

MAS ETH MTEC

Master's Thesis

Rodrigo Mendoza

Innovating With Constraints:

*Leveraging Generative AI to Enable Digital Venturing in Resource-Constrained,
Non-Technology-Oriented Small Enterprises*

Innovating With Constraints: *Leveraging Generative AI to Enable Digital Venturing in Resource-Constrained, Non-Technology-Oriented Small Enterprises*

Rodrigo Mendoza Diaz - MAS MTEC 2024–2026

2026-07-11

Table of Contents

Abstract	3
1.0 Introduction	4
1.1 Background	4
1.2 Theoretical Framework	4
1.3 Problem Statement	5
1.4 Research Questions	6
2.0 Methodology	7
2.1 Research Method: Action Design Research (ADR)	7
2.2 Data Collection	7
2.3 The Case of Durable Planung und Beratung GmbH	9
3.0 Findings	11
3.1 Introduction to Generative AI	11
3.1.1 What Generative AI Is and Why It Matters for a Constrained Firm	11
3.1.2 AI Coding Agents: The Tool Class of This Thesis	13
3.1.3 The Human-AI Sociotechnical System	13
3.1.4 SMEs and Generative AI Adoption	14
3.1.5 Results: Limits of AI Coding Tools	14
3.1.6 Stage 01: Design Principles Part I	15
3.2 Stage 01: Problem Formulation	17
3.2.1 Structure of the Literature Review	17
3.2.2 Effectuation as a Lens for Innovation	18
3.2.3 Bricolage: Creating Something from Nothing	19
3.2.4 The Relationship Between Effectuation, Causation, and Bricolage	20
3.2.5 First Conclusions and Boundary Conditions	21
3.2.6 Open Innovation	22
3.2.7 Absorptive Capacity: Preparing for Generative AI	24
3.2.8 Results: Innovation Framework	25
3.2.9 Stage 01: Design Principles Part II	26
3.3 Stage 02: BIE, Sub-cycle A: Resource Map and Prototype Specification	29
3.3.1 Analytical Framework for Resource Mapping	29
3.3.2 Resource Mapping Workshop	30
3.3.3 Main findings from the Workshop:	33
3.3.4 Critical Reflection	34
3.3.5 Sub-cycle A: Design Principles	34
3.4 Stage 02: BIE, Sub-cycle B: IT Prototype	36

3.4.1 Working definition of a Database	36
3.4.2 DUR Data Landscape	36
3.4.3 Data Basis: Sources, Selection, and Cleaning	37
3.4.4 The Prototype: finDUR 0.1	38
3.4.5 Building finDUR 0.1: Process and Decisions	41
3.4.6 The Role of GenAI During the Process	41
3.4.7 Management Evaluation: The Second Management Workshop	42
3.4.8 External Technical Evaluation: Software-Engineering Review	43
3.4.9 Sub-cycle B: Design Principles	45
3.5 Stage 3: Reflection and Learning	46
3.5.1 Working Under Constraint	46
3.5.2 What the Build Showed	46
3.5.3 What the Evaluations Showed	47
3.5.4 Limitations	48
3.5.5 Design Principles Stage 3: Reflection and Learning	48
4.0 Stage 4: Formalization of Learning	50
4.1 Answer to the First Research Question	50
4.2 Answer to the Second Research Question	51
4.3 Design Principles	52
5.0 Recommendations and Outlook	54
5.1 General Remarks for Durable	54
5.2.1 Recommendations on finDUR and a Database	54
5.2.2 Profile of Future Hires and Gatekeepers	56
5.2.3 Other Approaches: The Hackathon Proposal	58
5.3 Directions for Further Research	58
6.0 Closing Note	60
Acknowledgements	60
Declaration on the Use of Generative AI	60
7.0 References	61

Abstract

For resource-constrained, non-technology-oriented SMEs, capabilities such as software engineering, data science, software design, and broader digital literacy are often missing. These firms may therefore lack the resources and expertise needed to pursue digital venturing. Recent advances in Generative AI are presented as capable of lowering these barriers. This thesis examined to what extent Generative AI can compensate for missing organizational expertise and how it can support digital venturing under resource constraints. The problem was analysed through effectuation theory (Sarasvathy 2001) and bricolage (Baker and Nelson 2005), which explain how firms can act under uncertainty by using and recombining the means already available to them.

The study followed the Action Design Research method and completed one full cycle within a sustainability consultancy in the AEC industry. The intervention produced an IT prototype developed by a non-expert operator using AI coding agents in approximately eight working days. The result was evaluated through internal management feedback and an external software-engineering review.

Generative AI compensated for execution-level tasks, including script writing, data structuring, and debugging, but not for judgement, source authority, or validation. These remained human responsibilities and depended on prior domain and technical knowledge. The extent to which Generative AI can compensate for missing expertise therefore depends on the firm's ability to absorb and use its outputs. In this case, the technology was not the binding constraint; the operator and the organization reached their limits first.

By combining a literature review, resource mapping, prototype development, and evaluation, the study contributes to the emerging research on Generative AI and innovation in SMEs. Further research is required to test whether the resulting design principles apply in other organizational contexts.

1.0 Introduction

1.1 Background

Innovation involves introducing new goods, new production methods, new markets, new supply sources, or new forms of organization into the economy (Schumpeter 1934). Actors pursuing innovation must therefore develop novel solutions, implement them, and attempt to capture value from them. In its simplest abstraction, an actor pursuing an innovation project must decide how to allocate capital (Holmström 1989). Because innovation implies introducing something novel and capturing value from it, neither successful development nor market success is guaranteed. Innovation therefore carries irreducible uncertainty and the possibility of failure (McGrath 1999). By definition, then, innovation requires both the commitment of resources, whether time, capital, or organizational capacity, and the acceptance of failure. How much, in what form, and at what level of risk depends on the situation, but some minimum resource commitment is necessary to initiate any innovation process.

Large corporations can invest significant capital in research and development, and their scale allows them to absorb failure, run parallel projects, and attract specialized talent. Yet abundant resources do not guarantee successful outcomes. Bureaucratic structures, reputation concerns, and capital market pressures can push large firms toward conservative decisions (Holmström 1989), and too much slack can itself become an obstacle (Nohria and Gulati 1996). At the other end of the spectrum, startups benefit from full commitment to the venture, freedom to experiment, and no legacy business to protect (Holmström 1989). Resources may be misallocated and goals may drift, but the primary organizational objective remains innovation. Both corporations and startups have advantages that allow them to innovate freely.

Small and medium-sized enterprises sit in neither position. In Switzerland, SMEs are defined as firms with fewer than 250 employees and account for 99.7% of all market-economy businesses, generating 66.4% of total employment (Federal Statistical Office 2024). In the OECD area, SMEs average more than 99% of all businesses (OECD 2024). This thesis focuses specifically on small enterprises with ten to forty-nine employees (Federal Statistical Office 2024). Drawing on Holmström (1989), Brunswicker and Vanhaverbeke (2015), and De Massis et al. (2018), this thesis argues that established small enterprises occupy a structurally distinct position that existing frameworks do not fully address. At this scale, firms lack the financial and human capital of large corporations and the freedom and risk tolerance of startups. Having established a market presence, they must deploy their full resource base to remain competitive in their existing business. Any reallocation toward a new venture directly reduces the resources sustaining core operations. With budgets stretched to meet current demands, there is rarely surplus capacity or organizational slack available for innovation, let alone research and development. Unlike the startup, they have more to lose, which reduces affordable loss (Sarasvathy 2001). This category of firm sits in an uncomfortable middle ground where innovation resources can be more constrained than in the startup case. This thesis addresses that middle ground.

The concept of organizational slack helps explain why small enterprises, even when successful in their own markets, are more constrained than startups when it comes to innovation. Slack refers to resources that exceed the minimum necessary for current operations, including surplus employees, unused capacity, and excess capital (Cyert and March 1963). Research suggests an inverted U relationship between slack and innovation: too little produces paralysis, too much produces undisciplined experimentation (Nohria and Gulati 1996). Small enterprises sit outside the productive middle of this curve. They operate below the minimum slack threshold that sustained innovation requires. Under these conditions, innovating within the core business is already difficult. Venturing into an entirely new domain becomes almost impossible.

This raises the central question this thesis addresses. What happens when a small enterprise with limited organizational slack and no digital expertise decides to pursue a digital venture? Generative AI may provide part of the answer. Emerging interest in this technology and recent survey evidence suggest that these tools lower the cost of accessing technical expertise (OECD 2025; Barney and Reeves 2024). Whether this is sufficient to close the resource gap, or whether it merely amplifies existing advantages for firms that already possess them, remains an open question.

1.2 Theoretical Framework

Existing research on SME innovation focuses primarily on open innovation frameworks. Studies such as De Massis et al. (2018) and Brunswicker and Vanhaverbeke (2015) show that the most effective strategies for resource-

constrained SMEs involve relying on local networks, partnerships, and existing value chain relationships to compensate for internal resource constraints. This literature concentrates on firms innovating within their core business and established markets. It offers limited guidance for SMEs attempting to venture outside their known networks or familiar domain. Under these conditions, small non-technology-oriented firms appear effectively locked into their field of specialization, with digital venturing remaining outside the feasible set of possibilities.

Accessing skills such as software development, data engineering, or digital product design requires either hiring specialists or outsourcing to technology firms. Both options are out of reach for most small, resource-constrained enterprises. This seems to be changing with the public release of large-scale generative AI systems, or Large Language Models (LLMs), such as OpenAI's ChatGPT and Anthropic's Claude: tools that allow any user with a computer and a natural language prompt to generate code, structure data, draft documents, and prototype digital solutions without prior technical training. These systems are instances of a broader class of AI that produces new content by learning patterns from large datasets and generating outputs in response to natural language inputs (Feuerriegel et al. 2024). This class of technology has entered the mainstream as a potential source of accessible expertise for resource-constrained firms. Survey evidence from over 5,000 SMEs across seven OECD economies indicates that adoption is rising, but remains concentrated in peripheral tasks such as marketing and administrative support rather than core operations (OECD 2025). GenAI does not appear able to substitute for complete organizational units, and despite its code generation potential, no significant wave of digital transformation has followed in SMEs. It seems that productive value depends on the underlying Absorptive Capacity of the firm: the ability to recognize, assimilate, and apply new external knowledge to commercial ends (Cohen and Levinthal 1990).

GenAI allows anyone to generate software code without prior coding experience, which appears to lower the threshold for accessing a skill that was previously restricted to technical specialists. The cost of these tools is substantially lower than acquiring the same skills through hiring or outsourcing. The challenge is therefore not access in principle, but adoption in practice, particularly under conditions of limited digital literacy and constrained resources. In short: GenAI is accessible but implementation capacity is missing. This thesis examines whether, and if so to what extent, small enterprises with no prior digital expertise can use these tools when venturing into digital services under tight financial, organizational, and knowledge constraints.

To address this gap, this thesis returns to foundational frameworks in entrepreneurship and innovation theory. More recent approaches such as Lean Startup methodology (Ries 2011) and scientific entrepreneurship (Camuffo et al. 2020) offer valuable contributions but rest on assumptions that do not hold for the firms examined here. Lean Startup proposes iterative experimentation and a build-measure-learn cycle centered on the minimum viable product. This logic assumes the firm can build, test, and absorb the cost of repeated failure. For small enterprises with low affordable losses and no technical expertise to run experiments, the threshold to begin is already too high. Scientific entrepreneurship formalizes hypothesis-driven decision-making and raises the quality of strategic reasoning under uncertainty (Camuffo et al. 2020). It improves how entrepreneurs search for viable ideas, but it does not address what happens when the resources required to act on any hypothesis are absent.

Both frameworks assume a minimum resource base that the firms examined in this thesis do not possess. This research therefore returns to more foundational theories that begin from what the firm actually has. Causation, effectuation, and bricolage do not start from a target outcome or an experimental design. They start from the identity of the firm and its existing means, and ask what can be built from them (Sarasvathy 2001; Baker and Nelson 2005). This is the appropriate starting point for firms that cannot afford to experiment repeatedly, cannot build without assistance, and must act within the boundaries of what they already control.

1.3 Problem Statement

For non-technology-oriented SMEs, expertise such as software development, data engineering, and digital product design is often structurally absent. Existing SME innovation literature offers limited guidance for this specific condition, and research on GenAI adoption in SMEs remains nascent, with empirical evidence lagging significantly behind the pace of technological development.

To address this gap, this research investigates whether and to what extent Generative AI can compensate for missing organizational expertise and support digital venturing in a resource-constrained, non-technology-oriented SME.

1.4 Research Questions

This thesis is guided by two research questions:

1. To what extent can Generative AI compensate for missing organizational expertise required for digital venturing in a resource-constrained, non-technology-oriented Small Enterprise?
2. How can Generative AI be implemented in a non-technology-oriented Small Enterprise to support an effectual approach to digital venturing?

Together, these research questions move from problem diagnosis toward solution design and implementation. The first question examines the existence and limits of GenAI as a resource compensator in resource-constrained SMEs, while the second focuses on how such expertise can be operationalized within a concrete organizational setting. Neither question can be answered through theory alone. The first asks what GenAI can compensate for in practice, which requires observing the technology in use against real capability gaps. The second asks how implementation can proceed, which is a design question.

The limits of GenAI as a compensator are visible only at the point where a non-technical operator reaches a capability boundary during real work. A study design that observes without intervening cannot surface these boundaries. The research therefore requires building a working solution inside a constrained organization and evaluating it under operational conditions. The research problem addressed is practice-oriented.

2.0 Methodology

2.1 Research Method: Action Design Research (ADR)

Action Design Research (ADR) was selected for this thesis because the research problem is practical, not only theoretical: a resource-constrained firm faces a capability gap, a new technology may help close it, and the only way to learn whether it can is to build and test this inside the firm itself. ADR is designed for exactly this. It lets a researcher intervene in a real organizational problem, build an artifact as the proposed solution, and draw theoretical conclusions from how that artifact performs under real conditions (Sein et al. 2011). The method therefore pursues two goals at once: understanding an emerging organizational phenomenon, and building and evaluating a working solution in a real setting. ADR combines Design Science Research and Action Research, allowing the researcher to iteratively build, intervene, observe, and evaluate an artifact while it is being used in practice (Sein et al. 2011).

ADR was also selected because the artifact created in this research cannot be separated from its organizational context. The usefulness of GenAI as a resource compensator depends on the firm's existing resources, expertise gaps, workflows, and constraints. ADR explicitly treats artifacts as ensemble artifacts, meaning they are shaped by and embedded within the organizational environment in which they are developed (Sein et al. 2011). Finally, ADR supports the generation of prescriptive design knowledge (Sein et al. 2011). The thesis does not only aim to describe what happened in the specific organization, but to derive broader design principles about how GenAI can support digital venturing in resource-constrained SMEs and what underlying resources are needed for its implementation.

2.2 Data Collection

Data collection follows the four ADR stages: problem formulation, building intervention and evaluation (BIE), reflection and learning, and formalization of learning. The entire process is treated as the ensemble artifact, while the results can be understood through four interconnected outputs: an innovation framework, a resource map framework, and an IT prototype and the design principles derived from the whole cycle. Data collection is not separated from the intervention. The process does five things:

1. Helps define the IT prototype specifications
2. Builds the IT prototype
3. Documents the process
4. Evaluates where GenAI can compensate for missing organizational expertise in practice
5. Derives design principles and theoretical insights from the intervention process

The ensemble artifact is treated as a full research cycle that can later be iterated further, either to improve the IT prototype itself or to adapt it to other organizational contexts. While the actual construction of the prototype primarily occurs during Stage 2, the artifact continuously evolves and is formalized throughout the entire four-stage process. The expected outcomes of the full research cycle are:

1. An innovation framework that helps abstract and understand the innovation process and position the organization within it.
2. A resource map that allows management to identify where GenAI can realistically be deployed, under which conditions, and for which objectives.
3. An IT prototype developed with the support of GenAI and within the constraints of the organization.
4. An understanding of the minimum set of practitioner skills required to effectively guide and utilize Generative AI tools.
5. A final evaluation of the ensemble artifact and an assessment of whether the approach can be further applied, scaled, or requires revision before additional iterations.

The full research cycle was supported by two types of data. The first type was the documentation of the research itself. It includes workshop protocols, presentations, automatically generated logs from the sessions with the AI coding agents, voice-recorded reflections, and written feedback from management and an external technical expert.

The second type of data was the empirical data that feeds the prototype. This consists of the case firm's own project records. Section 3.4.2 explains this data basis in more detail, including which data was collected, where it came from, why it was selected, and how it was cleaned before being used in the prototype.

Company Case - 01

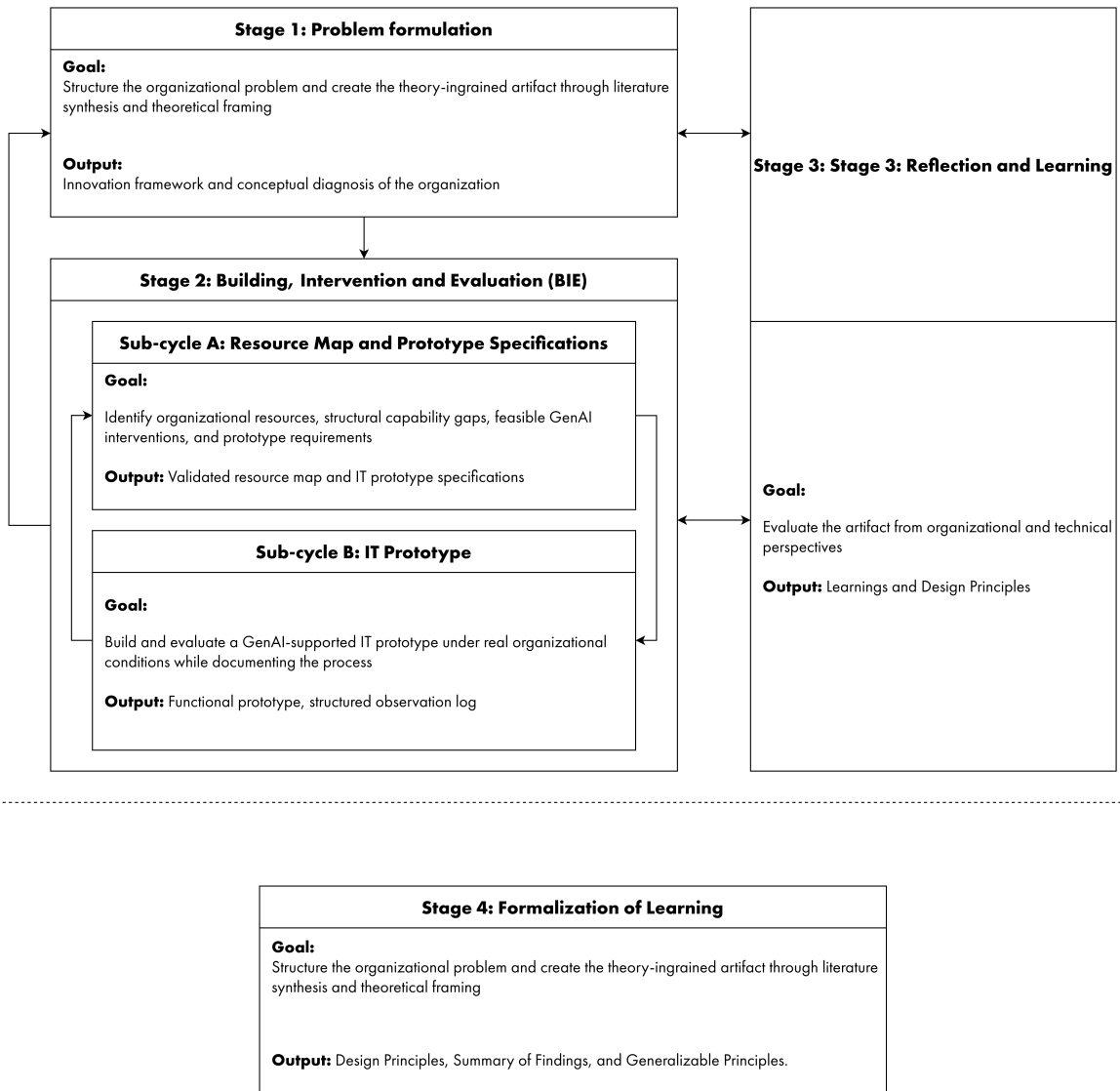


Figure 01: Action Design Research full research cycle.

Table 01: Action Design Research stages, goals, expected outputs, and participants.

