Engineering Large Foundational Models for Enterprise Integration

Deliverable D4.1
LFM Ecosystem Documentation and ELFMo Methodology

Project title	Engineering Large Foundational Models for Enterprise Integration
Project acronym	ELFMo
Project number	23004
Work package	WP4
Deliverable	D4.1
Dissemination level	PU (public)
License	CC-BY 4.0
Version	1.0
Date	2026-02-06

Contributors

Editor(s)	Robin Bornoff (Siemens Industry Software NV)
Reviewer(s)	Patrícia Alves (ISEP), Tuuli Lindroos (FSC),
Contributor(s)	Mikko Raatikainen (University of Helsinki), Andrea Vianello (Siili), Marcos Cobo (CIC), Juhani Kivimäki (University of Helsinki)

Abstract

This deliverable defines the ELFMo Methodology, a comprehensive, lifecycle-based framework for the conception, development, operation, and retirement of Large Foundation Model (LFM)–enabled Generative AI (GenAI) products in industrial and enterprise contexts. Responding to the challenges of unclear governance, and emerging regulatory constraints, the methodology treats GenAI systems as evolving socio-technical products rather than isolated models or tools.

The document presents a structured lifecycle spanning development and operation phases, covering use case definition, KPI selection, architecture design, model selection and benchmarking, data preparation, adaptation, evaluation, assurance, deployment, monitoring, continuous improvement, and controlled retirement. Risk management, quality assurance, and regulatory compliance are embedded as cross-cutting concerns throughout the lifecycle.

In addition, the deliverable positions the methodology within the wider LFM ecosystem, outlining representative model classes, service providers, development and monitoring tools, and deployment options by way of example rather than prescription.

Table of contents

1 Introduction

1.1 Context

The rapid emergence of Large Foundation Models (LFMs) and Generative AI (GenAI) technologies is transforming the way digital products and services are conceived and deployed. While these technologies enable unprecedented flexibility and capability, their adoption in industrial and enterprise contexts has been characterised by fragmented experimentation, ad-hoc tooling, and limited lifecycle governance. A 'wild west' of digital product creation. In parallel, regulatory expectations around trustworthy, transparent, and accountable AI systems are increasing, particularly within the European context.

Within the ELFMo project, these developments motivate the need for a coherent, lifecycle-based methodology that enables organisations to move beyond isolated pilots toward sustainable, governed GenAI products. This deliverable responds to that need by positioning LFM-enabled GenAI systems as evolving socio-technical products embedded within organisational, technical, and regulatory environments.

1.2 Objectives

The primary objective of Deliverable D4.1 is to define and document the ELFMo Methodology, a comprehensive, lifecycle-based framework for the conception, development, operation, and retirement of LFM-enabled GenAI products.

Specifically, this deliverable aims to:

- Provide a structured lifecycle model that integrates technical, business, and governance considerations
- Enable risk-informed decision-making across all stages of GenAI product development and operation
- Embed quality assurance, regulatory compliance, and trustworthiness as cross-cutting lifecycle concerns
- Situate methodological guidance within the practical realities of the contemporary LFM ecosystem.

D4.1 does not introduce new algorithms or standalone tools. Its contribution lies in structuring and consolidating the results to date of WP2 and WP3 into a coherent, end-to-end process. The deliverable shows how the methods, techniques, metrics, and

governance mechanisms developed across these work packages can be aligned and orchestrated within a single lifecycle framework.

1.3 Target Audience

This deliverable is published at PU (public) level and is therefore intended for a broad audience:

- **Consortium partners**, especially technical partners in WP2 and WP3, including industrial solution providers, AI service providers, and tool developers.
- **External stakeholders**, including:
    - **Industry and commercial actors**, seeking reliable methods for LFM adaptation and evaluation.
    - **Research peers and the scientific community**, interested in advancing benchmarking practices (see Section 3.6 of this deliverable).
    - **Standardization bodies and policymakers**, interested in technical baselines for compliance with European regulation (EU AI Act, GDPR).

1.4 Related documents

Readers may consult the following documents for complementary context:

- **D1.1 Use cases description and requirements** - Defines the industrial use cases addressed by the ELFMo project and specifies their functional, technical, business, and regulatory requirements, together with initial KPIs that guide subsequent development and validation activities
- **D2.1 Research baseline for model training and benchmarking** – defines the technical foundations of WP2 and serves as the baseline for this deliverable.
- **D2.2 Initial release of benchmarking techniques** - Presents the first operational framework for selecting, adapting, training, and benchmarking Large Foundation Models for enterprise use, defining reproducible evaluation methodologies, performance metrics, and efficiency-oriented practices to support evidence-based model comparison.
- **D3.1 Research baseline for risk, quality and conformity assessment tools and procedures** – provides additional baselines relevant for WP2 and WP3 activities.

- **D3.2 Risk, quality and conformity assessment methods, risk indicators and quality metrics** - Describes the initial set of risk assessment methods, quality indicators, and conformity metrics developed in ELFMo, providing practical approaches for human-in-the-loop governance, KPI-based evaluation, trustworthiness assessment, and continuous monitoring across LFM-enabled systems.

