a Case Study of Enhanced IT Service Management. *Information Systems Journal*, 36(5), 661–690. <https://doi.org/10.1111/isj.70029>
- Simón, C., Revilla, E., & Sáenz, M. J. (2024). Integrating AI in organizations for value creation through Human-AI teaming: A dynamic-capabilities approach. *Journal of Business Research*, 182, 114783. <https://doi.org/10.1016/j.jbusres.2024.114783>
- Teece, D. J. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal*, 28(13), 1319–1350. <https://doi.org/10.1002/smj.640>
- Teece, D. J., & Linden, G. (2017). Business models, value capture, and the digital enterprise. *Journal of Organization Design*, 6(1), Article 8. <https://doi.org/10.1186/s41469-017-0018-x>
- Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic Capabilities and Strategic Management. *Strategic Management Journal*, 18(7), 509–533.
- van den Broek, E., Sergeeva, A., & Huysman, M. (2021). When the Machine Meets the Expert: An Ethnography of Developing AI for Hiring. *MIS Quarterly*, 45(3), 1557–1580. <https://doi.org/10.25300/MISQ/2021/16559>
- Warner, K. S. R., & Wäger, M. (2019). Building dynamic capabilities for digital transformation: An ongoing process of strategic renewal. *Long Range Planning*, 52(3), 326–349. <https://doi.org/10.1016/j.lrp.2018.12.001>
- Weber, M., Engert, M., Schaffer, N., Weking, J., & Krcmar, H. (2023). Organizational Capabilities for AI Implementation—Coping with Inscrutability and Data Dependency in AI. *Information Systems Frontiers*, 25(4), 1549–1569. <https://doi.org/10.1007/s10796-022-10297-y>