Stage	Goal	Expected Output	Participants
Stage 1: Problem Formulation	Structure the organizational problem and create the theory-ingrained artifact through literature synthesis and theoretical framing	Innovation framework and conceptual diagnosis of the organization	Researcher / Practitioner
Stage 2 Sub-cycle A: Resource Map and Prototype Specifications	Identify organizational resources, structural capability gaps, feasible GenAI interventions, and prototype requirements	Validated resource map and IT prototype specifications	Researcher / Practitioner, DUR Management
Stage 2 Sub-cycle B: IT Prototype	Build and evaluate a GenAI-supported IT prototype under real organizational conditions while documenting the process	Practitioner skill profile, functional prototype, structured observation log	Researcher / Practitioner
Stage 3: Reflection and Learning	Evaluate the artifact from organizational and technical perspectives	Technical assessment and end-user assessment	DUR Management, External Technical Expert
Stage 4: Formalization of Learning	Synthesize findings into mechanisms, limitations, boundary conditions, and design principles	Theorization across the ensemble artifact, including practitioner skill profile	Researcher / Practitioner

2.3 The Case of Durable Planung und Beratung GmbH

The selected case is Durable Planung und Beratung GmbH (DUR), a Zurich-based AEC consulting firm specializing in sustainability, building physics, acoustics, certification support, and ESG-related advisory services for the built environment. Founded in 2011 by Jörg Lamster, the company operates as a highly specialized consultancy. Located in Zurich, DUR currently employs 24 people and provides services in sustainability strategies, client-side advisory, Minergie and SNBS certification guidance, integrated energy concepts, building renovation, building physics and acoustics, as well as building ecology and life-cycle assessments. The firm primarily serves project owners, developers, architects, general planners, and construction stakeholders seeking sustainability and compliance expertise across all planning and implementation phases.

In 2020, Wüest Partner Group acquired 100% of DUR, integrating the company into the broader Wüest Partner ecosystem while maintaining it as an operationally independent subsidiary. Since the acquisition, the company has grown from approximately 5 to 24 employees, reflecting the growth strategy of the group. The holding structure combines consulting, data analytics, software, and real estate technology services, while DUR itself remains primarily focused on sustainability and technical advisory work. At the same time, the company has been exposed to increasing strategic pressure to support the group's broader digitalization ambitions and growth objectives. Management has therefore committed to exploring how the firm could expand beyond traditional sustainability consulting toward more data- and information-oriented services for clients and investors during early planning phases.

DUR has no prior experience in software development, no dedicated technical staff, limited innovation resources, and no established reputation as a digital service provider. Strategic decisions of any significance require holding-level approval, limiting organizational agility and autonomous experimentation. The firm's identity is grounded in sustainability consultancy, not digital services.

At the same time, DUR is not starting from zero. The firm holds fifteen years of accumulated project data, deep domain expertise in sustainability and building performance consulting, an established professional network within the Swiss AEC sector, and indirect access to the capabilities of a holding ecosystem that includes data and software-oriented firms.

These characteristics make DUR a particularly relevant case for this research. The company represents a live or-

ganizational setting in which digital venturing efforts are pursued under real operational and resource constraints rather than within a simulated or idealized environment. The case therefore enables the research to examine not only whether Generative AI can compensate for missing digital capabilities in a non-technology-oriented SME, but also the practical limitations, organizational dependencies, and implementation challenges that shape such efforts during an ongoing innovation process.

The researcher occupies a dual position in this study. As an employee of DUR with two years of experience, he is an active member of the innovation group responsible for exploring the firm's digitalization strategy. He is therefore not an external observer but a practitioner embedded in the same organizational context the research examines. This dual role shapes both the research design and the nature of the empirical contribution, and is addressed explicitly in the methodology.

The ADR Team

- **Researcher/Practitioner:** embedded member of DUR's innovation group, directs the design process, and builds the artifact.
- **DUR Management:** the director and team manager validate the capability map and evaluate the relevance and fit of the resulting artifact.
- **External Expert:** provides technical review of the prototype, assessing feasibility, maintainability, and scalability.

3.0 Findings

The findings follow the four ADR stages set out in the methodology. Together they move the research from a theory-based diagnosis of the problem, through the construction and evaluation of the prototype under real conditions, to the formalization of what the cycle produced.

Stage 1 draws on a structured literature review in two parts. The first examines Generative AI as a technology, establishing what it is, why it matters for a resource-constrained SME, and where its actual limits lie. The second uses established innovation theory to abstract Durable's situation into an innovation framework that defines the challenge and provides the basis for specifying the IT prototype.

Stage 2 covers the intervention. Sub-cycle A maps the firm's available resources with management and translates them into a resource map and a prototype specification. Sub-cycle B describes the GenAI-supported construction of the prototype and reports the main execution learnings produced during the build.

Stage 3 evaluates the prototype build through internal management feedback and an external technical review.

Stage 4 formalizes the findings into design principles, a gatekeeper profile, and boundary conditions for using GenAI in resource-constrained SMEs.

3.1 Introduction to Generative AI

This section introduces only the concepts about Generative AI that the thesis requires: a working definition of the technology, and then, in more depth, the specific tool class the fieldwork depends on. The innovation framework in Section 3.2 draws on effectuation and bricolage, theories developed before Generative AI existed. Applying them to a firm that has access to this technology requires knowing what the technology can and cannot do, because the proposition this thesis examines is that GenAI may change what counts as available means under resource constraint.

3.1.1 What Generative AI Is and Why It Matters for a Constrained Firm

Artificial Intelligence (AI) can broadly be described as the field of computational systems capable of performing tasks that typically require human intelligence, such as recognizing patterns, processing language, making decisions, or learning from experience (Banh and Strobel 2023). Within AI, machine learning and deep learning represent important developments because they allow systems to learn patterns from data instead of being explicitly programmed for every task (Janiesch et al. 2021).

Generative AI (GenAI) is a specific field within AI that focuses on producing new content rather than classifying, or predicting existing data. It refers to computational techniques capable of generating seemingly new outputs, including text, images, audio, video and code based on patterns learned from large training datasets (Feuerriegel et al. 2024; Banh and Strobel 2023). Unlike many previous AI systems, which primarily supported classification, prediction, or rule-based decision-making, GenAI can create new content in response to user input and can be applied across a wide range of general-purpose tasks.

Feuerriegel et al. (2024) also distinguish between the model level, the system level, and the application level. The model level refers to the underlying AI model, the system level to the infrastructure and interfaces that make the model usable, and the application level to the concrete use of GenAI for specific tasks. Applications such as ChatGPT or Claude are therefore more than the underlying model. They combine models with interfaces, infrastructure, data processing, and additional tools that allow non-expert users to interact with them. This thesis focuses exclusively on the application level.

For non-technology-oriented SMEs, these technologies represent a clear opportunity to generate information without needing to know how to produce it technically. Code generation, data structuring, scripting and document drafting were previously accessible only through technical specialists, whether hired or outsourced. GenAI applications offer these capabilities through a natural language interface at subscription prices. For a firm like Durable, with no technical staff and no software budget, this is the first technology that puts software production within reach of a non-expert employee. Whether this access can be considered a new resource that translates into usable capability is precisely the empirical question of this thesis.

1950's	Artificial Intelligence (AI) <i>All computational systems capable of performing tasks that typically require human intelligence</i>			
	1980's	Machine learning (ML) <i>Computer algorithms that learn from data to make predictions</i>		
		2010's	Deep learning (DL) <i>Machine learning models inspired by the human brain. eg. Neural Networks</i>	
			2020's	Generative AI (Gen AI) <i>Deep learning systems that generate new content</i>

Figure 02: Evolution from artificial intelligence to generative AI.

Output Modality (Selection)	Model level	Model level	Application level
	<i>Underlying AI model for different data modalities (e.g., image, text, code)</i>	<i>Embedding model functionality to provide interface for interaction</i>	<i>Solving dedicated business problems and stakeholder needs</i>
Text Generation	X-to-text models, e.g., GPT-4 and LLaMA 2	Conversational agents and search engines, e.g., ChatGPT and YouChat	- Content generation (e.g., SEO and customer service) - Translation and text summarization
Image/Video Generation	X-to-image models, e.g., Stable Diffusion and DALL-E 2	Image/video generation systems and bots, e.g., Runway and Midjourney	- Synthetic product and advertising visuals - Educational content
Speech/Music Generation	X-to-music/speech models, e.g., MusicLM and VALL-E	Speech generation systems, e.g., ElevenLabs	- AI music generation - Text-to-speech generation (e.g., news, product tutorials, etc.)
Code Generation	X-to-code models, e.g., Codex and AlphaCode	Programming code generation systems, e.g., GitHub Copilot	- Software development - Code synthesis, review, and documentation

Figure 03: Generative AI model, system, and application levels.

3.1.2 AI Coding Agents: The Tool Class of This Thesis

For the research questions of this thesis, the decisive development is not text generation but AI coding agents. LLM applications react to prompts by generating content. AI agents extend this by combining a foundation model with external tools, memory, and an execution environment. They are not new model architectures but system-level configurations that allow a model to use tools, retrieve information, execute code, and work through multi-step tasks with limited human intervention (Sapkota et al. 2026; IBM 2026).

Sapkota et al. (2026) distinguish AI agents from agentic AI by complexity and coordination logic. AI agents are usually modular systems designed to execute specific tasks, while agentic AI refers to coordinated systems of multiple specialized agents that communicate, allocate subtasks, and work toward broader objectives. Coding tools such as Claude Code and OpenAI Codex are AI agents under this definition.

Table 02: Comparison of LLM applications, AI agents, and agentic AI.

Feature	LLM Applications	AI Agents	Agentic AI
Primary Goal	Content creation	Specific task execution	Complex workflow automation
Initiation	User prompt	Goal-triggered with tool use	Orchestrated system-level goal
Structure	Single model	LLM + Tool integration	Multi-agent system (Meta-agents)
Memory	None/Short context	Optional task-specific	Persistent/Shared episodic memory
Planning	N/A	Single or simple sequential	Dynamic multi-step / Recursive
Interaction	User-driven	Tool-driven	Inter-agent collaboration

Based on Tables 1–5 in Sapkota et al. (2026).

What makes coding agents relevant to the research question is the working loop they enable. The user defines a goal in natural language. The agent then decomposes this goal into subtasks, writes code, executes it, reads errors, and revises the output until the result works or the user redirects the process. Coding agents can also read an existing file structure, explain what they find, propose next steps, and modify files directly.

This changes the role of the user compared with a simple chat interface. A chat-based application can generate code or scripts, but the user still has to copy the code, place it in the right location, install dependencies, understand errors, and run the script correctly. Coding agents reduce this barrier because they can work directly in the development environment. They can create files, edit existing code, install or suggest dependencies, execute commands, and iterate on errors. This allows more complex code to be developed without requiring the user to manually manage every technical step.

For a non-technical operator, the barrier therefore changes rather than disappears. The operator no longer needs to write all code manually, but still needs to define the goal, provide domain context, evaluate the output, and decide when the result is good enough. In the fieldwork, coding agents performed exactly this role. They built, tested, and debugged the data pipeline behind the prototype, while chat-based applications played only a supporting role in explanation, reflection, and documentation.

The practitioner defines goals, provides domain knowledge, evaluates outputs, and redirects the process when needed. The resulting process can be understood as a socio-technical interaction between human judgement and AI-supported execution. The next section examines the nature and limits of this interaction.

3.1.3 The Human-AI Sociotechnical System

The technical workings of Generative AI are not the concern of this research. What matters is the interaction between the operator and the tool, and what the two can build together. An AI system and the person directing it form a

sociotechnical system whose output depends on both (Feuerriegel et al. 2024). This framing is crucial for setting the boundary of what a resource-constrained firm without digital expertise can realistically achieve.

The system has one defining property that is also its principal limitation. A non-expert operator can usually tell whether an output runs, but not whether it is robust, secure, or scalable. The tool lowers the threshold to begin, but it does not supply the judgment required to evaluate what it produces. The socio-technical system still needs an external mechanism for technical validation.

The error risk is not limited to hallucination. Two distinct failures compound. First, the model can generate output that is plausible but wrong, and conversational fluency makes the error hard to detect (Feuerriegel et al. 2024). Second, and more important, the model can follow poor instructions efficiently. If the operator frames the task incorrectly, omits context, or accepts a weak result, the system produces code that functions superficially while remaining fragile. The constraint is therefore the operator's knowledge, not only the model's reliability. "Good Prompting" depends on understanding the task well enough to know what the model must be told (Lucchetti et al. 2024).

Dell'Acqua et al. (2023) demonstrate this pattern empirically. Inside the users's capability frontier, AI assistance raised output quality and speed. Outside it, on tasks requiring integrative judgment, assisted users were less likely to reach the correct answer, and reached the wrong one faster and with more confidence. The technology amplifies what the operator already knows and obscures what they do not.

This is why this account of GenAI is necessary before the framework. GenAI and agentic coding change the resource problem for small firms without eliminating it. They may allow a firm to begin experimenting before it holds full technical expertise, but the empirical evidence already indicates that GenAI is not a fully autonomous programmer available to any firm. Value depends on internal judgment, task framing, and validation capacity.

3.1.4 SMEs and Generative AI Adoption

If GenAI could simply replace missing internal capabilities, SMEs would already be using it widely in their core activities. The available evidence suggests otherwise. OECD (2025) reports that only 29% of SMEs using Generative AI apply it in core revenue-generating activities. Most adoption is concentrated in peripheral tasks such as marketing, administration, and basic IT support, where prior knowledge requirements are lower and mistakes have fewer consequences.

GenAI can help reduce skill gaps, but this effect is not automatic. Among SMEs facing labor shortages, 39% reported that GenAI helped compensate for those gaps. This increases to 46% among firms where GenAI had already improved employee performance (OECD 2025). This suggests that GenAI works better where some knowledge, resources, and organizational structure are already present. The limitation is also connected to the technology itself: 35% of SME non-adopters cite output quality as a barrier (OECD 2025). This is a reported perception, not a technical assessment, but it shows that uncertainty about reliability is already a practical concern for firms.

The pattern is therefore clear. GenAI adoption among SMEs is real, but limited, and concentrated where risks are low and technical knowledge requirements are small. This challenges the idea that GenAI can simply replace missing expertise. The data shows that adoption is only peripheral, but it does not explain why the limits appear. That explanation is the task of the theoretical review that follows, which applies it directly to Durable's situation.

3.1.5 Results: Limits of AI Coding Tools

Section 3.1 produces the first findings of this thesis. GenAI is a statistical system that generates outputs by predicting likely patterns from training data. It can produce convincing content, but it does not understand the task in a human sense, judge whether the goal is appropriate, or assess the wider context of its output. Coding agents extend these features by decomposing tasks, using tools, writing and executing code, reading errors, and revising their outputs. This makes them especially relevant for resource-constrained firms, because a non-technical operator can move from a natural-language instruction to a working prototype without having to write code.

However, this autonomy also creates a risk. When a coding agent produces a complete artifact, a non-expert user may only be able to judge whether it appears to work. They may not be able to assess code quality, reliability, scalability, security, or long-term maintainability. The tool therefore reduces the execution barrier, but it does not remove the need for human judgement and technical expertise.

Generative AI remains an early-stage technology and several sources point to instability, unreliability, and limitations, particularly in specialized domains. GenAI is fundamentally a probabilistic system rather than a deterministic one. It generates statistically likely outputs rather than guaranteed correct solutions. Research shows that once tasks move outside the user’s own knowledge domain, the probability of failure increases significantly.

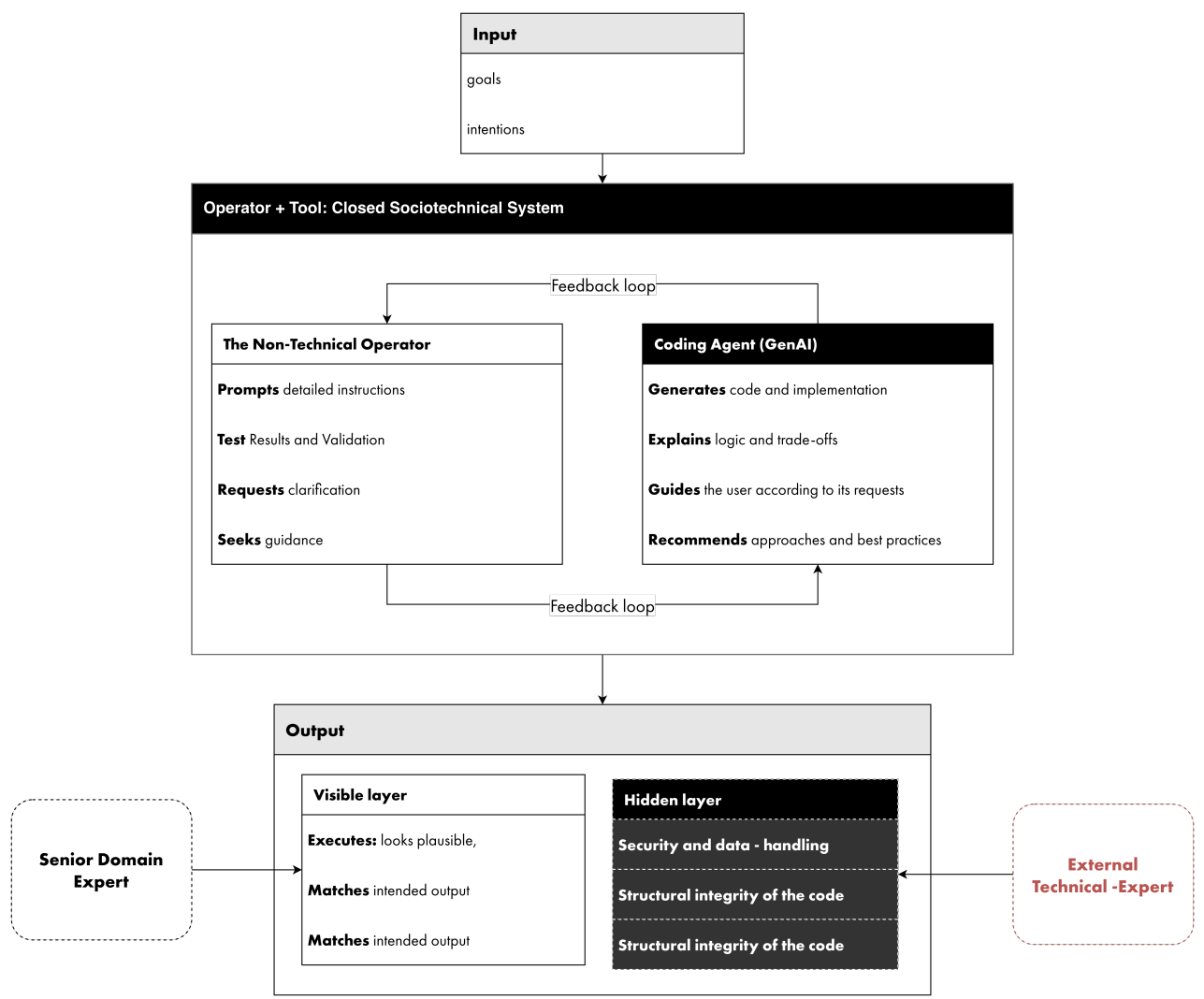


Figure 04: Human–AI sociotechnical system for non-expert prototyping.

Two findings follow from this. First, the value of GenAI depends on the operator’s ability to define the task, provide domain context, and evaluate the output. Second, prototypes produced in this way require external technical validation as well as senior domain expertise before they can be trusted, deployed, or scaled.

3.1.6 Stage 01: Design Principles Part I

This analysis resulted in the first design considerations for using coding agents under non-expert conditions. Task size and complexity determine the size of the output that later requires review, so larger tasks increase the opacity of agent-based work, particularly for non-expert users. Sequential development and repeated testing create opportunities to review outputs during the process rather than only at the end. Continuous documentation of prompts, decisions, errors, corrections, and tests keeps the process traceable for later review. Artifacts built with coding agents should be treated as prototypes, not as production-ready systems. External technical review is therefore needed before any prototype is trusted, deployed, or scaled. The following table summarizes these design principles.

Table 03: Stage 1A design principles for GenAI-supported prototyping.

ID	Design principle	Rationale
S1A.1	Keep outputs reviewable	The size and complexity of a task are directly related to the size of the output that later requires review. Larger tasks therefore increase the opacity of agent-based work, especially for non-expert users. Sequential development and repeated testing increase the possibility of reviewing outputs during the process, rather than only at the end.
S1A.2	Continuous documentation	Continuous documentation of prompts, decisions, errors, corrections, and tests is necessary to make the process traceable for later review.
S1A.3	Treat as prototypes, require external review	Prototypes developed with coding agents should be treated as prototypes, not as production-ready systems. External technical review is needed before prototypes are trusted, deployed, or scaled.

3.2 Stage 01: Problem Formulation

The second part of the literature review was conducted to better explain and understand the problem faced by Durable. It was not used as a general theory section, but as a way to generate insights for the problem formulation. Each theoretical perspective helped clarify one part of the problem and showed why a firm that is successful in its own market can still struggle to build digital innovation capability.

In ADR terms, the resulting framework functions as a theory-ingrained artifact: it translates the theoretical logic into a form that communicates the organizational problem to practitioners and provides a theory-based diagnosis of the process the firm is undergoing (Sein et al. 2011). It also forms the conceptual foundation for the resource map developed in the following sub-cycle.