- **FPP (Full Project Proposal)** – details the broader project rationale and innovation objectives

2 The ELFMo Methodology: Lifecycle for LFM-enabled GenAI Products

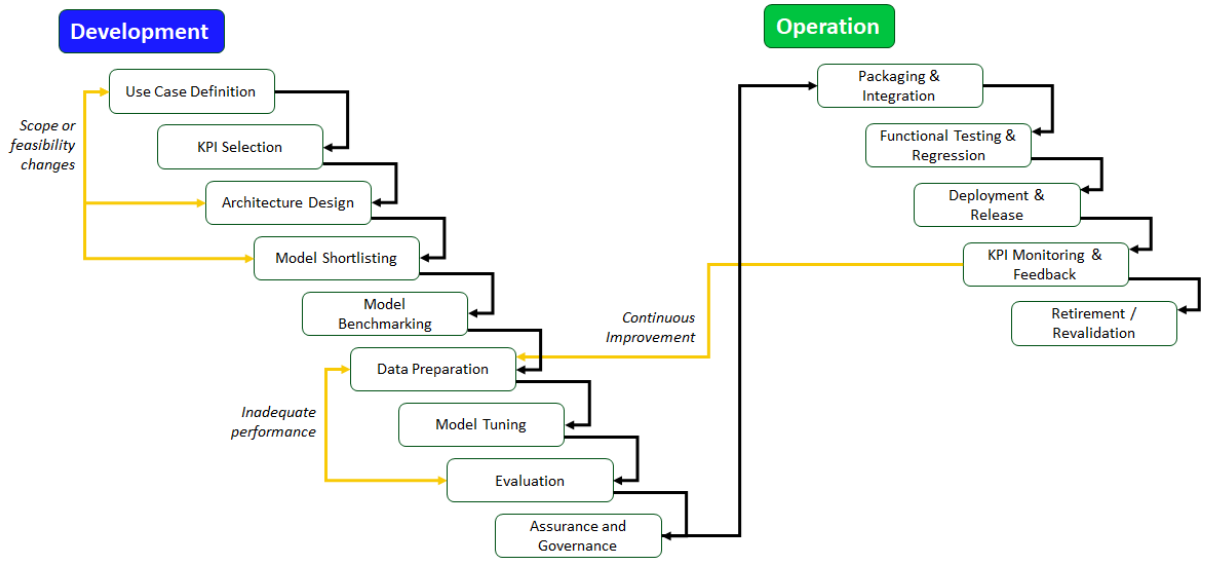

Figure 1: ELFMo GenAI Product Lifecycle

Figure 1 illustrates the end-to-end lifecycle defined by the ELFMo methodology, differentiating development and operation phases. It highlights iterative feedback loops driven by performance, feasibility, and scope changes, as well as continuous improvement during operation. While iterations are inherently possible, between any stages, the figure highlights the most important ones. The lifecycle integrates technical, business, and governance considerations and explicitly supports revalidation and controlled retirement of GenAI solutions.

The Development stages are further described in Section 2.2 and the Operational phases in Section 2.3.

2.1 Rationale for a Lifecycle-based Methodology

The rapid emergence of LFMs has fundamentally altered the landscape of AI-enabled product development. Unlike traditional machine learning systems, LFMs exhibit a high degree of generality and adaptability, enabling powerful new applications. However, they also exhibit non-deterministic behaviour and are opaque thus amplify technical, organisational, and regulatory risks. In industrial practice, this has led to a proliferation of isolated proofs-of-concept, ad-hoc integrations, and experimental deployments that often fail to transition into sustainable, trustworthy products.

The ELFMo project addresses this gap by advocating a lifecycle-based methodology that treats LFM-enabled GenAI systems not as isolated models or tools, but as evolving socio-technical products embedded in organisational, legal, and market contexts. A lifecycle perspective ensures that decisions made during early experimentation, such as model selection, data usage, or architectural patterns are systematically linked to long-term operational, compliance, and business implications.

Although the ELFMo lifecycle is presented as a sequence of stages for clarity (Figure 1), it is not intended to prescribe a linear or waterfall development process. In practice, LFM-enabled GenAI products evolve through frequent iteration, partial rework, and parallel activities. The methodology is deliberately compatible with agile and incremental development approaches, while retaining explicit lifecycle stages to support governance, risk management, and decision-making in industrial contexts. Feedback loops driven by performance, feasibility, scope change, and operational monitoring enable continuous adaptation without sacrificing traceability or assurance.

2.1.1 Limitations of Ad-hoc and Tool-centric GenAI Adoption

Current industrial adoption of GenAI is frequently driven by tool availability rather than methodological discipline. Organisations experiment with publicly available APIs or open-source models without a structured process for evaluating suitability, risks, or long-term maintainability. While such experimentation can generate short-term insights, it often results in fragmented architectures or craft-like developments (rather than systematic workflows and pipelines) that are difficult to scale or govern, unclear ownership of model behaviour and failure modes, late discovery of regulatory or data protection constraints, and an inability to systematically compare alternative solutions.

A tool-centric approach also tends to conflate model performance with product success, overlooking system-level properties such as robustness, explainability, operational cost, and user trust. The ELFMo methodology explicitly counters this tendency by embedding

GenAI capabilities within a broader product lifecycle that integrates technical, organisational, and regulatory considerations from the outset.

2.1.2 Relationship to Risk-based Engineering and MLOps

The proposed lifecycle draws on established practices from risk-based engineering, software product lifecycle management, and Machine Learning Operations (MLOps). However, LFMs introduce distinctive challenges that extend beyond classical MLOps assumptions. These include limited model transparency, dependency on external model providers, evolving behaviour under prompt or data changes, non-deterministic (non-repeatable) behaviour and heightened regulatory scrutiny.