3.2.1 Structure of the Literature Review

As argued in Section 1.2, more recent entrepreneurial approaches such as Lean Startup (Ries 2011) and scientific entrepreneurship (Camuffo et al. 2020) assume experimentation resources and affordable losses that the firms examined here do not have. Both Lean Startup and scientific entrepreneurship primarily address idea development and hypothesis validation once the necessary resources to act already exist. The problem exposed in this research is more fundamental: the conditions under which a non-technology-oriented small enterprise can access, recombine, or construct the minimum resources necessary to even begin digital innovation.

The review builds a sequential explanation of how resource-constrained, non-technology-oriented small enterprises encounter and respond to the resource gaps involved in initiating digital venturing. It begins with effectuation and bricolage as the primary lenses for innovation under uncertainty and resource scarcity (Sarasvathy 2001; Baker and Nelson 2005; Fisher 2012). Causation clarifies why a non-technology-oriented firm would deliberately enter digital markets despite lacking the necessary expertise, thereby creating a self-inflicted penurious environment. The review then examines open innovation as the established mechanism through which SMEs compensate for missing internal expertise (Chesbrough 2003; Brunswicker and Vanhaverbeke 2015), and absorptive capacity as the internal threshold that determines whether external knowledge can be used at all (Cohen and Levinthal 1990).

The review argues that these frameworks only partially resolve the problem faced by non-technology-oriented small enterprises pursuing digital venturing. Even after recombining internal resources and acquiring external support, an expertise gap remains. This unresolved gap is the entry point for Generative AI, whose conditions and limits were established in Section 3.1. The result is an integrated innovation framework that synthesizes causation, effectuation, bricolage, open innovation, absorptive capacity, and Generative AI, and that guides the development of the resource map and IT prototype.

Table 04: Theoretical concepts, roles, and contributions to the literature review.

Concept	Role in the Review	Main Contribution
Effectuation	Explains action under uncertainty	Shows how firms can begin with available means rather than predefined goals, but also where Durable's available means are insufficient.
Bricolage	Explains resource recombination	Shows how existing resources can be stretched and recombined, but also where this approach reaches its limits.
Causation	Explains entry into the problem	Shows why a non-technology-oriented firm may deliberately enter digital markets despite lacking the required expertise, creating a self-induced resource constraint.
Open innovation	Explains external resource access	Shows how SMEs try to compensate for missing internal expertise through external networks, partners, or knowledge sources.

Concept	Role in the Review	Main Contribution
Absorptive capacity	Explains limits of external knowledge	Shows that external knowledge cannot be used effectively unless the firm has enough prior internal knowledge to recognize, assimilate, and apply it.
Integrated innovation framework	Output of the review	Synthesizes the theories and GenAI into a framework that guides the resource map and the IT prototype.

3.2.2 Effectuation as a Lens for Innovation

There are two opposite approaches that frame how firms can deal with innovation under uncertainty: causation and effectuation (Sarasvathy 2001). Causal logic assumes that creating a business requires a clear definition of the product, business model, and desired outcome. The task is then to identify and execute the most efficient path toward that goal (Sarasvathy 2001). This logic assumes that outcomes can be specified in advance, that the required resources can be identified beforehand, and that uncertainty can be reduced to calculable risk. Such conditions are difficult to meet in contexts where markets are uncertain and resources are constrained. Causation is therefore a predictive and resource-intensive logic that depends on planning, market research, segmentation analysis, demand forecasting, and the organizational resources necessary to execute each stage (Sarasvathy 2001). For firms with limited budgets, no dedicated innovation teams, and undefined digital products, this process becomes slow and difficult to sustain.

Effectuation reverses this logic. Instead of defining an ideal outcome first and then assembling the resources required to achieve it, effectuation begins with the resources already available and asks what can be built from them (Sarasvathy 2001). Four principles define the approach:

1. **Affordable loss replaces expected return:** the central question becomes how much can be committed without unacceptable consequences.
2. **Strategic alliances replace competitive analysis:** uncertainty is reduced through stakeholder commitments and partnerships rather than extensive market prediction.
3. **Contingencies are treated as inputs:** unexpected developments are incorporated and exploited rather than avoided.
4. **Control replaces prediction:** the objective is not to predict the future but to shape and control what can realistically be influenced (Sarasvathy 2001).

These principles reflect the conditions faced by resource-constrained, non-technology-oriented SMEs. Such firms often lack a defined digital product, a predictable market, and the resources required to conduct extensive market validation before acting. Under these conditions, a causation-based approach becomes difficult to operationalize, while effectuation offers a more accessible starting point by relying on existing resources, affordable loss, and controllable means.

The relevance of effectuation for this research goes beyond its decision logic and rests primarily on the idea that innovators must begin from their current means. Sarasvathy (2001) defines three categories of means from which entrepreneurial action emerges: who you are, what you know, and whom you know. Rather than identifying all the resources required to achieve a predefined target, the entrepreneur asks what can be created from the resources already available. This expands the range of possible actions and outcomes because innovation begins not with a fixed goal, but with a given set of means from which multiple possible effects may emerge.

There is, however, a structural problem with the available means themselves. Sarasvathy's categories implicitly assume that some relevant resources already exist within the target domain. For non-technology-oriented SMEs, this assumption fails. "Who you are" does not include a digital identity, "what you know" does not include technical expertise, and "whom you know" reflects relationships tied to the firm's existing non-digital market. The digital resource base is largely absent.

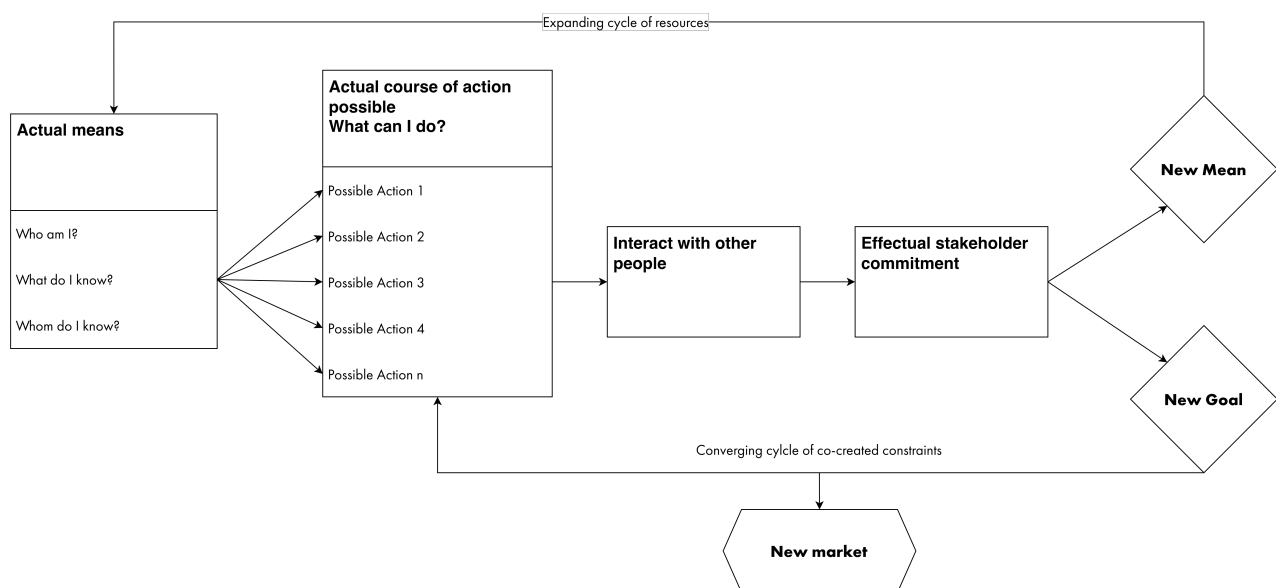
Applied to Durable, the diagnosis is concrete. Who the firm is: a sustainability consultancy, not a digital service provider. What it knows: building physics, life-cycle assessment, certification processes, not software development

or data engineering. Whom it knows: the Swiss AEC network of architects, developers, and planners, not technology partners or digital clients. None of Durable's effectual means reaches into the digital domain it has committed to enter.

Sarasvathy (2008) later extends effectuation through stakeholder commitments and partnerships, where each new stakeholder potentially introduces additional resources, relationships, and new possible goals. Technical expertise can therefore theoretically be accessed through alliances rather than developed internally. In practice, however, this often shifts rather than resolves the problem. Partnerships still require resources such as budget, negotiation capacity, governance alignment, and a sufficiently attractive value proposition to secure external collaborators. These are often precisely the resources constrained SMEs lack, increasing coordination and communication burdens rather than organizational agility.

Effectuation also suggests that when a path becomes excessively costly or difficult, firms should redirect their efforts toward goals that better fit their available means. For non-technology-oriented small enterprises, this creates a structural boundary around what can realistically be pursued. When technical expertise and viable partnerships are absent, the most rational effectual response may be to abandon digital venturing altogether. For Durable, a strictly effectual reading would therefore counsel retreat to its core consulting business. Why the firm cannot simply do this is explained in Section 3.2.4. Effectuation provides an appropriate starting logic for constrained firms, but it does not resolve the problem examined in this research.

This leads directly to bricolage. Once the available means define the limits of what is achievable, the question becomes whether those limits themselves can be expanded. Bricolage offers a broader interpretation of resources by focusing on recombination rather than acquisition (Baker and Nelson 2005). Instead of depending exclusively on external partners or entirely new investments, firms attempt to create new means from the resources already at hand.



1. **Making do:** a bias toward action and experimentation rather than waiting for ideal conditions or complete resource availability. This includes a refusal to fully accept existing constraints, standards, or institutional limitations.
2. **Combination of resources for new purposes:** the reuse and recombination of existing resources for applications different from those for which they were originally intended.
3. **Resources at hand:** the use of immediately available resources, including undervalued, discarded, inexpensive, amateur, or self-taught inputs that others may not consider strategically valuable.

Baker and Nelson (2005) identify five domains in which bricolage can operate: physical inputs, labor, skills, customers and markets, and institutional and regulatory environments. This thesis draws particularly on three aspects: the recombination of resources for new purposes, the use of resources at hand, and the productive use of amateur and self-taught skills. The five domains identified by Baker and Nelson (2005) were developed prior to the emergence of software and data as significant organizational resources for resource-constrained firms. Their “physical inputs” domain assumes tangible, discarded, or undervalued materials, a category too narrow for the resources relevant to this research. For this thesis, the domain is broadened and relabelled **Assets**, encompassing both physical and digital resources.

Through recombination and experimentation, firms may produce rough prototypes, automate internal processes, or explore digital opportunities that conventional resource logic would consider inaccessible. Importantly, recombination does not only create new products or services. It can also create new means: new internal resources that did not previously exist within the organization. This matters here because the objective is not initially the commercialization of a digital product, but the construction of internal resources that can gradually support a digital orientation.

Baker and Nelson (2005) further distinguish between **selective bricolage** and **parallel bricolage**. Selective bricolage occurs when firms apply bricolage only in specific domains while maintaining stable routines and standards elsewhere. Parallel bricolage occurs when firms rely on bricolage across nearly all organizational activities simultaneously. The distinction matters because selective bricolage is associated with greater growth potential, while parallel bricolage can trap firms in self-reinforcing cycles of low-quality resources and limited scalability. For this research, the implication is that bricolage should be a focused and limited response targeted at specific innovation problems, not a permanent organizational logic applied everywhere.

Applied to Durable, bricolage identifies what the firm has that conventional resource logic overlooks. Fifteen years of accumulated project data sit unstructured on a server: an undervalued physical input. One employee holds self-taught digital literacy in scripting and dashboarding: an amateur skill that formal capability assessments would not register. Existing software licenses, including Power BI through the holding, are resources at hand. Bricolage legitimizes treating these as innovation inputs and predicts that recombining them could produce a first internal artifact.

Bricolage also predicts where this ends. Even when firms successfully stretch and recombine existing resources, certain forms of expertise, such as software engineering, system architecture, or advanced digital product development, remain structurally difficult to reproduce through improvisation alone. Reallocating domain experts toward experimental digital work introduces uncertainty and organizational risk, and under regulated, resource-constrained conditions many firms are unwilling to sustain that level of experimentation. Bricolage is therefore not a substitute for missing technical expertise. It expands the range of possible actions and legitimizes self-taught skills, but the expertise gap does not disappear. Without sufficient expertise, organizational commitment, and resource allocation, the probability of successful digital venturing does not necessarily improve.

3.2.4 The Relationship Between Effectuation, Causation, and Bricolage

Until now, causation has been treated as unsuitable for resource-constrained, non-technology-oriented small enterprises. However, it cannot be completely ignored, because it explains how the venturing process starts. A causal decision explains why a non-technology-oriented firm would attempt digital venturing in the first place. Without such a decision, it would make little sense for a firm to voluntarily place itself in what Fisher (2012) describes as a penurious environment. Effectuation starts from existing means and a loose set of goals (Sarasvathy 2001). A firm with no digital expertise, no technical staff, and no experience in digital markets would only enter that space after recognizing and evaluating the opportunity. The decision to pursue digital venturing therefore requires an earlier

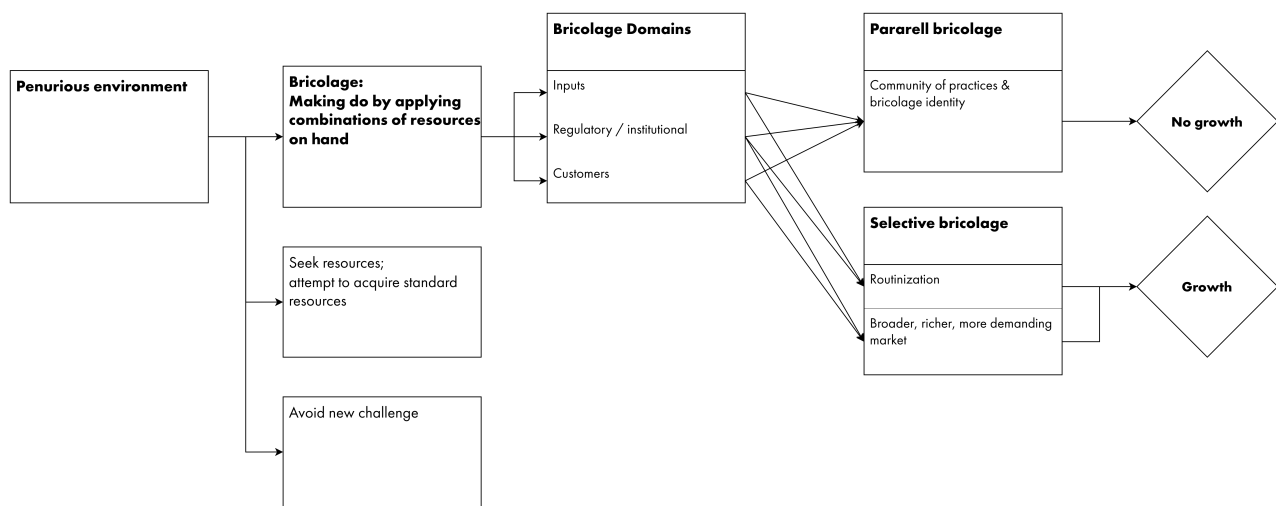


Figure 06: Bricolage domains and selective and parallel modes.

causal step: a strategic assessment based on market analysis, future goals, or the deliberate intention to pursue an opportunity outside the firm's current expertise.

Causation is not the logic used to implement the venture. It is the trigger that starts the process. This has a direct consequence for the effectual logic that follows. When management has formally recognized an opportunity and committed to a strategic goal, changing the effect becomes organizationally costly or can be seen as a failure, reducing the acceptable losses. The goal is no longer loose. It has been communicated and invested in. The effectual change of goals is presented by the framework as flexibility, but in practice it is experienced as failure.

This is exactly Durable's situation. The causal trigger came from outside the firm's own means: the Wüest Partner acquisition and the group's digitalization ambitions created strategic pressure, and management committed to exploring data-oriented services. Once that commitment was made and communicated within the holding structure, the effectual escape route of redirecting toward more familiar goals became costly. Durable did not begin as a resource-poor firm. It became resource-constrained the moment it committed to a digital direction its current resources cannot support.

Effectuation and bricolage alone are not enough to describe this type of situation. Both frameworks assume that the constrained condition already exists. Baker and Nelson (2005) describe firms operating in a penurious environment (Fisher 2012), while Sarasvathy (2001) describes effectuation as a response to uncertain goals and conditions. Neither fully explains firms that are successful in their existing markets but deliberately choose to enter a completely new domain where they possess no relevant expertise. In this case, the constrained condition is self-induced.

Causation therefore explains the origin of the problem addressed in this thesis, while effectuation and bricolage explain how firms respond after this constraint emerges. Fisher (2012) argues that these approaches are not contradictory. Causal, effectual, and bricolage behaviors can coexist, with the balance between them changing depending on the situation. When the strategic direction is fixed but the path forward remains uncertain, causal intent and effectual action can operate simultaneously, as shown in cases such as Tripadvisor and Flickr (Fisher 2012). This reflects the type of firm examined in this research. The interpretation adopted here is that causation, effectuation, and bricolage can be deliberately combined within a single analytical framework, with each explaining different phases and dimensions of the innovation process. Innovation in resource-constrained environments emerges not through a single logic, but through the interaction and shifting balance between causal intention, effectual action, and bricolage-based recombination.

3.2.5 First Conclusions and Boundary Conditions

The three innovation approaches can be combined into a conceptualization of the process a resource-constrained small enterprise undergoes when it ventures outside its core domain. This combined framework describes the penurious environment the firm enters and proposes an effectual approach with a bricolage-based complement to it.

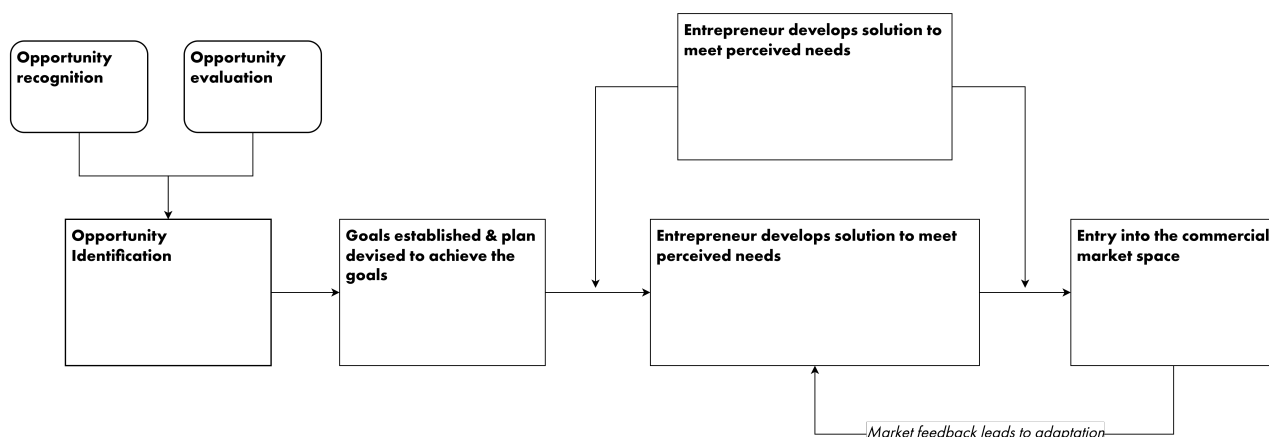


Figure 07: Market feedback and adaptation across innovation logics.

The combined framework does not solve the problem, but it makes the problem clearer and hints at the solution. By looking at causation, effectuation, and bricolage together, the framework shows where the internal approach reaches its limit. The firm can work with existing means, recombine available resources, and explore diverse courses of action. However, digital and technical expertise cannot simply be created from these existing means if it is not already available inside the firm.

The framework at this stage identifies where the capability gap must be compensated. If the knowledge is not available internally, it needs to come from external partners and collaborations. In effectuation terms, this relates to “whom I know”: external stakeholders, partners, or experts who can provide resources, knowledge, or technical capabilities that the firm lacks.

When resource-constrained SMEs innovate, they do so by relying on external networks, partnerships, and value chain relationships to compensate for missing internal capability (Brunswick and Vanhaverbeke 2015). De Massis et al. (2018) show that this external dependence is viable only under supporting conditions such as strong internal relationships, patient capital, and long-term institutional support. The framework points outward, and this is precisely where the open innovation literature begins. The next section therefore examines open innovation as the established mechanism through which constrained SMEs source external capability, and asks whether it resolves the boundary this framework has exposed.

3.2.6 Open Innovation

If the answer is not internal, the next logical place to look is outside the firm. Open innovation, as developed by Chesbrough (2003), proposes exactly this. Firms can compensate for missing internal resources by accessing knowledge, technology, and expertise from external actors such as partners, users, suppliers, universities, or networks (Chesbrough et al. 2006). Firms do not need to own all resources internally to benefit from them. For SMEs, this often takes the form of outsourcing, hiring consultants, creating partnerships, or external knowledge sourcing. However, for resource-constrained small enterprises, this only shifts the problem: external knowledge may be available, but accessing it still requires time, administrative capacity, coordination effort, or money.