In the ELFMo methodology, risk management is not treated as a separate activity but as a cross-cutting concern that informs decisions at every lifecycle stage. Early phases focus on feasibility and risk identification, development phases emphasise controlled adaptation and evaluation, and operational phases prioritise monitoring, governance, and continuous validation.

2.1.3 Positioning within the ELFMo Innovation Objectives

The lifecycle methodology defined in this deliverable provides the structural framework through which the ELFMo innovation objectives are realised. Risk-based decision making (Innovation 1) is operationalised through explicit lifecycle gates, KPIs, and evaluation criteria. Trustworthy adaptation and integration of LFMs (Innovation 2) is supported by structured development and assurance stages. Evidence-based quality and compliance assessment (Innovation 3) is embedded across both development and operation phases. Finally, fostering open and sovereign AI ecosystems (Innovation 4) is enabled through informed model selection, benchmarking, and lifecycle governance.

By situating individual tools, techniques, and use cases within this unified lifecycle, the ELFMo methodology aims to transform fragmented GenAI experimentation into repeatable, auditable, and sustainable industrial practice.

2.2 Development Phase of the ELFMo Lifecycle

2.2.1 Use Case Definition

In line with established software and systems engineering practice, the development phase of the ELFMo lifecycle begins with a rigorous and explicit definition of the target use case. Any structured software development process is fundamentally grounded in a

clear understanding of what problem is to be solved and why. In the context of ELFMo, this stage establishes the strategic intent, operational context, and feasibility boundaries of the proposed LFM-enabled GenAI product, and functions as the primary decision gate for determining whether the use of a Large Foundation Model is justified.

In contrast to exploratory prototyping, use case definition within the ELFMo methodology is a structured engineering activity. It requires the articulation of the problem space in terms that are meaningful both to business stakeholders and to technical and governance actors. This includes identifying the decisions or processes to be supported, the role of human users, and the consequences of erroneous or misleading system behaviour.

A critical aspect of this stage is recognising that GenAI and LFMs are not universally appropriate solutions. The methodology therefore explicitly supports negative feasibility decisions, allowing organisations to reject or defer GenAI adoption when simpler, more deterministic, or lower-risk approaches are sufficient. This avoids unnecessary technical complexity and mitigates downstream governance and compliance risks.

Key outcomes of this stage include a clearly scoped problem statement, an initial risk profile, and explicit success criteria that can be traced throughout the lifecycle.

This stage is directly grounded in the risk assessment and decision-support baselines established in WP3. In particular, D3.1 defines structured methods for early-phase risk identification and feasibility analysis across multiple domains, including consumer cybersecurity, telemarketing, and built-environment consultancy. These methods support the systematic evaluation of whether LFM adoption is justified by analysing privacy exposure, data quality dependencies, security vulnerabilities, and explainability requirements.

Furthermore, D3.1 introduces threat modelling as a core mechanism for anticipating misuse, attack vectors, and unintended behaviours at an early design stage. Integrating threat modelling into use case definition ensures that security and trustworthiness are treated as first-class design drivers rather than downstream concerns.

Complementing this, D3.2 provides a structured framework for identifying trustworthiness factors through the Trustworthiness Canvas and associated Risk and Guardrail Cards. These instruments support workshops and co-creation sessions that help stakeholders explicitly surface ethical, legal, and organisational trust requirements at the moment the use case is defined.

Together, D3.1 and D3.2 ensure that use case definition in the ELFMo methodology is not only business-driven but also risk-aware and governance-oriented from its inception.

2.2.2 KPI Selection

Once a use case has been defined, its objectives must be translated into measurable indicators. KPI selection provides the quantitative backbone for evidence-based decision making across the entire ELFMo lifecycle.

For LFM-enabled GenAI products, KPIs extend beyond classical accuracy metrics. They must capture multiple dimensions, including model behaviour, system performance, business impact, and trustworthiness. Typical KPI categories include model-level performance indicators (e.g. relevance, hallucination rate, robustness), system-level indicators (latency, availability, scalability, cost), business-level indicators (productivity gains, revenue impact, risk reduction), and governance-related indicators (bias metrics, explainability coverage, audit readiness).

The methodology emphasises KPI traceability: each KPI should be explicitly linked to a use case objective and revisited as the system evolves. This ensures that optimisation efforts remain aligned with business value rather than drifting toward narrow technical improvements.

In high-volume industrial contexts, such as Customer Experience (CX) operations, it is crucial to further distinguish between *operational* metrics and *strategic* value. For example, while a technical metric like "token generation latency" directly influences the "Average Handling Time" (AHT) - a critical operational KPI - optimization efforts must be balanced against quality indicators like "First Contact Resolution" (FCR) and "Customer Satisfaction" (CSAT).

A robust KPI framework should therefore pair efficiency targets with quality guardrails. For example, a successful Agent Copilot deployment might target a 12-18% reduction in AHT, but this success is conditional on maintaining or improving the Net Promoter Score (NPS). Furthermore, measurements should account for the "cognitive load" of the human operator; an effective LFM solution should not merely speed up the process but actively reduce the stress and complexity of information retrieval for the user, which can be measured through agent satisfaction surveys and turnover rates.

The selection of KPIs is closely aligned with the measurement frameworks developed in WP3. D3.2 provides concrete methodologies for defining, measuring, and operationalising business KPIs, internal performance KPIs, and Service Level Objectives

(SLOs). These frameworks demonstrate how technical performance indicators can be linked to business value, service quality, and compliance objectives.

While many of the KPIs in D3.2 originate from business performance scenarios such as telemarketing, the underlying methodology is directly transferable to trust-, risk-, and compliance-related KPIs. In particular, D3.2 introduces systematic approaches for translating abstract trustworthiness goals into measurable indicators, including KRIs (Key Risk Indicators) and KCIs (Key Control Indicators), which are essential for lifecycle governance of LFM-enabled systems.

In addition, D3.1 defines how business KPIs and technical monitoring metrics can be combined within integrated decision-support frameworks. This supports the ELFMo objective of aligning KPI selection not only with model performance, but with organisational risk tolerance and regulatory constraints.

2.2.3 Architecture Design

Architecture design translates conceptual requirements into a concrete system structure capable of supporting the intended GenAI functionality. For LFM-enabled systems, architectural decisions are particularly consequential, as they determine not only performance and cost but also risk exposure, observability, and regulatory compliance.

This stage addresses system topology (cloud, on-premise, hybrid, or edge), interaction patterns such as Retrieval-Augmented Generation or agentic architectures, data flow and trust boundaries, and integration with enterprise systems. Unlike traditional software, GenAI architectures must explicitly accommodate uncertainty in model behaviour and evolving dependencies on external model providers or data sources.

The ELFMo methodology therefore promotes architectures that support isolation of failure modes, monitoring by design, and controlled evolution. Architectural decisions are documented and justified in relation to both technical requirements and governance constraints.

This Architecture Design phase translates high-level functional, business, and governance requirements into a modular and evolvable system architecture that supports both controlled experimentation and stable production operation. Architecture design is therefore not only a technical structuring activity, but the point where trust, risk, and compliance considerations are concretely embedded into the system. It makes the

system's trust boundaries explicit, including inputs, retrieval sources, tools and actions, and outputs, and treats them as first-class design elements.

Mitigation hypotheses derived from the risk analysis should be implemented directly in the architecture through concrete safeguards, with clearly assigned ownership and measurable indicators to evaluate their effectiveness continuously. In this way, architecture design operationalises risk management: safeguards become architectural components rather than external procedures. This ensures that architectural decisions are grounded not only in performance and cost considerations, but also in observability by design, auditability, and readiness for regulatory compliance across the full system lifecycle.

From a governance perspective, this architectural grounding is reinforced by WP3. In D3.2, the Trustworthiness framework introduces the Guardrail Card as a mechanism to translate abstract risk mitigation ideas into operational controls. These guardrails define explicit responsibilities, success criteria, and review cycles, and are linked to measurable Key Control Indicators (KCIs). By embedding these guardrails directly into the architecture, governance is not treated as an external layer, but becomes an intrinsic property of the system design.

In addition, D3.1 highlights that architecture design is one of the primary instruments for enforcing security, privacy, and compliance by design. Architectural choices around deployment topology (cloud, on-premise, hybrid), data flow separation, and dependency management directly shape the system's exposure to risks such as data leakage, adversarial manipulation, and regulatory non-compliance. Integrating these considerations at design time ensures that assurance and governance objectives are implemented proactively rather than retrofitted after deployment.

Together, these contributions position Architecture Design in the ELFMo methodology as the convergence point between engineering, risk management, and governance: the stage where system structure, performance, scalability, and cost efficiency are explicitly balanced with human oversight, regulatory readiness, and the long-term trustworthiness of LFM-enabled GenAI systems.

2.2.4 Model Shortlisting

Model shortlisting reduces the broad landscape of available foundation models to a manageable set of candidates suitable for the defined use case and architecture. This process considers not only functional capability but also long-term sustainability and dependency risks.

Selection criteria include modality support, adaptability, resource requirements, licensing terms, intellectual property constraints, vendor lock-in risks, and ecosystem maturity. In regulated or strategically sensitive domains, model sovereignty and deployment control are treated as first-class concerns.

The outcome of this stage is a documented shortlist of candidate models, together with explicit rationale for inclusion and exclusion.

Model shortlisting should be executed as an evidence-informed filtering step that narrows the landscape of candidate LFMs based on both capability fit and deployment feasibility. WP2 provides the technical foundations for this: D2.2 defines a structured selection methodology and criteria for exploring, filtering, and ranking LFMs for enterprise integration, including model characteristics, ecosystem maturity, efficiency, and constraints relevant to industrial adoption.

In addition, D2.1 provides supporting baseline considerations for enterprise integration - emphasising that shortlist criteria must reflect not only raw capability, but also the practicalities of secure integration, maintainability, and operational constraints across cloud/on-prem/hybrid environments.

Finally, WP3 inputs should act as selection constraints rather than after-the-fact checks: D3.2's trustworthiness framing helps translate early trust requirements (e.g., privacy expectations, security posture, accountability needs) into explicit "must-have" or "exclusion" criteria that influence which candidate models can responsibly proceed to benchmarking.

2.2.5 Model Benchmarking

Benchmarking provides an evidence-based basis for selecting between shortlisted models. Generic public benchmarks are insufficient for most industrial use cases; evaluation must instead reflect domain-specific tasks, data characteristics, and risk profiles.

Benchmarking activities may include task-specific performance evaluation, stress testing for hallucination and bias, robustness assessment, and cost-performance analysis under realistic constraints. Results are used to inform a transparent and auditable model selection decision.

Benchmarking in ELFMo should be treated as the evidence gate between a plausible shortlist and a defensible model selection decision. WP2 provides the core benchmarking methods: D2.2 defines reproducible benchmarking approaches that

combine classical NLP metrics, semantic similarity measures, and LLM-as-a-judge strategies, enabling transparent comparison across models and configurations.