The firm must identify what it lacks, locate external expertise, negotiate collaboration, and manage integration. The firm also becomes dependent on “whom they know” and on external actors doing much of the heavy lifting. These relationships take time, require credibility, and transfer part of the control over the future to external actors. Externalizing the innovation process is therefore not a shortcut around internal capability building. It requires a minimum level of internal capacity to understand what is missing, evaluate what external actors can offer, and coordinate the collaboration effectively.

Both Chesbrough (2006) and Brunswick and Vanhaverbeke (2015) show that successful external knowledge sourcing depends on underlying internal resources. Firms still require absorptive capacity to recognize, evaluate, integrate, and manage external knowledge effectively. De Massis et al. (2018) show that constrained innovation depends on supporting organizational conditions such as patient capital, strong internal relationships, and long-term institutional

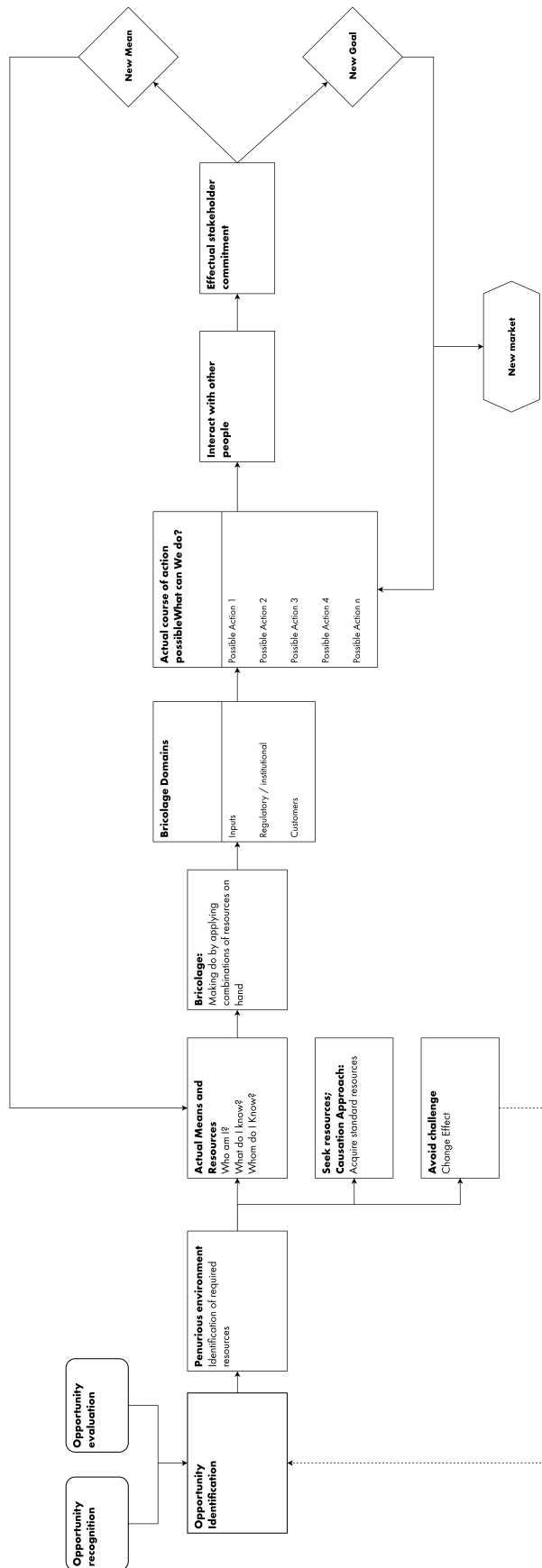


Figure 08: Bricolage action creates a new market.

support structures. This makes the internal resource problem visible again. Externalizing resources does not remove the investment requirement, and for firms with little room for failure, this can become a significant strategic risk.

For Durable, the diagnosis is again specific. The firm's established network sits in the Swiss AEC sector: architects, developers, planners. It contains domain credibility but no digital capability. The one external channel that does contain digital expertise, the Wüest Partner ecosystem with its data and software firms, is only indirectly accessible: strategic initiatives require holding-level approval, which reduces the agility that open innovation partnerships presuppose. Durable therefore illustrates the limit case: external knowledge exists in principle, but the coordination capacity, autonomy, and internal evaluation ability needed to use it are themselves constrained.

Open innovation literature does not offer a direct answer to the problem. What it shows is that some SMEs innovate by opening their boundaries, but only at the cost of significant coordination effort and managerial investment (Brunswick and Vanhaverbeke 2015). The useful insight is not that coordination substitutes for technical expertise. It is that managerial capability is required on top of it. Even if Generative AI could supply the full package of technical expertise, management would still need to recognize its value, coordinate its use, and commit the organizational effort to integrate it. The technical gap may become navigable through coordination rather than internalization, but the managerial and absorptive requirement does not disappear. It becomes the binding condition. Even when the entire body of external knowledge is dropped from the sky, the firm must still possess the capacity to absorb it (Cohen and Levinthal 1990).

3.2.7 Absorptive Capacity: Preparing for Generative AI

Two conclusions follow from the review so far. First, the established frameworks of constrained innovation, effectuation, and bricolage do not resolve the problem faced by non-technology-oriented small enterprises pursuing digital venturing. Each either assumes a minimum resource base these firms do not possess or requires coordination, expertise, and organizational resources that are themselves difficult to develop under constrained conditions. They do, however, frame the problem more precisely. Second, research on SME open innovation shows that small enterprises can innovate by relying on external networks and partnerships, but a minimum threshold of internal expertise remains necessary. Firms must still recognize what they need, evaluate what they receive, and integrate external inputs into their operations. Below this threshold, neither partnerships nor external networks produce usable outcomes.

This threshold is absorptive capacity. Cohen and Levinthal (1990) establish that the capacity to recognize, assimilate, and exploit external knowledge depends on prior related knowledge already held by the firm, and that this capacity accumulates incrementally over time. For non-technology-oriented small enterprises, the boundary condition is therefore not static. It can worsen with delay, but it can also be built over time. Because learning builds on prior knowledge, the absence of early investment makes later capacity more costly to develop. Cohen and Levinthal (1990) describe the extreme case as lockout: a firm can fall so far behind in a fast-moving field that it can no longer assimilate or exploit new knowledge, regardless of its value. The widening gap of firms that never begin is a form of technological debt and must be considered.

This argument supports the early implementation of GenAI. Delaying adoption risks leaving the firm without the knowledge and capabilities required to engage with the technology later. Early experimentation, even at low cost and limited scale, allows the firm to accumulate experience and build the internal capacity needed for future use. Bricolage suggests that self-taught and undervalued resources can become productive inputs (Baker and Nelson 2005), while Cohen and Levinthal (1990) show that even small early investments in knowledge improve future learning. These arguments suggest that constrained small enterprises could begin accumulating digital resources by implementing Generative AI, combining it with existing amateur skills, and building incrementally until digital venturing becomes feasible.

However, Generative AI cannot be understood as a magic solution for innovation, even if the technology becomes as powerful as promised. In any scenario, management still needs to understand what it can do, how it should be implemented, and how it connects to the firm's strategic goals. Theory therefore excludes Generative AI alone as an immediate source of innovation. It suggests instead that non-technology-oriented small enterprises must start early and build digital competencies over time. If a firm is too far behind to understand what the technology can do, how it should be implemented, or how its outputs should be evaluated, then the capability gap remains too large. In that case, GenAI cannot function as an innovation source, because the firm cannot absorb or apply what the technology

produces. Firms that delay experimentation may not only miss immediate opportunities, but also fail to accumulate the knowledge required to use GenAI later.

This firm-level problem leads to the more local question of who inside the organization can start this learning process. Cohen and Levinthal (1990) address this through the concept of the gatekeeper: an individual who holds enough background knowledge to monitor the external environment, translate incoming information, and make it intelligible to the rest of the firm. In the context of GenAI, such a person would need enough prior knowledge to evaluate outputs, guide experimentation, and connect technical possibilities to organizational problems and field expertise.

However, the gatekeeper is necessary but not sufficient. Cohen and Levinthal (1990) argue that an individual's absorptive capacity does not constitute the absorptive capacity of the whole unit. The wider organization must also share a minimum baseline of background knowledge, including common language, concepts, and symbols. Without this shared ground, the gatekeeper has no effective way to communicate what GenAI produces, and the organization cannot evaluate or integrate the results.

At Durable, no such gatekeeper existed before this research. The firm had no employee whose role included monitoring digital technology, and no shared internal vocabulary for data or software work. The researcher, a domain professional with self-taught digital literacy, was the closest available candidate. The empirical part of this research therefore asks how much prior knowledge the gatekeeper must hold, and how much shared understanding the organization needs, before GenAI can become a productive input for digital venturing.

3.2.8 Results: Innovation Framework

The final product of this literature review is a theory-based diagnosis of the conditions under which Durable, a resource-constrained, non-technology-oriented SME attempts digital venturing. The following table compresses the diagnosis: what each theory explains about Durable's situation, and what follows from it for the design.

Table 05: Theory-based diagnosis and design consequences for Durable.

Theory	What it explains in the diagnosis	Finding applied to Durable	Consequence for the design
Effectuation	How a constrained actor decides under uncertainty	None of the firm's means (who it is, what it knows, whom it knows) reaches into the digital domain	Begin from existing means, not from a target outcome
Bricolage	How the range of existing means can be expanded	Undervalued resources exist: fifteen years of project data, one self-taught employee, existing software licenses, access to coding tools.	Recombine selectively for one defined first step
Causation	Why the firm entered the constrained condition	The constraint is self-induced: a communicated strategic commitment was given. Redirecting or changing the goal is read as failure	Scope commitments within affordable loss
Open innovation	Why external networks do not simply close the gap	The relevant digital network is absent; the holding channel requires approval and coordination capacity the firm lacks	Do not treat external sourcing as a shortcut around internal capability
Absorptive capacity	The internal threshold for using any external knowledge	No gatekeeper existed; prior knowledge is minimal, and delay compounds the gap toward lockout (Cohen and Levinthal 1990)	Start early with small experiments and assign a gatekeeper

Theory	What it explains in the diagnosis	Finding applied to Durable	Consequence for the design
Generative AI (3.1.5)	The possible compensator, and its limits	Value depends on operator knowledge and task framing; outputs cannot be validated internally	Treat outputs as working models requiring external technical validation

Generative AI sits outside the firm's existing means: it is not internalized as a skill, and it is supplied by external commercial providers the firm does not control, which makes over-reliance a strategic risk. And the framework is deliberately narrow: dimensions such as corporate identity, organizational agility, innovation culture, or prior innovation experience remain outside its scope, in order to isolate the specific role of GenAI in resource-constrained digital venturing.

The synthesis of these elements is the innovation framework shown in the following figure. It combines several approaches to innovation. Causal intention explains why the venture is initiated. Effectuation defines what can be attempted from the means already available. Bricolage explains how undervalued or underused resources can be recombined into new resources. GenAI can expand this range further, but only under specific knowledge conditions. Open innovation and absorptive capacity show where internal action must be supplemented, interpreted, and validated through external knowledge.

This framework forms the basis for the resource-mapping framework developed in Sub-cycle A (Section 3.3.1). The innovation framework diagnoses the condition in which a firm like Durable finds itself when it attempts to explore digital markets without sufficient resources or technical expertise. The resource map then operationalizes this diagnosis by identifying which resources already exist, which resources are missing, and which resources can be recombined into a feasible first step.

3.2.9 Stage 01: Design Principles Part II

A second set of design considerations follows from this diagnosis. The firm's existing means should be mapped before defining what to build: what it is, what it knows, which resources it holds, and whom it can access. The resource constraint should be understood as self-induced, because communicated goals and prior commitments make redirection organizationally costly. The first step should stay within affordable loss, and it should recombine underused or undervalued resources, such as existing data, tools, and self-taught skills, into one clearly defined artifact. GenAI can extend existing expertise, but artifacts should remain within the operator's knowledge range and be treated as working prototypes. A gatekeeper is needed to initiate, coordinate, document, and communicate the process, while internal stakeholders validate relevance and external experts validate technical quality before anything is trusted, deployed, or scaled. External networks and GenAI are inputs that require internal coordination, judgement, and absorptive capacity, not substitutes for building expertise. Experimentation should therefore be treated as an investment in future capability rather than a demand for immediate returns, since the knowledge it generates can itself become a new resource. The following table summarizes these design principles.

ID	Design principle	Rationale
S1B.1	Map existing means first	The firm's existing means need to be mapped before defining what to build. This includes what the firm is, what it knows, which resources it has, and whom it can access.
S1B.2	Recognize constraints as self-induced	The firm's resource constraint needs to be understood as self-induced, because communicated goals and prior commitments can make redirection organizationally costly.

ID	Design principle	Rationale
S1B.3	Stay within affordable loss	The first step needs to remain within affordable loss, so that experimentation is possible despite limited resources and uncertain outcomes.
S1B.4	Recombine underused resources	Underused or undervalued resources, such as existing data, tools, and self-taught skills, can be recombined selectively for one clearly defined first step.
S1B.5	Existing expertise can be extended through GenAI	Existing expertise can be extended through GenAI, but artifacts should remain within the operator's knowledge range and be treated as working prototypes rather than finished systems.
S1B.6	Establish a gatekeeper role	A gatekeeper role is needed to initiate, coordinate, document, and communicate the process across the organization.
S1B.7	Validate internally and externally	Internal stakeholders are needed to validate relevance and usefulness, while external technical experts are needed to validate technical quality before prototypes are trusted, deployed, or scaled. External networks and GenAI should be treated as inputs that require internal coordination, judgement, and absorptive capacity, not as substitutes for expertise building.
S1B.8	Frame experimentation as investment in future resources	Experimentation and exploration should be understood primarily as investments in future resources and capabilities, not as activities that must immediately produce monetizable results or large efficiency gains. Through testing and exchange with others, the firm generates knowledge that can become a new resource or open new opportunities and markets.

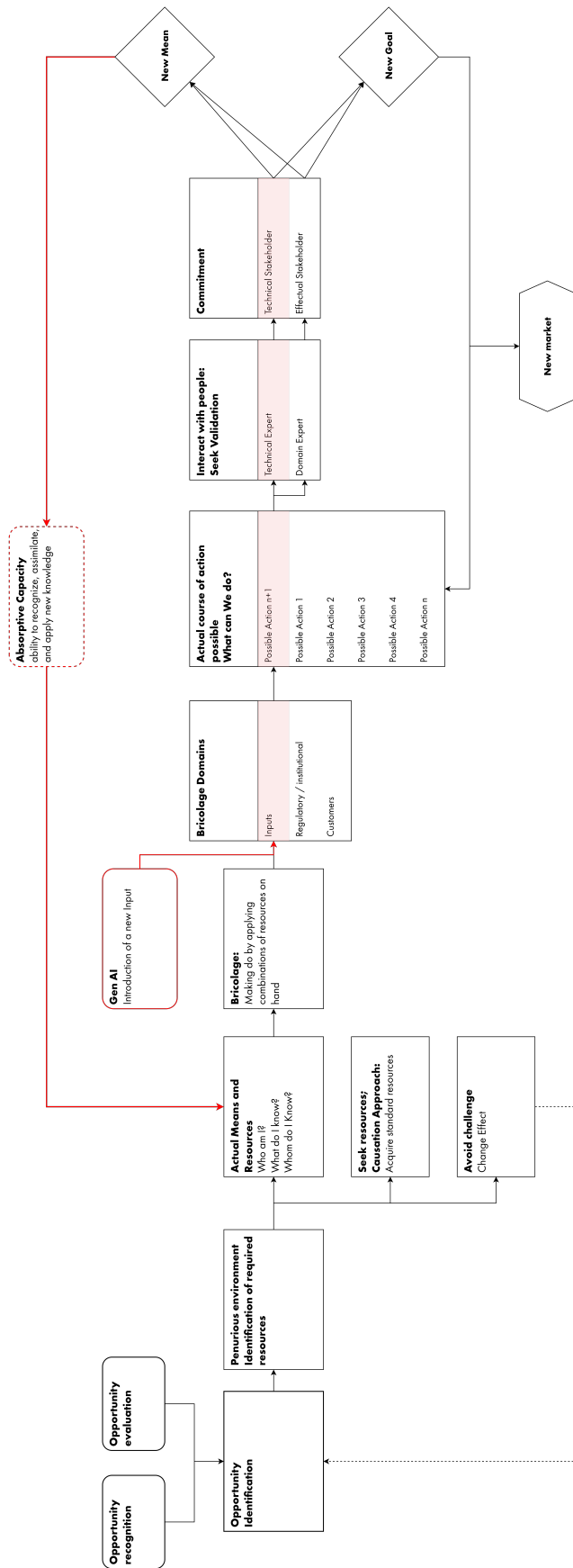


Figure 09: Integrated innovation path for resource-constrained digital venturing.

3.3 Stage 02: BIE, Sub-cycle A: Resource Map and Prototype Specification

Stage 2 moves the theoretical framing into the practical intervention. In Action Design Research, this stage combines building the artifact, intervening in the organization, and evaluating the result. The artifact is built, tested, and discussed inside the organization.

In this research, the alpha version was not a technical artifact but a resource-mapping exercise to identify needs that DUR had and that could be addressed using GenAI. Before building anything, the framework indicated the necessity of identifying first what the firm already had, what was missing, and where GenAI could realistically support the firm. This first sub-cycle therefore produced the resource map and the initial prototype specification.

3.3.1 Analytical Framework for Resource Mapping

The resource-mapping exercise translated the innovation framework into a practical artifact that could be used in discussion with management. Its purpose was to make the framework operational: instead of discussing resources, capabilities, and constraints only at a theoretical level, the workshop required a structure that could help identify what Durable actually had, what was missing, and what could realistically be recombined.

Barney's (1991) resource-based view was used to ground this discussion. As a seminal contribution in strategic management, it provides a clear way to identify and classify firm resources. Barney defines firm resources as the assets, knowledge, relationships, and organizational structures controlled by a firm. He groups these resources into three categories:

1. Physical capital: technology, equipment, location, data, and material assets.
2. Human capital: experience, judgment, knowledge, relationships, and skills.
3. Organizational capital: planning systems, coordination structures, routines, and internal or external relationships.

Just as with the case of Baker and Nelson (2005), Barney's (1991) paper on the resource-based view does not explicitly mention data as we know it today to be a valuable resource. The interpretation in this thesis is that data is best understood as an intangible strategic resource embedded in digital infrastructure, human expertise, and organizational routines, rather than as a standalone physical asset. Nevertheless, for simplification, it will be considered part of physical capital, as being part of the firm's data infrastructure. All other intangibles, such as the know-how to interpret, curate, analyze, or exploit data, or routines, are part of either the human capital expertise or the organizational structures and reporting systems.

The mapping tool is organized into five parts:

The first part applies Barney's three categories under a causal lens: what resources would a fully resourced digital venture require, regardless of what Durable currently has. This establishes a reference point, the ideal scenario in which the firm possesses every asset a digital venture would demand. It is not meant as an achievable target, but as the benchmark against which the actual gap, and GenAI's potential to bridge it, can be assessed.

The second part narrows the same RBV categories to what Durable actually possesses, structured through Sarasvathy's (2001) effectual questions: who we are, what we know, and whom we know. Each RBV category can sit under either of the first two questions. Software licenses, for instance, are not who Durable is, but what Durable knows how to use. Data, conversely, is something Durable is, part of its accumulated identity, but not necessarily something Durable knows how to exploit. This cross-tabulation makes visible not just what resources exist, but whether the firm recognizes and can act on them.

Whom we know is treated separately, divided per Sarasvathy (2001) into three categories: suppliers, clients, and partners. Suppliers are less relevant for a firm like Durable, but were kept for the sake of generalizing the framework.

The third part applies Baker and Nelson's (2005) bricolage domains directly, without mapping them onto RBV categories. This separation is deliberate. RBV is less helpful for classifying overlooked resources: an organizational routine that is simultaneously a recognized structure and an ignored asset is contradictory under Barney's logic. Bricolage's domains handle this condition directly, and were kept as originally defined for that reason. The only change made was to reframe physical inputs as Assets (physical and digital), to include data, software, and other digital assets that go beyond physical infrastructure.

GenAI is kept as a separate field, used as a container for brainstorming how it could serve as a compensatory mechanism for the gaps identified. This includes all possible applications of GenAI that could be recombined with the firm's overlooked or actual means.

The fifth part records the actual course of action: concrete project options, each treated as a distinct possible next step. The selected option was build as the IT Prototype.

The framework used can be seen in the following figure. This framework was implemented in Miro, a digital white-board tool, for the workshop session, with each field populated using digital sticky notes.

3.3.2 Resource Mapping Workshop

A resource-mapping workshop was held on 14 May 2026 with Durable's management. The participants were Adrian Eitle, Managing Director, and Nicolo Guariento, Team Manager of the Digitalization Team. The researcher prepared the session as a one-hour workshop.

The session was a 1 hour workshop divided into two parts. During the first part the researcher, presented the findings from the literature review in form of a presentation. In particular, the researcher explained the framework and gave a quick introduction to GenAI and its limits. During the second part, the workshop used the analytical framework to identify Durable's existing means, missing capabilities, and possible GenAI-supported actions.

The following figure documents the completed resource map produced in the workshop: Durable's physical, human, and organizational capital mapped against who we are, what we know, and whom we know, with the identified gaps marked. This is the visual anchor for the diagnosis in Section 3.2 and the specification in Section 3.3.3.

The workshop produced four findings. First, Durable's resources were identified as a clear domain expertise, fifteen years of project data, professional reputation, and access to the Wüest Partner ecosystem. As expected the capabilities required to convert these inputs into a digital service were missing.

Second, the workshop positioned GenAI as an accelerator rather than a substitute. Participants identified concrete tasks it could support: extracting structure from unstructured files, generating scripts, automating simple processes, and lowering the threshold for prototyping. None of these resolve the four constraints that block venturing, namely data governance, data quality, technical architecture, and product ownership. GenAI lowers the cost of entering the learning curve without supplying the organizational capabilities the venture ultimately requires.

Third, the action path (n+1 to n+3) functions as a capability-discovery mechanism rather than a product roadmap. Its stated logic is that several missing capabilities become visible only after a small artifact exists. The value of the first prototype is therefore diagnostic. It tests whether the project data can answer useful questions before further resources are committed.

Fourth, data quality is a big unknown. The uncertainty recurs at every layer of the board: under what the firm knows, under bricolage inputs, and in the questions guiding the prototype stage. After fifteen years of accumulated project files, the firm has no structured data that is consistent, or usable.

Table 07: Actual resources versus required capabilities.

Layer	Existing	Missing
Physical	Existing hardware, sustainability software licenses	Structured data, Cloud and data architecture infrastructure
Human	Domain experts, self-learned digital skills, Power BI users	Data engineers, developers, product owners
Organizational	Reputation, process knowledge, WP ecosystem access	Digital identity, AI governance, digital credibility

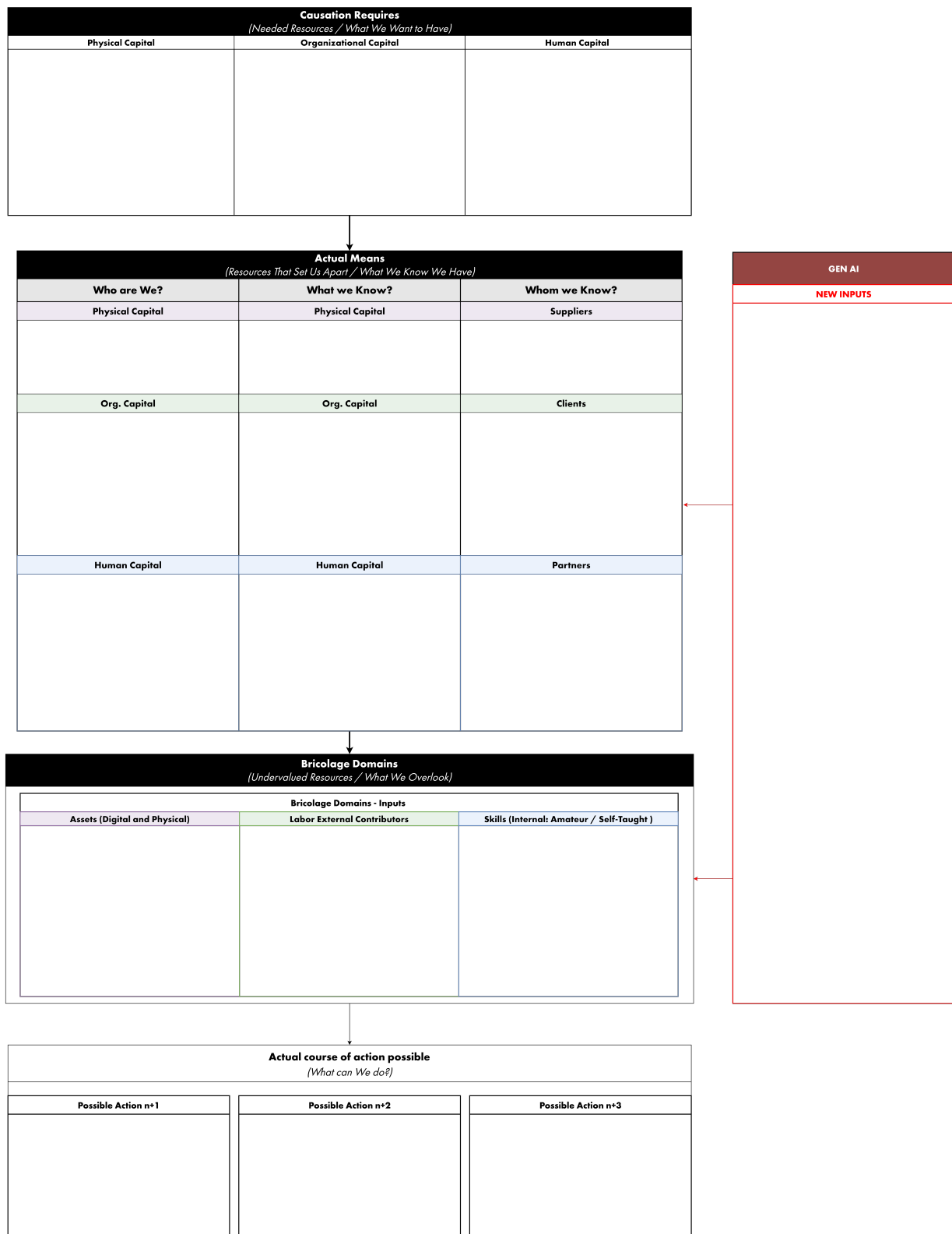


Figure 10: Analytical framework for resource mapping.



Figure 11: Completed workshop resource map.

Table 08: The role of GenAI as an accelerator rather than a substitute.

GenAI can do	GenAI cannot do
Structure unstructured files	Resolve data governance
Generate scripts	Verify data quality
Automate simple processes	Build technical architecture
Lower prototyping threshold	Supply product ownership

The Three Candidate Actions identified where:

n+1: internal exploration and alignment. Rejected as a standalone endpoint. Durable was not ready to build a client-oriented solution, no resources were committed, and the experiment window was one month. Any first action had to be internal, small, and simple enough to fit that constraint.

n+2: a low-tech internal MVP. Selected as the specification. Its purpose was to show the executive board what was possible from existing internal data, and why a structured database mattered. The MVP tested a prior question: whether Durable's project data held any value once organized. Only a positive answer would justify further investment.

n+3: a complete internal project database. Inconclusive. It raised open questions rather than settling a design, functioning as future ambition rather than specification. One concrete use case emerged: collecting past competition results to quantify embodied and operational CO2 across competing bids, once a data foundation existed.

Table 09: Action path for capability discovery rather than a product roadmap.

Action	Status	Rationale	Output
n+1: external exploration	Rejected as standalone endpoint	Durable not ready for a client-oriented solution; no resources committed; one-month experiment window. First action had to be internal, small, and simple	Constraint set for the specification
n+2: low-tech internal MVP	Selected	Show the executive board what existing internal data makes possible, and why a structured database matters. Tests whether project data holds value once organized; only a positive answer justifies further investment	MVP specification
n+3: complete internal project database	Inconclusive	Raised open questions rather than settling a design; functioned as future ambition. One concrete use case emerged: quantify embodied and operational CO2 across all projects, once a data foundation existed	Future direction, not a specification

3.3.3 Main findings from the Workshop:

The workshop's central finding is that Durable lacks digital capability almost entirely. No data engineering, no software development, no product ownership. The one internal asset is the researcher's self-taught scripting and working knowledge of Power BI, and Power BI already sits under available software. Everything else is missing.

This defines a narrow path. The researcher can steer GenAI to support the scripting and scraping of the server to test whether Durable's project data justifies investment in a proper database. The Wüest Partner ecosystem is treated

as external technical control, not as a building partner: internal budget constraints make acquiring those services unfeasible until a clear use case exists. Data quality remains the binding unknown. After fifteen years of project files, usability is unverified, and the full path is gated on this condition.

The prototype therefore serves as a diagnostic instrument and proof of concept, not a product. Its purpose is to test whether anything useful can be recovered from the server before further capability or budget is committed. Missing capabilities surface only after this first artifact exists.

Table 10: Resource-mapping findings by dimension.

Dimension	Finding
Digital capability	Almost entirely absent. No data engineering, no software development, no product ownership
Sole internal asset	Researcher's self-taught scripting and working knowledge of Power BI (already listed under available software)
GenAI role	Supports scripting and scrapes the server to test whether project data justifies a proper database
Wüest Partner ecosystem	External technical control, not a building partner. Internal budget rules out paid services until a clear use case exists
Binding unknown	Data quality. After fifteen years of project files, usability is unverified. The full path is gated on this condition
Prototype status	Diagnostic instrument and proof of concept, not a product. Tests whether anything useful can be recovered from the server before further capability or budget is committed

The workshop selected n+2 as the action carried forward into Sub-cycle B: the construction of a simple internal database prototype based on existing Durable project data. At this point, the prototype did not yet have a defined name, feature list, or technical architecture. The decision was better understood as permission to explore than as a finished product brief.

3.3.4 Critical Reflection

The specification produced by the workshop was not a bounded requirement. It amounted to a shared desire to investigate what could be made visible from existing project data. The purpose was not to build a finished tool, but to test whether Durable's internal data could become useful once extracted, cleaned, and structured.

This outcome should still be read critically. Of the one hour allocated, a substantial part was used to establish shared context: introducing the thesis framing, the literature review, Action Design Research, effectuation, bricolage, and the basic concepts of Generative AI. This left limited time for the actual resource-mapping exercise. As a result, the workshop could define a direction, but not a precise scope, feature list, or success criterion.

Securing the resources was itself difficult. Organizing a one-hour workshop proved hard to arrange. The researcher provided the time as part of the thesis, and committed to building the prototype on his own time. That offer was what made management's commitment possible. As a result, the workshop could define a direction, but not a precise scope, feature list, or success criterion.

Producing a well-specified prototype requirement through this kind of participatory workshop requires more time than was available. Participants unfamiliar with the theoretical framing need a separate context-setting session before the actual resource-mapping work begins. Durable's resource constraints did not only limit technical capability. They also limited the quality of the specification process used to define what should be built.

3.3.5 Sub-cycle A: Design Principles

These design principles follow from the workshop. They target the workshop itself as part of the artifact, informing how the resource-mapping process should be improved in future iterations:

Table 11: Stage 2A design principles for prototype specification.

ID	Design principle	Rationale
S2A.1	Diagnose before building	Test whether existing data holds value before committing to a tool or product
	Direction over specification	Under time constraint, the workshop can set direction but not fix scope, features, or success criteria
S2A.2	Separate context from mapping	Participants unfamiliar with the framing need a prior context-setting session before resource mapping
S2A.3	Constraint shapes specification	Resource limits degrade not only technical capability but the quality of the specification process itself

3.4 Stage 02: BIE, Sub-cycle B: IT Prototype

Sub-cycle B is the core empirical stage of the research. It moves from specification to construction: the n+2 decision reached in Sub-cycle A had to be turned into a working artifact, inside a one-month window, by a single researcher-practitioner supported by GenAI. The process was documented twice. The first log was created after each interaction with the AI coding agent: a template was generated, and after each coding session the GenAI system summarized all user prompts. A second log was kept through the researcher's own voice-recorded reflections. Read together, these two logs give both a record of what was built and an internal monolog of what the researcher understood, misunderstood, and learned while building it.

The IT prototype was built primarily with Claude Code and OpenAI Codex through VS Code, which served as the code editor and interaction environment.

3.4.1 Working definition of a Database

A working definition of a database was established to clarify the prototype's minimum requirements and to set the boundaries for the one-month development period. A database is a digital repository for storing, managing, and securing organized collections of data (IBM 2024). In this context it is understood as a structured storage location where an organization stores selected information in an ordered form, so that it can be found, updated, and combined quickly. It holds only specifically chosen values, not all available information. A software layer stores, reads, updates, and deletes that data, and multiple users and programs can access it simultaneously. Every database serves a defined purpose: CRM for customer relationships, project tools for tasks and scheduling, ERP for finance and accounting.

Four guiding principles shaped the construction of the database:

1. Selected data is stored in a clear structure
2. Only specifically chosen values are held, not all available information
3. A software layer stores, reads, updates, and deletes that data
4. Multiple users and programs can access it simultaneously.

Fifteen years of project files exist as unstructured documents, not as selected values held in a defined structure and served through a software layer. The prototype described in this section is the first attempt to test whether that raw material could be organized toward such a structure. This definition sets the reference point for what follows.

3.4.2 DUR Data Landscape

Durable already operates several separate databases. They serve different purposes, but are only partly integrated, or not integrated at all, and do not form a central data foundation. Where connections between systems do exist, they were built for the reporting needs of the Wüest Partner group, not for Durable's own internal use. All data generated by Durable resides on the Durable server, yet the server is not a structured database of internally collected insights; it works only as a storage for working documents and information.

This gap is fundamental given Durable's core business. The firm certifies buildings and calculates the embodied carbon and energy efficiency of planned projects. It produces these numbers for clients one project at a time, but holds no consolidated view of its own work. Projects rely on a mix of proprietary software, Excel tools, and third-party tools, with no central database: each is a set of separate closed files that must be opened individually to identify key values. The firm cannot state how many projects it has completed, for which clients, or of what type, without opening each file by hand or drawing on the implicit knowledge of senior staff.

The immediate goal was therefore not an emissions analysis of the full project portfolio. That remains a long-term ambition. The prototype was the first step: build from the ground up by organizing every project, assigning each a unique identifier, and recording basic attributes such as client, project type, and count. Only once that foundation exists can project-level indicators be aggregated for wider analysis.

The prototype therefore served two purposes: to establish a first structured mapping of Durable's projects, and to test methods for extracting values locked inside closed proprietary files. This condition, established before the workshop and confirmed during it, justified the intervention. Durable first needed to show that a structured internal foundation

was worth building at all. The prototype should therefore build a basic database in its minimal expression: a working tool, and a means of communication that makes the abstract idea of a database tangible.

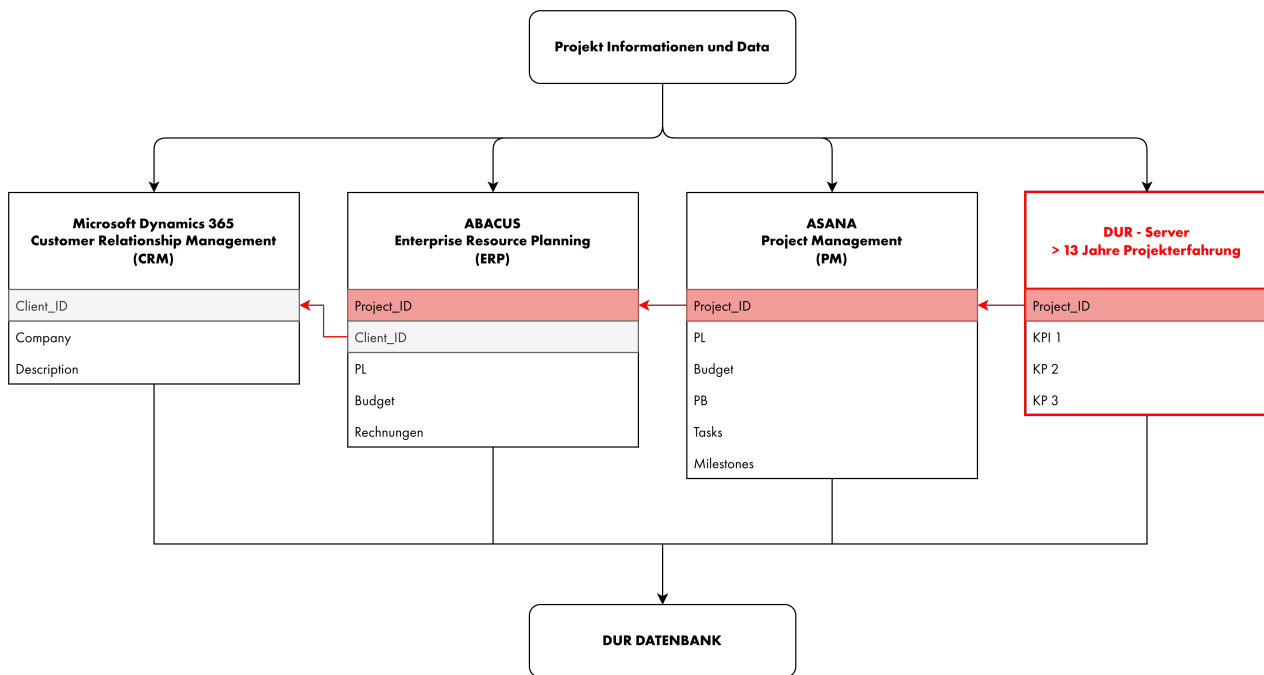


Figure 12: Prototype data-integration architecture.

3.4.3 Data Basis: Sources, Selection, and Cleaning

The pipeline was organized around a single stable identifier. Every project carries a Project ID issued by the firm's ERP system, and the server's folder structure is built on that same ID. This gave the intervention one reliable anchor in an otherwise inconsistent server, and made it possible to structure the extracted data as relational tables keyed on the Project ID rather than as isolated files.

The data was held in CSV throughout. This was a deliberate choice for a self-taught operator, not a technical default. A CSV is a direct visual representation of a table: if the columns come out as expected, the script did its job. For an operator who could not read the code with an expert's eye, the output format itself became the verification mechanism. CSV is also the fastest path into Power BI, which reads the final tables directly.

The planned pipeline ran in six steps:

1. Scan the project server to build a map of which projects exist and where each file sits, output as relational CSV tables keyed on the Project ID.
2. Copy every file matching defined criteria (format, date, type, active status) into the working repository, leaving originals untouched.
3. Parse the copied files: OCR for procurement PDFs, standard Python parsing for structured formats.
4. Structure the parsed results into tables, each traceable to the Project ID.
5. Extract complementary records from the firm's other systems (Dynamics CRM, Abacus ERP, Asana project management), joined on the same identifier.
6. Consolidate everything into a single master table feeding one Power BI dashboard.

In practice, the server scan came first and set the pace. The server holds far more than project files, so the work began with archived projects only, both to bound the scope and to develop and test the scraping scripts on lower-risk material. Archived projects sit under a `Work_in_Progress` folder, one folder per project, each folder name beginning with the Project ID. The script first enumerated all project folders, then walked their subfolders to record the path of every file, producing a consolidated project-to-file map. This was the first structured view of the server the firm had ever held.

Two file types were then selected for parsing: procurement offers and the proprietary calculation files. The offers, in PDF form, carry project metadata such as client and location. The proprietary files come from Lesosai, the energy and ecological-balance software used across Durable's projects, and hold building categories, embodied and operational greenhouse gas values, and location data.

Lesosai is a closed system: its results must be extracted manually, no central log of outputs exists, and not all projects retain end-of-project documentation. The single most valuable source of business-relevant data is therefore the one least accessible to automated scripting. This is not a technical detail; it is a direct consequence of the firm's absent data governance, and it shaped what the prototype could and could not recover in a limited time frame.

Four internal sources fed the final dataset: the project file server (the primary map), the Lesosai files, the procurement PDFs, and the Asana records, the last requiring a dedicated cleanup pass. The sources were selected on three grounds. They were internal, so no external coordination was needed and no client data left the firm. They are the foundation information on operational data for Durable's core business. With this information all other indicators and metadata on each project can be added during the next iteration.

Cleaning rules were fixed before large-scale processing began. A single folder, 00-Database, was defined as the single source of truth, with strict separation between raw, cleaned, and exported data. Duplicate project identifiers were repaired, and one identifier was kept coherent from the initial server scan through file copying and parsing into the final Power BI join. Where sources disagreed, most visibly on project location, values were merged with explicit provenance rather than forced into a single silent answer: fields carry provenance columns, confidence scores, and review flags, so that uncertainty is recorded rather than hidden. Filtering rules were audited after consolidation.

The resulting dataset found around 1,600 projects, more than 300 clients, and more than 700'000 files, consolidated into a master table built in four versions: an initial 30-column consolidation, a second pass improving client coverage against a richer export, a third adding Lesosai building categories, and a fourth adding project-size estimation.

3.4.4 The Prototype: finDUR 0.1

The resulting data tables were displayed in the form of a Power BI dashboard called `finDUR 0.1`. It consolidates project data from the server, Lesosai files, procurement documents, Asana, and Abacus around one shared identifier: the `Project_ID`. This made the data queryable in one interface. Users can filter projects by status, location, use, or client, and trace a single project across the different systems that hold information about it.

The dashboard was divided into four main views: Overview, Project-Control, Project-Finder, and File-Finder. The start page, shown in the following figure, gives access to these sections and presents the prototype as a navigable internal tool rather than a static report.