D2.1 complements this by framing benchmarking and validation as part of a broader adaptation lifecycle, linking model evaluation to both technical metrics and business KPIs, and emphasising that evaluation must account for deployment trade-offs (quality vs. latency vs. cost).

WP3 strengthens benchmarking by ensuring it covers risk and conformity dimensions. D3.2 provides methods, indicators, and quality metrics oriented toward trustworthy enterprise deployment, and D3.1 establishes baseline procedures for risk- and compliance-aware assessment - together ensuring that benchmarking explicitly tests for failure modes (e.g., hallucination sensitivity, robustness, misuse risk) rather than only average-case performance.

Finally, WP1 ensures benchmarking remains grounded in real value: D1.1 defines use-case requirements and KPIs that can be operationalised into domain-relevant evaluation scenarios and acceptance thresholds, avoiding over-reliance on generic public benchmarks.

2.2.6 Data Preparation

Data preparation is a central cost and risk driver in GenAI product development. Data may be used for fine-tuning, retrieval, evaluation, or monitoring, each with distinct quality and governance requirements.

Key concerns include data quality, bias, provenance, privacy, and traceability. The ELFMo methodology treats data as a lifecycle asset, requiring continuous governance rather than one-off preparation. Links between datasets, model versions, and outputs are explicitly maintained to support auditability and continuous improvement.

Data preparation should be treated as a cross cutting, continuous discipline across the system lifecycle, not as a one-off preprocessing task. A process and tooling driven approach is required to enable systematic curation, automation, and reproducibility of datasets. This includes mechanisms to generate derived datasets, such as synthetic data or privacy preserving variants, which are especially relevant when real world data is scarce or highly sensitive.

This phase should make data governance an integral part of the system: provenance, versioning, access control, and traceability links between datasets, model versions, and

evaluation outcomes should be managed as first class artifacts. This is essential to ensure auditability and to enable continuous improvement.

In addition, D2.1 provides the baseline rationale for treating data preparation as a core enabler of trustworthy adaptation, highlighting automation of cleaning/normalisation workflows and the need to maintain traceability and compliance links between data, models, and outcomes across the lifecycle.

WP3 further constrains this stage by emphasising privacy, governance, and regulatory compliance as first-class requirements for any dataset used for fine-tuning, retrieval, evaluation, or monitoring - particularly where sensitive or proprietary enterprise data is involved.

2.2.7 Model Tuning and Adaptation

Model tuning adapts the selected foundation model to the specific use case. Techniques range from prompt engineering and retrieval augmentation to parameter-efficient fine-tuning.

Model tuning and adaptation should follow a graduated strategy, escalating from minimally invasive techniques (prompting, RAG) toward more invasive approaches (parameter-efficient fine-tuning) only where measurable benefit justifies added cost and risk. WP2 provides the methodological and technical basis for this: D2.1 outlines adaptation strategies (including fine-tuning and continuous learning) and frames them explicitly in terms of performance, safety, and compliance constraints in enterprise settings.

D2.2 complements this by describing practical approaches for efficient fine-tuning (e.g., LoRA/QLoRA and related parameter-efficient methods) and by situating tuning within a benchmark-driven workflow so that each adaptation step is evaluated and traceable rather than ad hoc.

From a trustworthiness perspective, WP3 ensures that adaptation is paired with explicit mitigation planning. D3.1 and D3.2 provide risk-oriented procedures and trustworthiness tools (risk/guardrail framing) that help ensure adaptation does not increase unacceptable risk exposure (e.g., privacy leakage, bias amplification, unsafe behaviours), and that necessary controls are defined early enough to be integrated into both the tuned system and its surrounding workflow.

2.2.8 Evaluation

Evaluation assesses whether the adapted system satisfies functional, non-functional, and governance requirements. This includes automated regression testing, system-level validation, and human-in-the-loop assessment where appropriate.

Evaluation results inform the decision to proceed to operational deployment and establish baseline metrics for ongoing monitoring.

Evaluation activities can also include verification that identified risks remain within acceptable bounds (KRIs) and that defined guardrails are effective (KCIs).

WP2 provides the evaluation backbone needed for evidence-based decisions. D2.2 defines benchmarking techniques and evaluation frameworks (including combinations of classical NLP metrics, semantic similarity measures, and LLM-as-a-judge approaches) suitable for comparing models and system variants under realistic constraints.

D2.1 further positions evaluation as part of a validation workflow that must reflect enterprise deployment trade-offs and lifecycle needs (e.g., quality vs. cost vs. latency, and robustness under changing prompts/data/dependencies), ensuring evaluation produces decision-grade evidence rather than one-off test results.

Evaluation should remain anchored to WP1's use-case requirements and KPIs: acceptance thresholds and "definition of done" criteria should be traceable back to the operational goals and constraints documented in D1.1, so that evaluation outcomes map directly to business value, user impact, and deployment readiness.

2.2.9 Assurance and Governance

Assurance and governance activities establish confidence that the system complies with ethical, legal, and organisational requirements. These activities include bias assessment, explainability measures, misuse prevention, documentation, and audit preparation.

Rather than a final checklist, assurance is treated as a continuous process that extends into the operational phase.

This lifecycle stage is directly supported by the governance frameworks introduced in D3.1 and D3.2. D3.1 establishes AI governance as a continuous organisational process, encompassing responsibility allocation, transparency requirements, documentation

practices, and regulatory alignment. It positions governance not as a final compliance check but as an ongoing management activity that evolves with the system.

D3.2 operationalises this governance perspective through its Governance Policy Card, which consolidates multiple guardrails into organisation-wide policies with defined ownership, KPIs, and review cycles. This provides a concrete mechanism for scaling assurance practices beyond individual projects and embedding them into enterprise governance structures.

The combination of KRIs and KCIs, introduced in D3.2, ensures that assurance is measurable, auditable, and continuously verifiable. This directly aligns with the ELFMo methodology's view of assurance as an active lifecycle discipline rather than a static certification step.

2.3 Operation Phase of the ELFMo Lifecycle

The operation phase of the ELFMo lifecycle addresses the sustained, trustworthy, and economically viable use of LFM-enabled GenAI products in real-world environments. While development focuses on feasibility and controlled validation, operation confronts continuously evolving conditions: changing user behaviour, drifting data distributions, evolving regulatory expectations, and rapid advances in foundation model technology. As a result, this phase is critical for ensuring that GenAI solutions remain reliable, compliant, and aligned with business objectives over time.

In the ELFMo methodology, operation is not treated as a passive post-deployment state, but as an active phase characterised by continuous observation, decision making, and controlled evolution. Responsibilities during this phase span technical operations, governance, business ownership, and human oversight.

2.3.1 Packaging and Integration

Packaging and integration mark the transition from a validated development artefact to an operational product component. This stage focuses on productisation rather than experimentation, ensuring that GenAI capabilities can be reliably consumed within enterprise environments.

Key concerns at this stage include the definition of stable APIs and service contracts, versioning strategies for models and prompts, and integration with existing enterprise systems and workflows. Unlike conventional software components, GenAI services

often encapsulate probabilistic behaviour, which must be explicitly communicated to downstream consumers through interface contracts and documentation.

Integration activities must also address security, access control, and data handling boundaries. In many enterprise contexts, GenAI systems operate across organisational or trust boundaries, making it essential to clearly define responsibility for inputs, outputs, and decision outcomes. To manage these complexities effectively, integration should ideally be mediated through a dedicated orchestration layer rather than direct point-to-point connections. In practice, this often takes the form of a "low-code" or "no-code" process orchestrator that wraps the non-deterministic LFM components within deterministic business logic.

This architectural pattern allows for the standardization of connectors to external systems (such as CRMs, ERPs, or ticketing platforms) while maintaining a centralized control plane for data flow. By decoupling the LFM inference engine from the core business systems via an event-driven bus, organizations can ensure that the stochastic nature of Generative AI does not compromise the data integrity of transactional systems. Furthermore, this orchestration approach facilitates the implementation of "swappable" model backends, allowing the underlying LFM to be updated or replaced without disrupting the broader enterprise integration. The ELFMo methodology therefore emphasises explicit interface definitions and documentation as enablers of both operational robustness and auditability.

2.3.2 Functional Testing and Regression

Once integrated, GenAI products must be protected against unintended behavioural drift. Unlike traditional software, LFM-enabled systems may change behaviour as a result of updates to models, prompts, retrieval data, or even external dependencies such as hosted model APIs.

Functional testing in the operation phase therefore extends beyond conventional test cases. It includes validation of representative interaction scenarios, monitoring of output distributions, and verification that previously accepted behaviour remains within defined tolerances. Regression testing plays a critical role in detecting subtle degradations such as increased hallucination rates, loss of relevance, or changes in tone or intent.

The ELFMo methodology promotes risk-based testing strategies, where the depth and frequency of testing are proportional to the potential impact of failure. High-impact or

safety-relevant use cases require more stringent regression controls and human oversight than low-risk informational applications.

2.3.3 Deployment and Release Strategies

Deployment and release strategies determine how GenAI products are introduced into live environments and how updates are propagated over time. Given the uncertainty inherent in probabilistic models, uncontrolled deployment can expose organisations to significant operational and reputational risk.

The methodology therefore advocates controlled deployment mechanisms such as staged rollouts, canary releases, shadow deployments, or sandboxed environments. These approaches allow organisations to observe real-world behaviour under limited exposure before committing to full-scale release. Release decisions are informed not only by technical readiness but also by governance approvals and risk assessments.

Deployment strategies must also account for different operational contexts, including cloud-based services, on-premise installations, and hybrid or edge deployments. Each context introduces distinct constraints related to latency, data locality, security, and compliance.

2.3.4 KPI Monitoring and Feedback Loops

Continuous monitoring is central to the operational integrity of LFM-enabled GenAI products. Once deployed, systems are exposed to dynamic environments in which both data and usage patterns evolve over time. Without systematic monitoring, performance degradation or emerging risks may remain undetected until they cause material harm.

In the ELFMo methodology, monitoring encompasses multiple layers. Model-level monitoring tracks behavioural indicators such as relevance, hallucination frequency, bias signals, and confidence measures. System-level monitoring addresses latency, availability, error rates, and cost efficiency. Business-level monitoring evaluates whether the system continues to deliver expected value in terms of productivity, revenue, or risk reduction.

Feedback loops connect monitoring results back to decision-making processes. Alerts, dashboards, and review cycles enable timely interventions such as retraining, configuration changes, or temporary rollback. Human-in-the-loop mechanisms remain essential, particularly for high-impact decisions or ambiguous situations. Human-in-the-Loop must be defined as a specific protocol. Effective monitoring requires the definition

of explicit escalation triggers - scenarios where the system is mandatorily forced to defer to human judgment. These triggers should include technical signals, such as low model confidence scores, as well as semantic signals, such as the detection of sensitive intents (e.g., "intention to cancel" or "complaint to official body") or signs of user frustration (sentiment analysis).