Before the prototype, this was not possible. The data already existed, but it was fragmented across separate systems and could not be queried as one portfolio. The prototype therefore shows that Durable's existing data has potential value once it is structured and connected. The resulting model is relational in structure, but not yet a designed database schema. It connects dated source tables around one central project table. The following figure shows the source tables connected around the central project table.

The architecture works, but the coverage remains incomplete. Business-relevant fields such as building use, location, floor area, and greenhouse gas values are only populated for part of the portfolio. Portfolio-level comparisons therefore rely on the populated minority. This limitation is visible in the Overview page in the following figure. The grey areas in the pie charts show the share of projects where no usable information was available.

The Project-Finder view, shown in the following figure, turns the dashboard from a summary page into a search interface. Users can search by `Project_ID` or project name, filter by work status, client, use, construction type, city, or SIA phase, and locate projects geographically. This view is important because it demonstrates the central function of the prototype: not only counting projects, but making fragmented project information searchable.

In the terms of Section 3.4.1, `finDUR 0.1` is not a finished database. It is an early-stage internal database prototype: partial, unvalidated, but functional. It tested whether a Durable employee, supported by GenAI, could create a working artifact from fragmented internal data under severe constraints: a limited time period, limited resources,

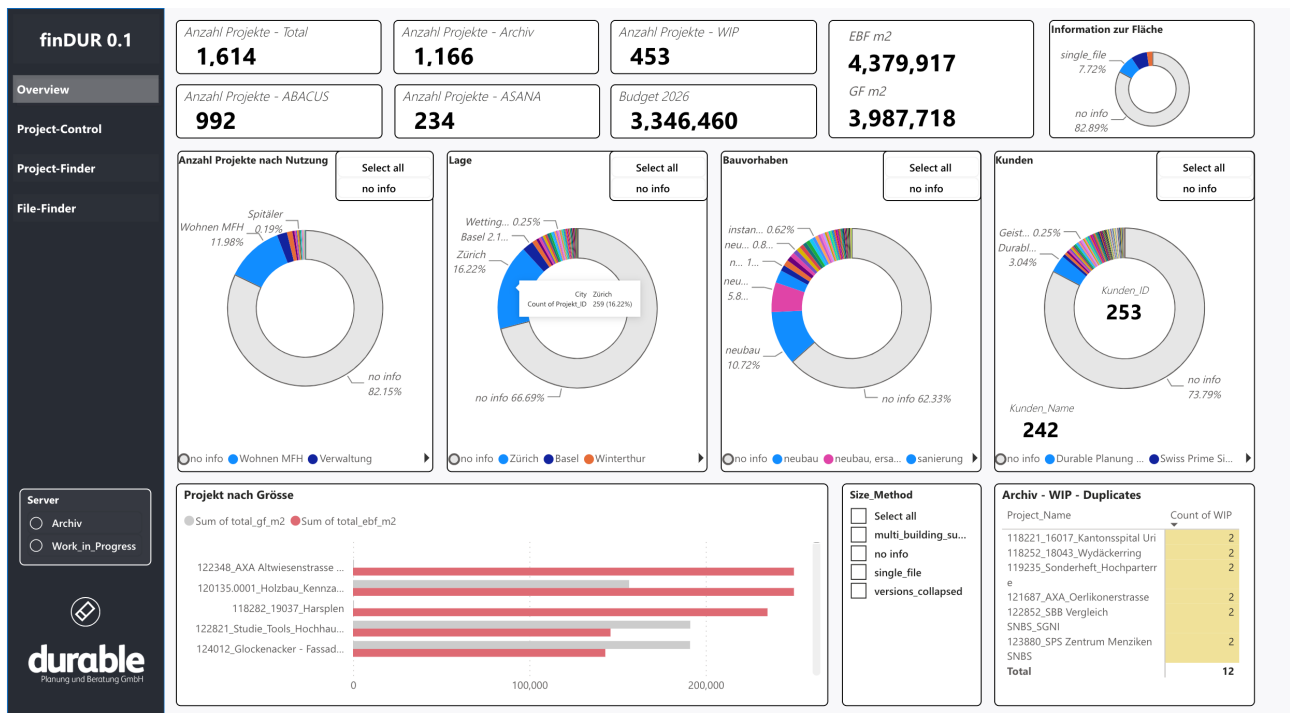


Figure 15: finDUR 0.1 overview dashboard and data coverage.

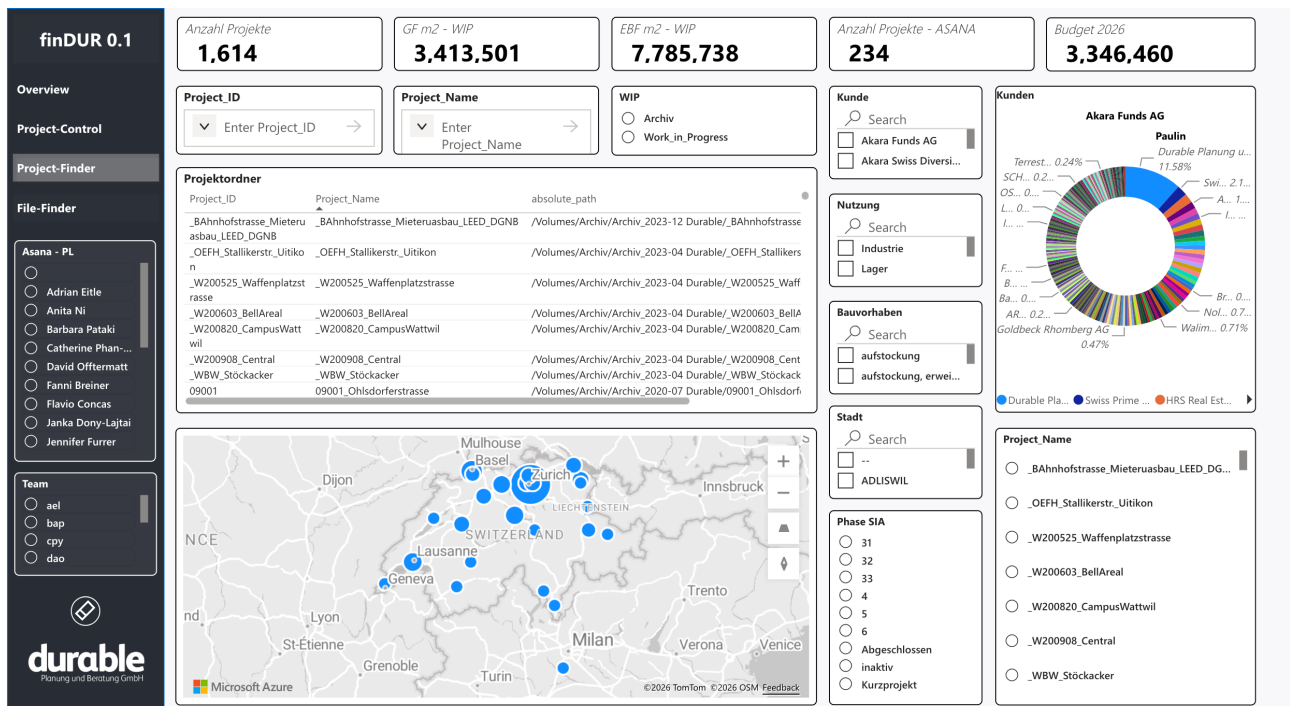


Figure 16: finDUR 0.1 Project Finder.

and development work carried out during the researcher's own master thesis time. It also exposed what minimum knowledge was required and which capability gaps remained visible after the attempt.

The dashboard is therefore only the visible output of the prototype. The more relevant finding lies in how it was built: which decisions were made, where GenAI helped, where it failed, and which parts still required human judgement. The next section reconstructs this development process and the decisions behind finDUR 0.1.

3.4.5 Building finDUR 0.1: Process and Decisions

The prototype was built through several short working sessions between 29 May and 16 June 2026. Each session closed with a practical decision, not a finished product.

The first step was not coding, but creating control over the project structure. The researcher's scripts, CSV exports, and folders had become difficult to follow. The first decision was therefore to define clear roles for the folders: 01-DUR-Bank as the working project folder, 00-Database as the single source of truth, and Power BI as a reporting layer that only reads final CSV exports.

The next sessions focused on keeping the data coherent. The central task was to preserve one stable Project_ID across the server scan, file copy, parser outputs, cleaned tables, and Power BI model. This became the core technical rule of the prototype.

The later sessions built the master table, Mother_Projects_Main. This table combined project metadata, client data, Lesosai information, location data, and project-size estimates. Errors only became visible once this table existed, for example missing projects, conflicting locations, or incomplete fields. The work was therefore not one script solving one problem, but a gradual stabilisation of folders, identifiers, sources, and tables.

Power BI was mainly chosen because it was inside the researcher's existing skill range. Tables can be joined through a common key without first building a separate database stack. This kept the work within the one-month development window and allowed the prototype to run in a sandboxed IT environment managed by Wüest Partner. A web-app tool was possible in theory, and GenAI could have supported it, but the one-month constraint made Power BI the safer choice. This shows a central finding: GenAI expanded the perceived options, but the final implementation still stayed close to existing means.

3.4.6 The Role of GenAI During the Process

This section evaluates GenAI by separating the prototype into the actual tasks required to build it. The aim is not to claim that GenAI helped in general, but to identify where it reduced the capability gap and where it did not.

The basis for this assessment is a consolidated analysis of the interaction logs, coding-session summaries, and voice recordings. After each coding session, GenAI was used to produce a structured summary of what had been attempted, what had changed, where difficulties appeared, and which decisions were made. The voice recordings were then transcribed and combined with these summaries into one consolidated log. This log was analyzed again with a GenAI agent to identify recurring task types and difficulty patterns. The result was checked by the researcher against his own perceived difficulties during the process.

The following table summarizes this distribution. It shows that GenAI contributed most where the task could be translated into instructions, scripts, or checks. Its contribution was weaker where the task required judgement, domain knowledge, validation, or a decision about what the data actually meant.

Table 12: Distribution of GenAI and human contributions during prototype development.

Task	What it involved	GenAI contribution	What remained human
Locating and collecting data	Finding usable data across an unstructured fifteen-year archive	Wrote the inventory scan scripts	Deciding scope; instructing the agent not to crawl the server
Cleaning and identifiers	Repairing duplicates, reconciling disagreeing sources	Wrote repair and merge scripts; proposed provenance conventions	Defining what counts as a duplicate and which source wins

Task	What it involved	GenAI contribution	What remained human
Defining useful variables	Building categories, project-size estimation, client fields	Implemented the definitions once specified	Choosing the variables and judging whether values made sense
Structuring the information	Single source of truth, raw/clean/export separation, archive rules	Proposed the governance rules and reframed cleanup as a governance problem	Ratifying the rules and enforcing them across sessions
Writing and debugging scripts	Python parsers, consolidation scripts, pipeline code	Strongest contribution: generated, executed, tested, and refined code iteratively	Recognizing when a fix targeted the wrong layer of the pipeline
Using GenAI effectively	Prompt scoping, token discipline, protecting client data, cross-checking models	None: this skill sat entirely on the human side	Learned during the process; a precondition, not an output
Connecting output to the dashboard	Keeping one identifier coherent from server scan to the Power BI join; building the report layer	Kept the identifier logic consistent across scripts	The Power BI layer itself, built from the researcher's prior skill
Validating the results	Testing outputs before touching the live server; auditing filters; manual corrections	None that could be trusted without checking	Entirely human: domain knowledge was the only available ground truth

The table shows that GenAI was most useful where the work could be translated into explicit rules. Once the researcher defined what had to happen, GenAI could write scripts, repair files, merge tables, and keep identifier logic consistent across the pipeline.

Its limit appeared where the task required judgement. GenAI could propose structures and code, but it could not decide which source was authoritative, whether a duplicate was real, whether a project category made sense, or whether an output was plausible in Durable's context. These decisions depended on the researcher's domain knowledge.

The main finding is therefore not that GenAI built the prototype independently. It accelerated execution and helped the researcher move through technical tasks that would otherwise have blocked the process. However, it did not replace scope-setting, data interpretation, validation, or responsibility for the result.

For the first research question, this means that GenAI partially compensated for missing technical capability, especially in coding and debugging. It did not compensate for judgement. The capability gap was reduced enough to produce a functional prototype, but it was not removed.

3.4.7 Management Evaluation: The Second Management Workshop

The prototype was presented on 16 June during an Executive Board meeting with the extended management group. The participants were: Adrian Eitle (Managing Director), Nicolás Guariento (Manager of Digitalization Team), Jörg Lamster (Founding Partner) and Barbara Pataki (Managing Director). The meeting was set up as a prototype review. It was part of a broader presentation by Adrian Eitle on the potential of digitalization and digital venturing at Durable. *findUR 0.1* was therefore inserted into a wider strategic discussion and used as a communication object to make the data opportunity visible.

This setting shaped the evaluation. The prototype could not be presented as a work in progress, with open technical questions, unresolved data gaps, and possible next iterations. Instead, because of the format and time constraints, it was presented as a finished product. The theoretical background and learning process behind it had to be shortened, and the presentation focused mainly on the data problem and the dashboard as visible output.

This frames the discussion and is important to mention because Durable operates under strict resource constraints. Internal tool development competes with billable consulting work, and any positive management reaction can imply a need for budget, ownership, and follow-up capacity. The workshop therefore did not only evaluate the prototype's usefulness. It also implicitly tested whether management was willing to release resources for another iteration.

The later feedback from Adrian Eitle and Nicolo Guariento should be read in this context. Their reaction was sobering and treated the prototype as ready for active use. Adrian's usefulness criterion was strict: a tool is valuable only if it simplifies a work step, shares knowledge, or reveals a relationship, not simply because it visualizes data well. Nicolo questioned whether the dashboard solved a concrete everyday problem and suggested that it would likely remain useful only for the researcher or a small group of users.

At the same time, this feedback reflects the position in which management was placed. Treating the prototype as successful would have implied the need for further budget and organizational commitment, which had not been secured. The prototype was therefore evaluated as if it were a completed deliverable, even though its research function was to expose what a first GenAI-supported iteration could achieve and where the next capability gaps would appear.

The finding is therefore limited but important. The prototype made Durable's data problem visible through a working artifact, but the workshop format did not allow it to be discussed as an unfinished development process. It moved the conversation from an abstract opportunity to a concrete object that management could question, but it did not trigger a resource commitment. At the time of writing, whether Durable will fund a second iteration remains open.

The final implication for the framework is that the effectual stakeholder commitment did not occur. The prototype created a possible new mean for Durable, but it did not become an organisational resource because management did not commit time, budget, or ownership to a second iteration. In effectual terms, the artifact remained outside the company's active means.

This is important because the value of the first iteration was not only the dashboard itself, but also the learning generated during the process: the data gaps, the minimum capabilities required, the limits of GenAI, and the decisions needed for a next cycle. Without stakeholder commitment, these learnings cannot yet be translated into routines, roles, or structures inside the firm.

The first iteration therefore succeeded as an experiment but failed to become embedded. It produced a working artifact and relevant learning, but not a committed organizational resource.

3.4.8 External Technical Evaluation: Software-Engineering Review

A separate technical workshop was conducted on 2 July with Prabhakaran Santhanam, an external Senior Software Engineer from Datahouse, Wüest Partner's sister company. The one-hour session was a one-to-one review assessing findUR 0.1 for scalability, technical rigor, data structure, and the shape of a possible next iteration. The first part introduced the research context, the prototype, and the role of GenAI. The second was an open technical discussion.

His feedback was positive but sobering. He confirmed the prototype as a credible minimum viable product and praised the clarity of its data structure. He was equally clear that the distance to a production application is large. findUR 0.1 has no database migration path, no automated testing, no dependency or license checks, no credential management, and no role-based access control. Power BI supplies some of this, but leaving it to build a multi-user interactive application requires a level of complexity beyond what a self-taught operator with GenAI can reach. These are baseline requirements for anything past a local, single-user experiment, not refinements.

The most useful part of the review for the thesis argument concerns how the expert's own role shifts. In Santhanam's words, the human in the loop needs to be an "expert-in-the-loop". The central point is that GenAI reduces the expert's workload rather than removing the expert, making him or her even more necessary. The change is that GenAI makes the expert-in-the-loop's task more relevant.

For a senior engineer, GenAI frees enough time that he can focus on tasks that were usually outside the scope of the software engineer. It pushes the role toward software architecture: looking at the project as a whole, judging whether it is production-grade, rather than assembling individual blocks. He illustrated this with a penetration test. A task he would previously outsource to an external firm at roughly 15,000 CHF he can now perform himself in about a week with AI support, reaching an internal-review standard he estimates at 90 to 95 percent, because he has read enough such reports to know what to look for and to close the last gap. The formal certification still requires the external license, but the internal review no longer does. His example suggests what this thesis will argue: that GenAI extends existing judgement, but does not create it on its own.

On staffing, Santhanam's position was cautious. A domain expert with some technical exposure can use GenAI to reach a prototype. A developer with AI can work with more independence. A junior developer with AI is not yet enough for a complete product, because AI does not close the judgement gap.

For him, the key role remains the senior engineer. The senior acts as architect, reviewer, and tutor, while AI absorbs lower-level implementation work. This matters because production experience cannot be skipped. Someone still needs to know what deployment, maintenance, bugs, security, and long-term reliability require.

His feedback confirms the same pattern as the management evaluation. The prototype works as a first iteration, but it is not yet an organisational capability or a production system.

Santhanam's concrete recommendations for a next iteration are summarised in the following table. They are recorded as an expert advice on what might be relevant in a certain time period, not as road map.

Table 13: External technical expert recommendations for the next iteration.

Implementation horizon	Concept	Recommendation	Reason
Next iteration	Data storage	Move from CSV to PostgreSQL	CSV exports are fragile and do not scale for repeated use
Next iteration	Pipeline	Build one orchestration script	Replace manual script execution with one controlled pipeline
Next iteration	Pipeline	Schedule extraction	Run extraction automatically, with validation before writing
Next iteration	Pipeline	Containerize crawler	Make extraction reproducible and easier to deploy
Next iteration	Testing	Add unit tests	Prevent code changes from silently breaking the pipeline
Next iteration	Testing	Create dev/test environment	Test changes on anonymized data before touching production
Next iteration	Code quality	Add linting standards	Keep generated and human-written code maintainable
Next iteration	Security	Manage credentials properly	No passwords or access keys should be stored in Git
Next iteration	Data model	Keep relational structure	PostgreSQL fits the current structured schema; JSON can be stored if needed
Later development	Versioning	Add migrations	Schema changes need controlled release and rollback logic
Later development	Security	Check dependencies and licenses	Avoid vulnerable packages and restrictive licenses
Later development	Documentation	Document pipeline and dashboard	Needed for handover and onboarding
If scaled	Security	Add role-based access	Required once several users or external users access the system
If scope expands	Compliance	Check GDPR exposure	Risk increases if personal data is collected
If scaled	Scalability	Use the database as the scaling base	Multi-user use depends first on proper database storage

3.4.9 Sub-cycle B: Design Principles

These design principles follow from the build. They are based on the learnings during the process of producing the prototype:

Table 14: Stage 2B design principles derived from the prototype build.

ID	Design principle	Rationale
S2B.1	Stay within the operator's knowledge range and use GenAI to extend it.	GenAI amplifies what the operator already knows; pushed beyond it, output cannot be judged and failure risk rises
S2B.2	Keep judgement human: meaning, source authority, validation	GenAI cannot decide which source is authoritative, whether a duplicate is real, or whether an output makes sense in the firm's context
S2B.3	Prior domain knowledge is a precondition, not an output	The operator could guide GenAI only because he already understood the firm's data, identifiers and systems
S2B.4	Route every output to a validation step the operator cannot skip	Functional output can still be wrong invisibly; the budget data pulled from Asana instead of Abacus ran without error yet was incorrect
S2B.5	Keep AI tasks narrow to preserve control and reviewability	Larger tasks increase the size and opacity of the output that later requires review, especially for a non-expert operator
S2B.6	Build skills through the process, not before it	Each interaction created the prior knowledge that made the next more productive; capability was produced by the work, not by training
S2B.7	Use GenAI to make ideas tangible fast; move from discussion to artifact	GenAI produces visual and functional inputs quickly enough for the organization to react to something concrete rather than abstract intentions

3.5 Stage 3: Reflection and Learning

In ADR, reflection and learning is a continuous stage that runs alongside building, intervention, and evaluation (Sein et al. 2011). Its task is to move the findings from the specific problem instance to the broader class of problems it belongs to. In this case, the researcher annotated decisions, difficulties, errors, and reflections throughout the development of finDUR 0.1. These notes were part of the build.

This section organizes those reflections and stays close to the Durable case. It shows where GenAI helped, where it did not, which decisions mattered, and which limits appeared. The reflections also show how the literature review became visible in practice, especially through existing means, organizational constraints, absorptive capacity, and stakeholder commitment. The section therefore reflects on both the researcher's learning curve and the organizational limits that slowed adoption. The generalizable insights and design principles are formalized in Stage 4.