In these scenarios, the system should degrade gracefully from an "Autopilot" mode to a "Copilot" mode, presenting the human supervisor with the context and a suggested draft rather than taking autonomous action. This feedback loop serves a dual purpose: it mitigates immediate operational risk and generates high-quality, annotated data ("golden datasets") that can be used to retrain the model and reduce the frequency of future escalations.

The monitoring phase builds on the integrated business and model monitoring frameworks defined in D3.1. Section 3 of D3.1 describes how data collection, analysis, alerting, and corrective actions can be combined into a unified monitoring pipeline that covers both technical system behaviour and business-level performance indicators.

In addition, D3.2 extends monitoring into the governance domain by introducing continuous validation of trustworthiness through KRIs and KCIs. These indicators allow organisations to verify that identified risks remain within acceptable thresholds and that defined guardrails remain effective in real operation.

By linking KPI monitoring to both operational performance and governance controls, the ELFMo methodology ensures that feedback loops support not only optimisation and improvement, but also sustained regulatory compliance, risk containment, and trust preservation.

2.3.5 Continuous Improvement

Continuous improvement transforms operational evidence into actionable change. Rather than treating deployment as the end of development, the ELFMo methodology positions operation as an iterative learning phase.

Improvement activities may include incremental model updates, refinement of prompts or retrieval strategies, expansion of training or evaluation datasets, and architectural adjustments. Importantly, changes are prioritised based on monitored KPIs and risk assessments, ensuring that improvement efforts remain aligned with business objectives and governance constraints.

The methodology also recognises the rapid evolution of the foundation model ecosystem. Periodic re-benchmarking against emerging models or techniques enables organisations to make informed decisions about upgrading, migrating, or consolidating GenAI capabilities.

2.3.6 Retirement, Revalidation, and Decommissioning

The final stage of the operation phase addresses the controlled retirement or revalidation of GenAI products or components. Unlike traditional software, GenAI systems may become unsuitable not only due to technical obsolescence but also due to regulatory change, shifts in acceptable risk, or the availability of superior models.

Retirement decisions are triggered by factors such as sustained KPI degradation, unacceptable risk exposure, changes in scope or requirements, or loss of compliance with evolving regulatory frameworks. In some cases, revalidation may be sufficient, involving renewed evaluation and assurance activities. In others, full decommissioning is required.

The ELFMo methodology emphasises planned decommissioning to preserve audit trails, documentation, and organisational knowledge. Controlled retirement ensures that GenAI components do not silently persist beyond their intended lifecycle and that successor systems can be introduced without unmanaged risk.

2.4 The LFM Ecosystem: Models, Providers, and Tooling Landscape

The ELFMo methodology is situated within a rapidly evolving and heterogeneous LFM ecosystem. Effective lifecycle-based decision-making therefore requires not only methodological guidance, but also a concrete understanding of the surrounding ecosystem of models, service providers, development tools, and deployment options. This section provides a consolidated, non-exhaustive overview of representative ecosystem elements, named *by way of example rather than prescription*, to support informed and pragmatic decision-making across lifecycle stages.

Rather than attempting an exhaustive or static catalogue, the intent is to illustrate *typical classes of tools and services* currently used in industrial practice, together with the trade-offs they introduce in terms of performance, cost, risk, and regulatory compliance. Detailed benchmarking, risk assessment, and evaluation techniques referenced here are developed further in WP2 and WP3 deliverables, notably D2.2, D3.1, and D3.2.

2.4.1 LFM Model Landscape

The contemporary LFM model landscape spans a spectrum from large proprietary, cloud-hosted models to open and open-weight models that can be self-hosted or adapted within enterprise environments.

Representative proprietary model families include models offered through managed APIs by major providers (e.g. large general-purpose conversational and multimodal models). These models typically provide strong out-of-the-box performance, rapid access to state-of-the-art capabilities, and managed scalability. However, they introduce dependencies related to vendor lock-in, limited transparency into training data and internal model structure, pricing volatility, and potential data sovereignty concerns.

In contrast, open and open-weight models - for example those distributed via public repositories or research-led initiatives - enable deeper inspection, fine-tuning, and controlled deployment. Such models are particularly relevant for organisations requiring on-premise deployment, stricter data governance, or greater control over lifecycle evolution. These benefits are offset by increased responsibility for benchmarking, optimisation, and operational management.

Within the ELFMo lifecycle, model selection is therefore treated as a strategic decision rather than a purely technical one, informed by use-case criticality, regulatory exposure, operational constraints, and long-term sustainability. WP2 contributes systematic benchmarking techniques for comparing model capabilities, while WP3 provides complementary risk and compliance evaluation criteria.

2.4.2 Service Providers and Hosting Options

Beyond the models themselves, the LFM ecosystem includes a diverse range of service providers offering hosted inference, managed fine-tuning, orchestration, and monitoring capabilities. Examples include hyperscale cloud platforms providing integrated AI services, specialised AI platform providers offering model hubs and lifecycle management, and system integrators delivering bespoke GenAI solutions.

Managed services can significantly reduce time-to-market, particularly during early development and pilot phases. They often provide built-in scalability, security features, and integration with existing enterprise tooling. At the same time, reliance on external service providers raises considerations related to data transfer, jurisdictional compliance, service availability, and long-term cost predictability.