3.5.1 Working Under Constraint

The BIE phase was shaped by the same resource constraint the thesis studies. The artifact was built while the researcher worked a 90 percent job, with most development time taken from evenings and thesis time. No separate budget existed, management attention was limited, and the workshop time could not extend beyond one hour. The rhythm of the work was also dictated by fixed meeting dates and the thesis submission deadline, which kept the development window especially tight.

The build was therefore not a clean test of what GenAI can do under ideal free experimental conditions. It was a test under the real conditions of a constrained SME, where innovation time competes with billable work, management attention is scarce, and deadlines shape what can realistically be attempted.

This context explains several later decisions. The researcher stayed close to existing means, especially Power BI, because it was the tool he already knew and could use within the available time. So the experimentation was also limited by time and resources. Although GenAI expanded the perceived technical options, it did not remove the pressure to choose the safest available path.

3.5.2 What the Build Showed

The build separated execution from judgement. GenAI supported execution: it wrote scripts, parsed files, merged sources, repaired tables, and proposed governance rules. This allowed one employee without formal software training to reach a functional prototype without hiring or outsourcing.

Judgement remained human. The researcher had to decide which data mattered, which source was authoritative, whether duplicates were real, and whether outputs made sense in Durable's context. This was possible only because he already understood the firm's data, project identifiers, internal systems, and Power BI. Prior domain knowledge was therefore not optional.

The researcher deliberately kept the AI tasks narrow. He did not push GenAI as far as it could theoretically go, because he wanted to keep control over the product and understand each step. The tasks given to the agents were often technically simple: write a script, repair a table, merge files, or check an identifier. The hard work was deciding what should be extracted, where the information belonged, and how the sources should be organized.

This process remained trial and error. GenAI accelerated experimentation by making it possible to test, adjust, and repeat steps quickly within the tight timeframe.

The process accelerated learning. The interaction logs show how early prompts asked for basic fixes, while later prompts asked for diagnosis, source-of-truth rules, and uncertainty handling. The researcher learned to treat AI output as a proposal rather than ground truth and to cross-check answers across agents. GenAI therefore helped build capability, through repeated guided interaction.

Across the process, the researcher acted as a gatekeeper in the sense of Cohen and Levinthal (1990): he monitored an external input, translated it into Durable's context, judged the output, and made it usable for the firm. This role also included communicating the work to management: building presentations, simplifying technical concepts, and explaining why the prototype mattered. The relevant profile was therefore not a software engineer or a passive end user, but a technically literate domain expert who could guide, judge, and translate the process.

3.5.3 What the Evaluations Showed

Evaluations were positive in tone, but there was no commitment. This is itself a finding. The prototype was developed under severe resource constraints, partly outside normal working hours, and represented a significant amount of internal exploration. However, the management discussion did not focus on the prototype as part of an iterative learning process. It focused mainly on whether the visible artifact justified further budget investment, staffing, or strategic attention.

This shows the lack of slack for innovation inside the firm. Even when a working prototype was available, the organization did not have the time, budget, or shared language to evaluate it as a step in resource-building. The artifact was judged mainly as a result, not as part of a broader system that included the theory, the learning process, the data pipeline, and the organizational capability gap it exposed.

A further limitation was that management mainly saw the visible layer of the artifact: the Power BI dashboard. The underlying scripts, extraction logic, cleaning routines, and data structures were harder to communicate in the available time, although they also represent new internal resources. These components could potentially be reused for other tasks, but they were not part of the management discussion.

This shows another form of limited absorptive capacity. Without shared technical vocabulary and enough time for exploration and explanation, the organization focused on the dashboard as a finished object rather than on the broader capability created during the build. The discussion therefore stayed at the level of visible output and budget implications, while the less visible learning, scripts, and reusable tools received little attention.

The technical review confirmed a different but related boundary. The prototype was credible as an MVP, but not close to production. It lacked a database migration path, automated testing, access control, credential management, and other baseline requirements. The reviewer confirmed that a domain expert with some technical grounding can reach a prototype with GenAI, but not a production system without expert supervision.

The main implication is that the gatekeeper role reached an organizational limit. The researcher could identify the opportunity, build the artifact, translate technical ideas, and make the data problem visible. However, without management readiness, shared language, and committed resources, this translation did not become organizational absorption. In Cohen and Levinthal's terms, individual absorptive capacity was present, but the organization did not yet absorb the learning.

The first iteration therefore succeeded as an experiment but failed to become embedded. It produced a working artifact and relevant learning in a very limited time, but not a committed organizational resource. The next cycle would need a clearer owner, committed resources, and a process designed for discussion rather than presentation. The following table consolidates the principal learning from the first iteration.

Table 15: Consolidated learning from the first prototype iteration.

Learning	What it shows
The build reproduced the firm's resource constraint	The prototype was built under the same lack of slack the thesis studies: limited time, no separate budget, scarce management attention, and tight deadlines.
GenAI compensated execution	It helped write scripts, parse files, merge sources, repair tables, and propose basic governance rules.
GenAI did not compensate judgement	Decisions about meaning, source authority, duplicates, validation, and usefulness remained human.
Prior domain knowledge was necessary	The researcher could use GenAI because he already understood Durable's data, identifiers, systems, and Power BI.
The AI tasks were deliberately kept narrow	The researcher did not push GenAI to its limit, because keeping control and oversight mattered more than maximizing automation.
Learning happened through trial and error	GenAI accelerated experimentation by making it possible to test, adjust, and repeat steps within the tight timeframe.

Learning	What it shows
The gatekeeper role became visible	The researcher translated between GenAI, Durable's data, technical concepts, and management.
The organization did not absorb the learning	Management saw mainly the dashboard, while the scripts, extraction logic, cleaning routines, and reusable tools remained less visible.
Lack of shared vocabulary limited discussion	Without time and common technical language, the broader capability behind the artifact was difficult to explain and evaluate.
Management did not commit	The prototype became a credible object for discussion, but no time, budget, or ownership was released for a second iteration.
The technical review confirmed the boundary	The prototype was a credible MVP, but production use would require database migration, testing, access control, and expert supervision.
Persuasion competed with continuation	Because the prototype also had to convince management, effort shifted toward presentability rather than a process others could continue.
The first iteration succeeded but did not embed	It produced a working artifact and relevant learning, but not yet an organizational resource.

3.5.4 Limitations

The scope of these conclusions is narrow and must be stated. This is a single case, with a single practitioner, studied over a short window of time and with only one iteration. The researcher was embedded in the organization and was himself the operator whose learning the study observed. This dual role gave access that an external observer would not have had, but it also introduces bias. The researcher's own expectations, prior knowledge, and interest in the prototype's success shaped both the build and the interpretation of the results.

The evaluation outcomes must therefore also be read critically. Management's lack of commitment may reflect limited organizational absorptive capacity, but it may also reflect weaknesses in how the researcher communicated the prototype, framed the workshops, or separated the artifact from the broader learning process. The prototype may have been presented too much as a finished object and not clearly enough as an unfinished iteration. This limits the strength of any claim about management resistance.

The minimum threshold identified here is also calibrated to one person's prior knowledge. It is difficult to separate what was generally required from what this specific researcher already knew. The prototype was not deployed and was not tested with a broader user group. It was only demonstrated to management and reviewed by one external expert. No colleagues used it independently, and no structured user feedback was collected. As a result, nothing can be said about maintenance, degradation, everyday usability, user experience or interface.

The evaluations were collected from two managers inside one firm and one external expert. The gatekeeper profile is inseparable from the researcher, who was also the practitioner and operator. Generalizing from this cycle therefore requires two things the next iteration must test. First, separating the gatekeeper role from the researcher role, to better observe the basic knowledge the gatekeeper needs and to allow more neutral observation of the results. Second, exposing the artifact to a real user group rather than a single management pitch.

3.5.5 Design Principles Stage 3: Reflection and Learning

Table 16: Stage 3 design principles from reflection and learning.

ID	Design principle	Rationale
S3.1	Keep an expert in the loop	GenAI reduces the expert's workload but makes the role more necessary; the expert shifts from writing code to judging whether the system is production-grade
S3.2	Staff with a senior engineer as orchestrator and tutor	A junior developer with AI is not enough for a complete product; AI does not close the judgement gap, so production experience cannot be skipped
S3.3	Validate relevance internally, technical quality externally	Internal stakeholders judge usefulness and fit; only an external technical expert can assess scalability, security, and maintainability
S3.4	Separate prototypes from production systems	The prototype was credible as an MVP but lacked database migration, testing, credential management, and access control, all baseline requirements for production
S3.5	Secure minimum slack	Below a minimum slack threshold GenAI cannot compensate, because its value depends on human time to guide, validate, and integrate outputs (Nohria and Gulati 1996)
S3.6	Design the process for continuation, not persuasion	When the prototype must also convince management, effort shifts toward presentability rather than a process others can continue
S3.7	Individual absorptive capacity is not organizational absorptive capacity	The operator learned faster than the firm; without shared language and committed resources, individual learning did not become organizational absorption (Cohen and Levinthal 1990)

4.0 Stage 4: Formalization of Learning

This thesis asked whether Generative AI can compensate for missing organizational expertise in a resource-constrained, non-technology-oriented SME, and how such compensation can be implemented under an effectual logic. Both questions were examined through one complete Action Design Research cycle at Durable Planung und Beratung GmbH, comprising an innovation framework, a resource-mapping workshop, the development of the findUR 0.1 prototype, and internal and external evaluations. This chapter corresponds to Stage 4 of ADR (Sein et al. 2011). It formalizes the learning from the case, answers the research questions, presents the resulting design principles, and defines the scope and limits of the findings.

4.1 Answer to the First Research Question

To what extent can Generative AI compensate for missing organizational expertise required for digital venturing in a resource-constrained, non-technology-oriented Small Enterprise?

The answer is: only to a limited extent, and only under specific conditions.

The framework developed in Stage 1 already gives clear limits to what compensation can mean. Under an effectual logic, bricolage expands the firm's means by recombining existing data, tools, and self-taught skills into new resources (Baker and Nelson 2005). GenAI enters this framework as a possible input for recombination, not as a source of innovation in itself. It sits outside the firm's existing means: it is not internalized as a skill, and it is supplied by external commercial providers the firm does not control, which makes over-reliance a strategic risk.

Absorptive capacity sets the theoretical limit. External knowledge cannot be used without related knowledge inside the firm (Cohen and Levinthal 1990). GenAI can therefore not compensate where the firm lacks the minimum ability to recognize, guide, and evaluate its output. The framework remains narrow by design. It does not examine wider factors such as corporate identity, agility, innovation culture, or prior innovation experience.

The empirical work confirms it. GenAI can compensate for parts of execution-level expertise. AI coding agents supported script writing, data structuring, explanation, and pipeline design. Within three weeks and about eight full working days, one employee without formal software training produced a functional dashboard covering more than 1,600 projects and 700,000 files. Under conventional resource logic, this work would have required hiring, outsourcing, or dedicated internal coding expertise, even only to make the Python scripts work as intended. In this narrow sense, the compensation is real.

However, this compensation did not remove the software expertise gap. GenAI supported action, but it did not replace judgment. As Section 3.4.5 documents in detail, decisions about architecture, data structure, variable definitions, validation, what not to build, when an output was wrong, and when a script was good enough remained with the human operator throughout. The complexity of the artifact therefore stayed at amateur level. The quality of the output still depended on the task the tool was given, and the task depended on domain knowledge the tool could not provide: knowledge of the firm's data, its project identifiers, and its internal logic. The idea of building a database came from a human workshop, and that decision was not facilitated by GenAI.

The external technical review confirmed the boundaries. The prototype was credible as a testing prototype, but the distance to a production-ready application remained impossible to close. This would require expertise the operator did not possess and that no one in the firm has. The coding agents did provide guidance when asked about best practices, but this guidance remained difficult for the user to judge as he had not seen a best practice before. Nevertheless, the agents helped the researcher understand problems faster and reduced the learning curve. Each small interaction created knowledge and experience.

This is a key positive finding. GenAI did not only produce code. It supported learning. The researcher could ask questions, test alternatives, receive explanations, and move from abstract uncertainty to a tangible artifact within a short period. T

A second positive finding is that the company also gained something concrete to evaluate: instead of discussing digital venturing through abstract use cases, management could see a working representation of what an internal data tool might look like. What the Gen AI facilitates especially well is the fast production of visual and functional inputs. It helps move from discussion to functional artifact. From static intentions to testing a prototype.

GenAI makes sketches tangible by producing prototypes, dashboards, and scripts quickly enough for the organization to react to them. This does not mean the artifact is reliable or scalable. It means the firm can learn from something visible instead of only talking about what might be possible.

The compensation is therefore conditional. It depends on prior domain knowledge, practical AI literacy, and the presence of an internal gatekeeper who can guide the tool, judge the output, and translate it into the organization. It also depends on management commitment and a willingness to treat prototypes as learning devices rather than finished products.

The answer to the first research question is therefore precise: GenAI does not replace missing expertise. It lowers the threshold for action and partially substitutes technical labour, but not technical judgement or experience. The main constraint shifts from producing outputs to verifying, interpreting, and integrating them. Durable could attempt more with GenAI, but it could only trust what its internal capabilities allowed it to understand and explore only what its existing knowledge made visible.

4.2 Answer to the Second Research Question

How can Generative AI be implemented in a non-technology-oriented Small Enterprise to support an effectual approach to digital venturing?

The answer is: Generative AI should be implemented as an external accelerator and a source of capability building, not as a source of innovation. Innovation still requires human judgement, organizational readiness, and resource building, which GenAI cannot supply on its own. The empirical research demonstrated this. GenAI was never pushed close to its limits; the user and the organization were. The capacity to leverage the technology therefore rests largely on the user. As the technology improves, the gap between an expert directing coding agents and a digitally illiterate organization may widen sharply.

The experiment tested the framework under realistic conditions: a resource-constrained SME with little internal software expertise, limited slack, and a clearly defined affordable loss. Using existing project data, subscription-based tools, and his own limited digital skills, the researcher produced a working prototype in about eight full working days across three weeks. The external technical review confirmed that the result was credible as a Prototype, although still far from production-ready.

Within these limits, GenAI worked well as an accelerator. It lowered the barrier to producing working code, supported rapid experimentation, and helped the researcher learn while building. The result does not show the full capability of GenAI, but what one non-expert user could achieve under severe constraints. Whether the perceived learning represents lasting capability would need to be tested in another cycle or with another user.

The limits to this implementation are arguably the most important findings. The central issue is slack. The prototype was built largely on borrowed time: evenings, thesis hours, and a single one-hour workshop. There was no budget, ownership, or released budget. The organization did not commit resources, did not absorb the learning, and evaluated the artifact mainly as a finished product rather than as one step in an iterative process. It was then set aside.

This supports Nohria and Gulati (1996) at the implementation level. Below a minimum slack threshold, GenAI cannot compensate, because its value depends on human time to guide, validate, and integrate its outputs. The operator can temporarily absorb the gap through overtime, but this is neither sustainable nor scalable. It also excludes decision-makers from the construction of the tools they may later be asked to fund. Without management involvement, a prototype is unlikely to become an organizational resource.

The slack constraint therefore sets the real precondition. The implementation question is not whether GenAI is good enough. The findings show that the main issue lies in how the organization prepares itself to use it. The firm must release enough capacity for someone to guide the tool without relying on time taken from billable work, and decision-makers must be involved throughout the process rather than only at the end.

This is the central finding of the experiment. Implementation depends on the organization. Generative AI did not remove the established problems of technology adoption; the same requirements for time, learning, coordination, and management commitment still apply.

The case supports the framework’s basic logic. It showed that a resource-constrained firm can recombine existing data, limited internal skills, and GenAI to produce a functional prototype within a short period. It also confirmed that this only works when the scope remains small, the operator can judge the output, and the artifact is treated as part of a learning process rather than as a finished product. The framework is therefore considered a valid abstraction of the process followed in this thesis and of the challenges that other non-technology-oriented SMEs may face when entering digital markets.

The experiment strengthened and refined the framework by showing which steps worked in practice and where additional conditions, especially slack, expert support, and stakeholder commitment, were required.

The innovation framework points to a straightforward implementation sequence. First, determine what the firm has at hand and where its strengths lie, then identify the resources it is overlooking. From that pool, select a gatekeeper, ideally one with sufficient domain expertise. Map the gaps against the best-case scenario, the causal benchmark, and attempt to build something useful for the company. Build a first iteration. Do not judge it on functionality alone; treat it as a learning process for the operator, management, and the organization. Involve internal stakeholders to test the results, and iterate on usability and on the idea itself as far as resources allow. Once the results hold, a technical expert must support the next technical iteration. Only when technical and stakeholder assessments align should the artifact be implemented across the organization and, eventually, considered for the market.

4.3 Design Principles

Stage 4 formalizes the generalized learning from the ADR cycle. The principles developed in Stages 1, 2, and 3 described the design requirements and findings of each stage. The following table consolidates these stage-specific principles into a broader set that extends beyond the BIE cycle and the Durable case. Together, they address both research questions by defining how GenAI can support digital venturing in a resource-constrained, non-technology-oriented SME and under which conditions that support remains useful. A central principle is that GenAI should be treated as an external accelerator for capability-building, not as the source of innovation itself.

Table 17: Consolidated design principles from the ADR cycle.

ID	Design principle	Rationale
S4.1	GenAI serves as an external accelerator for capability-building	GenAI accelerates execution, experimentation, and learning, but it does not define the opportunity or replace internal capability-building
S4.2	GenAI extends the operator’s reach within their knowledge range	GenAI amplifies what the operator already knows; beyond that range the output cannot be judged and the risk of undetected failure rises
S4.3	Tasks must stay reviewable and the process must be documented continuously	Small, sequential tasks allow outputs to be checked throughout. A record of prompts, decisions, errors, and tests keeps the process traceable and transferable
S4.4	Judgement remains human and validation must be made unavoidable	Decisions about meaning, scope, source authority, plausibility, and usefulness stay with the operator. Functional output can still be wrong in ways the operator cannot see
S4.5	GenAI renders ideas tangible quickly	Fast production of visual and functional artifacts lets the organization react to something concrete rather than to abstract intentions

ID	Design principle	Rationale
S4.6	AI-built artifacts should be treated as prototypes	Functional output is not reliable or production-ready. A credible prototype still lacks database migration, testing, credential management, and access control, all baseline requirements that need external review before deployment
S4.7	An expert should be included in the loop as reviewer and educator	A senior expert can identify unknown requirements, assess production readiness, guides the operator, and builds technical understanding inside the firm
S4.8	Existing means and constraints are mapped first	The firm's data, tools, skills, relationships, time, and organizational limits are identified before deciding what to build
S4.9	A minimum of slack is secured and the first step stays within affordable loss	GenAI depends on human time to guide, validate, and integrate outputs. Below a minimum slack threshold it cannot compensate (Nohria and Gulati 1996). The first iteration remains small enough to fail without major cost
S4.10	The first artifact should be for internal use	A low-risk internal tool creates practical value while supporting experimentation and learning
S4.11	A technically literate domain expert holds the gatekeeper role	The gatekeeper connects domain knowledge, GenAI, technical experts, management, and the wider organization
S4.12	The process is designed for continuation and organizational absorption	A prototype becomes an organizational resource only when the process is built for others to continue, ownership is assigned, and resources are committed. Individual absorptive capacity is not organizational absorptive capacity (Cohen and Levinthal 1990)

5.0 Recommendations and Outlook

The main goal of this research was practical as well as theoretical. It aimed to address a real problem inside a real firm and contribute to theory through that process. The effort was therefore never only about delivering a working dashboard. It was also about capturing the learning from a full ADR cycle and leaving Durable with practical knowledge and actionable recommendations for using GenAI in future digital venturing and innovation initiatives.

The dashboard was the visible output, while the reusable value lies in what the process revealed: where GenAI helps, where it does not, what the operator must know, and what the organization must provide. This chapter translates that learning into recommendations Durable can act on, regardless of whether the prototype continues.

5.1 General Remarks for Durable

The strongest finding of this thesis is that GenAI's capacity to compensate for missing resources depends less on the tool than on the firm and its users. The title itself points to this: *"Leveraging Generative AI to Enable Digital Venturing in Resource-Constrained, Non-Technology-Oriented Small Enterprises"*. The emphasis of this research was on leveraging. The central question was never whether GenAI is capable enough, but whether firms are prepared to use its capabilities, and whether they can leverage this new technology at all. Across the months of this research the technology was never pushed to its limit. The user and the organization reached theirs first. For firms such as Durable the main constraint is therefore not the tool, but the organizational readiness required to guide and absorb it. Digital venturing enabled by GenAI is ultimately a firm-level adoption problem.

For non-technology-oriented firms such as Durable, the ceiling of coding agents and agent orchestration is largely irrelevant, because the binding constraint remains human: knowledge, judgement, time, and software experience. The recommendation that follows is therefore not about choosing a better tool. It is about building the capabilities the tool cannot supply, and beginning to reduce the unknown-unknowns in data management and digital technologies.