The ELFMo methodology therefore encourages explicit evaluation of hosting and service models as part of architectural design and assurance activities. Hybrid approaches - combining externally hosted inference with internally managed data pipelines or governance layers - are commonly viable and allow organisations to balance agility with control. Such decisions are revisited during operation as usage patterns, regulatory expectations, and cost structures evolve.

2.4.3 Development, Orchestration, and Integration Tooling

Across the LFM lifecycle, a growing ecosystem of software tools supports development, adaptation, orchestration, and integration. Typical examples include:

- **Prompt and workflow orchestration frameworks**, which enable structured composition of prompts, tools, and model calls;
- **Retrieval-Augmented Generation (RAG) toolkits**, integrating vector databases and document pipelines;
- **Model experimentation and evaluation environments**, supporting rapid comparison of prompts, models, and configurations;
- **Data preparation and annotation tools**, facilitating dataset curation and quality control.

In industrial practice, such tools are frequently combined rather than used in isolation. From an ELFMo perspective, tooling choices are guided by their ability to support traceability, reproducibility, and lifecycle integration, rather than by raw feature richness. WP2 provides concrete guidance on toolchains supporting benchmarking and adaptation, while WP3 focuses on tooling that enables evidence collection for risk and conformity assessment.

Crucially, given the pace of ecosystem change, the methodology avoids hard dependencies on specific products. Instead, it emphasises modularity, open interfaces, and the ability to substitute tooling components as requirements evolve.

2.4.4 Monitoring, Governance, and Assurance Tooling

Operational GenAI systems require continuous visibility into model behaviour, system performance, and emerging risks. A parallel ecosystem of monitoring and governance tools has therefore emerged, addressing aspects such as output evaluation, drift detection, policy enforcement, and audit support.

Representative examples include:

- **Model and output monitoring platforms** tracking quality, bias indicators, and behavioural drift
- **Observability tools** capturing latency, throughput, and error characteristics
- **Governance artefacts and tooling**, such as model cards, system cards, and audit logs
- **Human-in-the-loop review systems** enabling escalation and oversight for high-impact decisions.

These tools are particularly relevant during the operation phase (Section 2.3), where continuous assurance replaces one-off validation. WP3 deliverables elaborate methods and indicators for risk, quality, and conformity assessment that can be instantiated using such tooling.

2.4.5 Deployment and Runtime Ecosystem

The deployment ecosystem for LFM-enabled GenAI products spans cloud, on-premise, edge, and hybrid environments. Runtime considerations include latency constraints, hardware availability (e.g. GPU, accelerator access), energy efficiency, and security boundaries.

Examples of runtime components include lightweight inference engines, containerisation platforms, workload routing mechanisms, and hardware abstraction layers. Increasingly, enterprises deploy mixed strategies in which different models or configurations are selected dynamically based on workload characteristics or risk profile.

Within the ELFMo lifecycle, deployment is treated as a revisitable decision rather than a terminal state. Deployment choices directly influence the feasibility of monitoring, governance, and compliance activities, and are therefore tightly coupled with operational assurance.

2.4.6 Implications for Lifecycle-based Decision Making

The diversity and rapid evolution of the LFM ecosystem reinforce the central premise of the ELFMo methodology: sustainable GenAI adoption requires structured, lifecycle-wide

decision-making. No single model, provider, or tool is universally optimal, and premature standardisation on specific technologies can introduce long-term risk.

By situating lifecycle activities within a concrete but non-prescriptive view of the LFM ecosystem, this section complements the methodological guidance in Sections 2.2 and 2.3. Together, these elements provide a foundation for informed technical, organisational, and strategic decisions, supporting the development of GenAI products that are not only performant, but also trustworthy, compliant, and economically viable.

3 Conclusions

This deliverable has introduced the ELFMo Methodology, a structured, lifecycle-based approach for the conception, development, operation, and retirement of Large Foundation Model (LFM)–enabled Generative AI products in industrial and enterprise contexts. The methodology responds to the growing need for systematic, risk-informed, and governable approaches to GenAI adoption, moving beyond ad-hoc experimentation toward sustainable productisation.

D4.1 establishes the conceptual and methodological foundations of the ELFMo approach. It defines a coherent lifecycle spanning development and operation, embeds risk management, quality assurance, and regulatory considerations as cross-cutting concerns, and situates lifecycle decisions within the realities of the contemporary LFM ecosystem. By doing so, it provides a common reference framework for technical, organisational, and governance stakeholders across the consortium.

The document deliberately focuses on structure, principles, and decision logic rather than exhaustive prescriptions or tool-specific guidance. Representative models, services, and tools are discussed by way of example to illustrate typical ecosystem patterns, while avoiding premature standardisation in a rapidly evolving technological landscape. This positions the methodology to remain robust as models, platforms, and regulatory expectations continue to evolve.

The ELFMo Methodology is intended to be refined, validated, and operationalised in subsequent project stages. In particular, Deliverable D4.3 will build on this foundation by incorporating concrete lessons learned from project use cases, deeper integration with WP2 tooling and benchmarking results, and tighter alignment with WP3 risk, quality, and conformity assessment methods. Together, these future contributions will transform the methodological framework defined here into a fully instantiated and empirically validated approach for trustworthy GenAI adoption.

In this sense, D4.1 should be understood not as a final specification, but as a reference baseline that enables consistent dialogue, informed decision-making, and structured evolution across the ELFMo project and beyond.