This is why adoption should be taken seriously now, not later. If the tool keeps improving and the constraint is absorptive capacity, the gap between a prepared firm and an unprepared one widens at the rate the tool improves. Falling behind is not linear for firms that truly manage to push the tool to its limits will certainly be able to build tools faster and find new business models. Firms that do not will be locked out.

This is not a recommendation to build chatbots, deploy autonomous agents, or automate work with AI applications. For a firm like Durable, that would be premature. The narrower recommendation is to use coding agents to explore ideas quickly, build simple prototypes, and develop internal skills. In the innovation process, GenAI is most valuable in the divergent phase: opening options, testing possibilities, and turning abstract ideas into working artifacts that management and employees can evaluate and improve.

This exploratory use still requires caution. Coding agents rely on external providers, which creates risks around pricing, dependency, and data protection. Their outputs are probabilistic and require sufficient knowledge to review. Durable should therefore use them where their exploratory value is high and errors remain cheap and visible, while keeping judgement, sensitive data, and validation inside the firm. This cautious use can help build the internal skills and data foundations needed before more advanced applications become realistic.

Durable will not become a "digital" firm. Until the firm builds up its data stack and organizes its own information. There is no point discussing chatbots or agents automating work. That destination is miles away. But the basis has to be built, and it has to be built now. The tool has made starting cheap for the first time. Starting is the whole recommendation.

5.2.1 Recommendations on finDUR and a Database

The first recommendation is to make a clear decision on whether to continue, redirect, or close the initiative. Leaving the prototype without committed resources risks losing the capability and learning developed during the process. Closing or changing direction should not be treated as failure. The prototype has already generated useful knowledge, and that learning can inform future initiatives even if finDUR itself is not continued.

Regardless of whether finDUR is continued, redirected, or closed, the cycle should end with a formal debriefing. Its purpose is to capture the learning, document the resources created, and ensure that the knowledge does not

remain only with the researcher. This thesis contributes to that process, but it should be complemented by an internal discussion with management.

The first task is to clarify what the prototype actually is. The dashboard is only the interface and can be replaced or redesigned. The prototype includes also the all the extraction scripts, cleaned tables, data structures, and governance rules behind it. These elements were difficult to communicate during the evaluation, which contributed to the artifact being judged mainly as a finished dashboard. Any decision about its future should therefore assess the full prototype and the learning it produced, not only its visible layer.

5.2.1.1 If Durable decides not to continue with finDUR This should be treated as a legitimate outcome rather than a failure. Once management has a clear understanding of what was built, the decision may simply show that this particular direction requires further reflection or does not fit the firm's current priorities.

The learning from the cycle should still be retained. A new workshop could revisit the opportunity space and identify smaller, faster experiments. Future iterations should reduce scope, shorten development cycles, and test ideas earlier. Lean Startup, scientific entrepreneurship, and similar experimentation frameworks could help structure this process. The scripts, data structures, and lessons from finDUR should also be documented and reused where relevant, so the work contributes to future initiatives even if the prototype itself is discontinued.

5.2.1.2 If Durable continues with finDUR: Continuing finDUR does not mean committing to the current dashboard. It means building on the data extraction and structures already created. The priority should be identifying which project values are most useful to Durable and improving how they are extracted from the server. The dashboard is only an interface and can be replaced or reconfigured with relatively little effort once the underlying data is available.

The main difficulty remains the extraction process, especially from proprietary software for which no automated path exists. This work does not necessarily require a senior software engineer. The cycle showed that a gatekeeper supported by GenAI can continue developing the extraction pipeline.

However, the gatekeeper should not work alone for an extended period. The next iteration requires a senior domain expert as a sparring partner and internal client. This person should define what the data must answer, assess whether the outputs are useful, and prevent the development from moving too far in one direction. The next iteration therefore needs two owners: one responsible for the AI-supported development, and one responsible for relevance and validation. Both require a small recurring time budget.

Continuation also means accepting that capability-building takes time. Progress will be incremental across skills, infrastructure, governance, routines, and shared language, and it will require resources before it produces direct returns. The immediate priority is therefore not to build an application, but to establish data governance. Durable first needs to define which data is worth collecting and under which standards.

For example, Lesosai results could be exported at a defined project phase, in a fixed format, and under a consistent naming convention. This would allow scripts to find and process them automatically without adding substantial work for employees. At this stage, standards that make existing work machine-readable are more valuable than a new database interface for manual input.

A last suggestion is, the dashboard should be opened to the wider DUR team, not as a finished product, but as a way to collect reactions, questions, and ideas. This would provide a low-cost basis for deciding which data and functions should be developed next.

Table 18: Recommended actions, conditions, and first steps.

Action	What it means	Conditions	First actions
Debrief, required first	Establish a shared understanding of what was built. The structured data, scripts, and governance rules are the prototype; the dashboard is only one interface.	One session involving the researcher and management.	Review the extraction pipeline, data structures, governance rules, reusable scripts, and lessons from the cycle.
Close or redirect	End finDUR as the chosen direction while preserving its learning and reusable resources.	Management concludes that the current direction does not fit the firm's priorities or available resources.	Archive the scripts and data structures, document the lessons, and use them to inform smaller or different future experiments.
Continue	Continue developing the extraction process and data foundation, not the current dashboard.	Two owners: one gatekeeper responsible for AI-supported development and one senior domain expert responsible for relevance and validation. Both require recurring time.	Identify the most valuable project data, improve extraction from proprietary software, define data standards, and open the dashboard to the wider DUR team for feedback.
Establish governance	Make existing project information consistently machine-readable before building a new application.	Agreement on which data should be collected, when it should be exported, and how it should be named and stored.	Define export rules for sources such as Lesosai, including project phase, format, location, and naming convention.
Test with users	Use the current dashboard as a feedback tool rather than as a finished product.	Access for a broader internal user group and a clear method for collecting reactions and questions.	Ask employees which questions the data should answer, which fields are missing, and which views would support their daily work.

5.2.2 Profile of Future Hires and Gatekeepers

One of the most interesting findings of this thesis is the identification of a staffing model for GenAI-supported digital initiatives. The development of finDUR revealed not only a capability gap, but also a clear division of roles. If the initiative continues, the findings suggest assigning three roles: a gatekeeper, a technical expert, and a senior domain expert acting as effectual stakeholder able to commit resources.

A second iteration would benefit from defining and assigning these roles from the start. The technical expert, ideally a senior software engineer, can initially be sourced externally once internal commitment is secured. If Durable decides to pursue digitalization more actively, the findings support hiring this expertise internally. This would reduce dependence on a single initiative and allow several digital opportunities to be explored in parallel.

The Gatekeeper: A Technically Literate Domain Expert (Practitioner)

The first role formalizes what the researcher performed informally. The gatekeeper needs to be able to build prototypes effectively with the help of Coding Agents. The gatekeeper needs to make the technical decisions, and produce the working artifacts: what the practice now calls vibe coding, building through natural-language instruction rather than writing code directly. Validating outputs and explaining the results to management are also part of the role, but the main task is to build and develop the artifact.

The gatekeeper role requires a small recurring time allocation and should be performed by a domain practitioner or expert alongside their normal consulting work. The role must sit between the technical system and the business context, with enough process knowledge to guide decisions independently, assess outputs, and translate technical possibilities into relevant applications. Its part-time nature helps keep the role grounded in current projects and operational needs.

This is why the technical grounding is not optional. Vibe coding produces something usable only if the operator has a basic working grasp of what sits underneath: how code is structured, how an IDE and a terminal work, what a script is doing, what a data pipeline is. Without this foundation the operator cannot judge the output, cannot redirect the agent when it fails, and cannot state clearly what the task should be. The literacy is what separates directing the tool from merely prompting it. The profile is therefore a technically literate domain expert, not a software engineer and not a passive user. Strong domain knowledge is a multiplier, while basic scripting literacy provides the technical foundation for the role.

The researcher's profile is the only directly observed case in this study and therefore serves as a practical reference point. Greater technical depth would likely improve the role, but a lower level may also be sufficient. Strong domain expertise, a clear idea of what should be built, and a willingness to learn may already provide an adequate basis.

The required technical knowledge is limited. It includes a basic understanding of coding agents such as Claude Code and Codex, including what they can do and where their limitations lie, basic Python scripting, and confidence using an IDE and terminal. Experience with Power BI or similar tools is also useful for understanding interfaces and data pipelines. In the Architecture, Engineering and Construction Industry context, experience with parametric tools such as Grasshopper may indicate similar computational thinking. Newer graduates increasingly bring some of these skills from their education, which may make the profile easier to develop over time.

The gatekeeper reaches prototypes, not production systems. Escalation to external technical expertise is a defined step, not a failure of the role.

The senior domain expert: effectual stakeholder and decision-maker

The second role is the one this cycle most visibly lacked. Management evaluated the dashboard because the dashboard was the only layer it could read. If custom internal tools begin to multiply, someone in management must be able to look beyond the user interface, speak about technical solutions at a basic level, and keep the growing tool landscape under control. Management up-skilling is a key condition for the firm absorbing what its gatekeepers produce.

This role also faces outward. A senior domain expert who understands clients and the market is the person who can present working prototypes to clients in the right frame, converting internal experiments into effectual stakeholder commitments (Saravathy 2001). Where one person combines management authority, domain expertise, and gatekeeper skills, the process becomes more direct. However, the three functions remain necessary, even if they are performed by fewer people. The four eye principle still holds.

The technical expert: expert-in-the-loop

If Durable commits resources to a broader digital transformation, access to senior software-engineering expertise will become necessary, whether through direct hiring, shared group resources, or periodic external support. The external review showed that production experience cannot be substituted by GenAI. While the technology can reduce the amount of routine coding, it increases the importance of expertise in architecture, infrastructure, quality assurance, and supervision. The senior engineer's role therefore shifts from implementing every component directly to establishing the conditions under which less experienced users can develop reliable solutions.

This expertise may become useful earlier than expected. Durable's first software applications would likely be simple internal tools with a limited purpose, operating in a controlled environment and without public access. With GenAI support, a senior engineer could develop and supervise this type of software relatively quickly and with limited risk. This would increase Durable's internal development capacity while also building the technical knowledge required for more advanced applications. If the firm later decides to develop an external product, the relevant expertise would already be available rather than having to be sourced under time pressure.

The development also works in the opposite direction. GenAI allows domain experts to move beyond their traditional roles and become prototype builders who can translate ideas into working artifacts. The technical expert is needed to ensure that this process remains controlled and that prototypes do not move beyond the level at which they can be safely understood and validated.

Together, the three roles address the gaps identified during this cycle. The gatekeeper builds and translates, the technical expert validates and teaches, and the senior domain expert defines relevance, makes decisions, and

commits resources. Each role responds to a limitation observed in the research.

Table 19: Recommended roles, functions, and boundaries for continued development.

Role	Profile	Function	Boundary
Gatekeeper	Technically literate domain expert: scripting basics, IDE navigation, Power BI, conceptual command of coding agents	Directs agents, validates outputs, translates to the organization, builds shared vocabulary	Prototypes, not production; escalates to the technical expert
Technical expert	Senior software engineer with production experience (hired, shared in group, or periodic)	Reviews, architects, teaches; unblocks others; exploits AI at full depth	Internal sandboxed tools first; external products only when ready
Senior domain expert	Manager or senior consultant with market knowledge and basic technical literacy	Decides, commits resources, controls tool proliferation, presents prototypes to clients	Governs and commits; does not build

5.2.3 Other Approaches: The Hackathon Proposal

A complementary recommendation is to use part of the internal training budget for a one-day GenAI prototyping workshop. The format could build on existing innovation workshops, but focus on using coding agents to turn internal problems into simple working artifacts.

This approach would address several limitations observed during the finDUR cycle. It would replace borrowed evening work with formally allocated time, involve management directly in the process, and create a defined space for experimentation. This would provide space for broad fast experimentation with minimum risk.

The workshop should operate within clear boundaries. The technical environment, access rights, templates, and basic instructions should be prepared in advance so that participants can focus on the problem rather than on software setup. This preparation should also introduce participants to coding agents such as Codex and Claude Code, as well as interfaces such as an IDE. The aim is to provide a practical demonstration of what GenAI coding tools can support and to let participants test these possibilities directly.

Participants should work in pairs or small teams that combine domain experts with basic technical literacy. The person who understands the operational problem should work directly with someone able to guide the coding agent. This creates the same combination identified in the staffing recommendation: a domain expert and a gatekeeper working together.

Evaluation should also be part of the workshop design. A software expert should review technical feasibility, while an internal stakeholder should assess usefulness and organizational fit. The evaluation criteria should be defined before the workshop so that prototypes are judged not only by their visible interface, but also by their data logic, learning value, and potential for further development.

Management participation is important. Decision-makers should be involved during the process rather than only evaluating the final output. This would help build shared vocabulary and organizational absorptive capacity.

The expected output should be one simple prototype per team. These prototypes should be treated as learning artifacts, not finished products. The most promising idea could then move into a second iteration with clearer ownership and resources. The workshop would not only generate ideas. It would also test whether the gatekeeper role can be developed internally and identify employees with the relevant combination of domain knowledge, technical interest, and willingness to learn.

5.3 Directions for Further Research

Four directions follow from this cycle, drawn from what it could not resolve.

The first is the research design itself. In this cycle the researcher, practitioner, and gatekeeper were one person. The next iteration must separate them: an observer who is not the operator, and an operator who is not invested in the

artifact's success. Only then can the gatekeeper threshold be measured rather than embodied. The same iteration should vary the operator profile: more domain expertise with less technical grounding, and the reverse. This case offers one working reference point; whether it marks a floor, a ceiling, or neither is an open question.

The second is the compensation construct. The case pulled it apart into execution, judgement, and maintenance. Execution compensated well, judgement did not, maintenance was never reached. Operationalizing these dimensions separately would let future work answer "to what extent" with more precision than one case allows.

The third is the framework. Effectuation and bricolage explain how to start under constraint, but they say little about how to test what gets built. Integrating Lean Startup and scientific-entrepreneurship principles (Ries 2011; Camuffo et al. 2020) into the innovation framework, with explicit hypotheses, short feedback loops, and predefined evaluation criteria, could tighten the build-test loop. This matters more with each tool generation: as execution gets cheaper, disciplined validation becomes the scarce activity.

The fourth is scope. This study can serve beyond replication. Its framework, workshop design, and design principles form a template for preparing resource-constrained SMEs for GenAI adoption: mapping means, selecting gatekeepers, and running first experiments within affordable loss. Applied across firms with different levels of slack and digital maturity, it would test how far the Durable findings generalize, and whether GenAI can accelerate not just one prototype, but a first digital transformation.

6.0 Closing Note

The main limit reached in this study was not the technology but the user and the organization. As GenAI improves, firms that build absorptive capacity early will benefit more from each new capability, while firms that delay will find the gap increasingly difficult to close. At the same time, acquiring basic digital skills and producing first prototypes has never been cheaper or faster. The practical implication is not to adopt every new tool. It is to begin building the human and organizational capacity required to use them.

Acknowledgements

I thank Durable Planung und Beratung GmbH for opening its data, supporting the research, and committing time to the workshops. I thank Prof. Dr. Robin De Cock for his supervision and insights throughout this thesis.

Declaration on the Use of Generative AI

During the works of this thesis, the author used Generative AI tools, including Claude and ChatGPT, to check language, improve readability, basic translation and refine the styling of the text. AI coding agents (Claude Code and OpenAI Codex) were used in the empirical work to build the prototype, as fully described in Chapter 3. The ideas, arguments, frameworks, and findings are the author's own and were not generated or copied from these tools.

One last reflection belongs here, because it complements the topics from this master thesis. GenAI accelerated and facilitated much of this work. The experience of writing the thesis confirmed the finding during the experiment and coding exercise. The tool amplified what I brought to it and helped have a faster broader view of many topics. Yet I remain sceptical of its usage. Delegating parts of the writing process created some quiet insecurity about one's own writings, a constant urge to verify and improve against the machine. The operator of this thesis, like the operator of finDUR, gained speed and lost some certainty. That trade off is the summary of what this technology currently gives and takes.

7.0 References

- Baker, Ted, and Reed E. Nelson. 2005. "Creating Something from Nothing: Resource Construction Through Entrepreneurial Bricolage." *Administrative Science Quarterly* 50 (3): 329–66.
- Banh, Leonardo, and Gero Strobel. 2023. "Generative Artificial Intelligence." *Electronic Markets* 33 (1): 63.
- Barney, Jay B. 1991. "Firm Resources and Sustained Competitive Advantage." *Journal of Management* 17 (1): 99–120.
- Barney, Jay, and Martin Reeves. 2024. "AI Won't Give You a New Sustainable Advantage." *Harvard Business Review*.
- Brunswick, Sabine, and Wim Vanhaverbeke. 2015. "Open Innovation in Small and Medium-Sized Enterprises (SMEs): External Knowledge Sourcing Strategies and Internal Organizational Facilitators." *Journal of Small Business Management* 53 (4): 1241–63.
- Camuffo, Arnaldo, Alfonso Cordova, Alfonso Gambardella, and Chiara Spina. 2020. "A Scientific Approach to Entrepreneurial Decision Making: Evidence from a Randomized Control Trial." *Management Science* 66 (2): 564–86.
- Chesbrough, Henry W. 2003. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Press.
- Chesbrough, Henry W., Wim Vanhaverbeke, and Joel West, eds. 2006. *Open Innovation: Researching a New Paradigm*. Oxford University Press.
- Cohen, Wesley M., and Daniel A. Levinthal. 1990. "Absorptive Capacity: A New Perspective on Learning and Innovation." *Administrative Science Quarterly* 35 (1): 128–52.
- Cyert, Richard M., and James G. March. 1963. *A Behavioral Theory of the Firm*. Prentice-Hall.
- De Massis, Alfredo, David Audretsch, Lorraine Uhlaner, and Nadine Kammerlander. 2018. "Innovation with Limited Resources: Management Lessons from the German Mittelstand." *Journal of Product Innovation Management* 35 (1): 125–46.
- Dell'Acqua, Fabrizio, Edward McFowland, Ethan R. Mollick, et al. 2023. "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality." *Organization Science*.
- Federal Statistical Office. 2024. "Small and Medium-Sized Enterprises." Swiss Confederation. <https://www.bfs.admin.ch/>.
- Feuerriegel, Stefan, Jochen Hartmann, Christian Janiesch, and Patrick Zschech. 2024. "Generative AI." *Business & Information Systems Engineering* 66 (1): 111–26.
- Fisher, Greg. 2012. "Effectuation, Causation, and Bricolage: A Behavioral Comparison of Emerging Theories in Entrepreneurship Research." *Entrepreneurship Theory and Practice* 36 (5): 1019–51.
- Holmström, Bengt. 1989. "Agency Costs and Innovation." *Journal of Economic Behavior and Organization* 12 (3): 305–27.
- IBM. 2024. "What Is Artificial Intelligence (AI)?" IBM Think. <https://www.ibm.com/think/topics/artificial-intelligence>.

- IBM. 2026. "The 2026 Guide to AI Agents." IBM Think. <https://www.ibm.com/think/ai-agents>.
- Janiesch, Christian, Patrick Zschech, and Kai Heinrich. 2021. "Machine Learning and Deep Learning." *Electronic Markets* 31 (3): 685–95.
- Lucchetti, Rafael, Emily Wu, Arjun Guha, Molly Feldman, and Craig Anderson. 2024. *Substance Beats Style: Why Beginning Students Fail to Code with LLMs*.
- McGrath, Rita Gunther. 1999. "Falling Forward: Real Options Reasoning and Entrepreneurial Failure." *Academy of Management Review* 24 (1): 13–30.
- Nohria, Nitin, and Ranjay Gulati. 1996. "Is Slack Good or Bad for Innovation?" *Academy of Management Journal* 39 (5): 1245–64.
- OECD. 2024. "SME Indicators, Benchmarking and Monitoring." <https://www.oecd.org/en/topics/sub-issues/sme-indicators-benchmarking-and-monitoring.html>.
- OECD. 2025. *Generative AI and the SME Workforce: New Survey Evidence*. OECD Publishing.
- Ries, Eric. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.
- Sapkota, Bhuwan, Konstantinos Roumeliotis, and Binay Karkee. 2026. *AI Agents Vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges*.
- Sarasvathy, Saras D. 2001. "Causation and Effectuation: Toward a Theoretical Shift from Economic Inevitability to Entrepreneurial Contingency." *Academy of Management Review* 26 (2): 243–63.
- Sarasvathy, Saras D. 2008. *Effectuation: Elements of Entrepreneurial Expertise*. Edward Elgar.
- Schumpeter, Joseph A. 1934. *The Theory of Economic Development: An Inquiry into Profits, Capital, Credit, Interest, and the Business Cycle*. Harvard University Press.
- Sein, Maung K., Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. 2011. "Action Design Research." *MIS Quarterly* 35 (1): 37–56.