# Dexon BPM Methodology

## Executable Business Architecture

## Implementation methodology on Dexon BPM · Version 8

August 2026

## Contents

Chapter 1. Purpose, Nature, and Scope of the Methodology ..... 7

**1.1 Purpose of the methodology ..... 7**

**1.2 Methodological scope ..... 9**

**1.3 Nature of the methodology ..... 12**

**1.4 Theoretical foundations of the method ..... 12**

**1.5 Scope of application and assumptions ..... 14**

**1.6 Limits of applicability ..... 15**

**1.7 Light Version of the Method ..... 16**

Chapter 2. Guiding principles ..... 17

**2.1 Architecture before automation ..... 17**

**2.2 Traceability as a condition of rigor ..... 18**

**2.3 The goal of the process as a design unit ..... 18**

**2.4 BPMN as a formal language, not as a starting point ..... 18**

**2.5 Data as a condition for execution, decision and measurement ..... 19**

**2.6 Measurement by design, not at the end ..... 19**

Chapter 3. Business Architecture Fundamentals ..... 19

**3.1 Business architecture as a governable representation of the organization ..... 20**

**3.2 Business model, value proposition and service portfolio ..... 20**

**3.3 Deliberate strategy, emergent strategy and learning ..... 21**

**3.4 Capabilities, Processes, and Services ..... 21**

**3.5 Processes as coordinated implementation of capabilities ..... 22**

**3.6 Strategic Capability Foundation: Resources, Dynamic Capabilities and Operating Model .. 22**


All Rights Reserved – Dexon Software-2026

**3.7 The company in production and the company that transforms it ..... 23**
Chapter 4. The Dexon business traceability chain ..... 24
**4.1 Definition ..... 24**
**4.2 Methodological reading of each link ..... 24**
On the order of the chain and other decompositions ..... 25
**4.3 The metamodel of the executable architecture ..... 25**
Structural objects in the chain ..... 27
Target Specification Objects ..... 28
Derived concepts and roles ..... 29
**4.4 Relationships and cardinalities ..... 30**
Chain Relationships ..... 30
Design contract relationships ..... 32
Transversal relationships of responsibility ..... 33
**4.5 Process categories and traceability required ..... 33**
**4.6 Architectural integrity restrictions ..... 34**
How each constraint is verified ..... 36
Relationship with the notion of reference architecture ..... 38
The scope of the word "formal" ..... 38
Chapter 5. Introduction to Objective-Driven Process Design ..... 39
**5.1 Definition ..... 39**
**5.2 Difference from traditional TO-BE ..... 39**
**5.3 Relationship to BPMN ..... 40**
**5.4 Formulation of the process objective ..... 40**
**5.5 TO-BE approval criteria ..... 40**
Chapter 6. Strategy, capabilities and value creation ..... 41
**6.1 Strategy as a governable intention ..... 42**
**6.2 Business model and value proposition ..... 42**
**6.3 Capabilities as a bridge between intent and execution ..... 43**
**6.4 Value streams and value creation ..... 43**
**6.5 Relationship between capabilities, services and processes ..... 44**
**6.6 Strategy Maps and Verification of Value Assumptions ..... 44**


All Rights Reserved – Dexon Software-2026

Chapter 7. Services and offers as a materialization of value ..... 45
**7.1 Service as a unit of business value ..... 45**
**7.2 Difference Between Capacity, Service, Offering, and Process ..... 46**
**7.3 Supply as a mechanism for accessing value ..... 46**
**7.4 Relationship with Dexon BPM ..... 47**
Chapter 8. Objective-Driven Process Design ..... 47
**8.1 Formal definition ..... 48**
**8.2 Objective of the process ..... 48**
**8.3 Expected Outcomes, Stakeholders and Satisfaction Criteria ..... 49**
**8.4 Restrictions, Risks, and Decisions ..... 49**
Risk Register as a Governed Object ..... 50
Two elements connect the register to the rest of the frame. ..... 50
**8.5 KPIs derived from the objective ..... 50**
**8.6 Formal specification of the objective of the process ..... 50**
**8.7 Semantics of Satisfaction, Decomposition, and Conflict ..... 52**
Chapter 9. Goal-oriented TO-BE design methodology ..... 54
**9.1 Methodological principle ..... 54**
**9.2 Phase 1 — Formulating the Intent of the Process ..... 55**
**9.3 Phase 2 — Define objective, results and satisfaction criteria ..... 56**
**9.4 Phase 3 — Analyzing AS-IS with Purpose ..... 56**
**9.5 Phase 4 — Design TO-BE trajectories ..... 56**
**9.6 Phase 5 — Designing Data, Rules, and Automation ..... 56**
**9.7 Phase 6 — Validate Goal Satisfaction ..... 57**
**9.8 Phase 7 — Prepare for Implementation, Measurement, and Governance ..... 57**
Chapter 10. Processes and data as an integrated system ..... 58
**10.1 The structural unity between process, data and result ..... 59**
**10.2 Ontological model and semantic coherence ..... 60**
**10.3 Data as Business Assets ..... 60**
Chapter 11. Information architecture and business data model ..... 61
**11.1 Business entities ..... 61**
**11.2 Relationships, Integrity, and Business Rules ..... 61**


All Rights Reserved – Dexon Software-2026

**11.3 Relationship with Dexon Designer and Related Data ..... 62**
Chapter 12. Data for decision-making ..... 62
**12.1 Operational decisions ..... 63**
**12.2 Decision Automation ..... 64**
**12.3 Execution Zones: People, Algorithms, and Agents ..... 65**
**12.4 The two ways of executing a trajectory ..... 67**
The conversion rule is declarative, not configurative ..... 68
The vocabulary of the decision is the results ..... 68
The two formal seams ..... 69
Consequence of traceability ..... 69
**12.5 Data Quality as a Condition of Automation ..... 69**
Chapter 13. Integration between processes, data and automation ..... 69
**13.1 Forms, Events, and Execution ..... 70**
**13.2 Integrations, ETL, and Services ..... 70**
**13.3 CMDB and ICs as operational support ..... 70**
**13.4 Role of Dexon Modules ..... 71**
Chapter 14. Data governance and operational governance ..... 71
**14.1 Process Owners and Data Owners ..... 71**
**14.2 Lineage, historicization and auditing ..... 72**
**14.3 Post-departure government ..... 72**
**14.4 Governing and Managing – The COBIT Distinction ..... 72**
**15.1 Indicators as proof of the objective ..... 74**
**15.2 Operational Evidence ..... 74**
**15.3 Measurement for Learning and Improvement ..... 74**
**15.4 Disciplined derivation of indicators: GQM and OKR ..... 75**
**15.5 The object of measurement: every measurement rests on an attribute of the metamodel ..... 75**
**15.6 Variables and indicators: one object, two roles ..... 77**
The composition and its dependency graph ..... 78
An Implementation Warning ..... 78
**15.7 Operating Model Outcome Attributes and Quality Attributes ..... 79**


All Rights Reserved – Dexon Software-2026

**15.8 Measurement alignment is relationships, not fields ..... 79**
**15.9 Alignment Boards ..... 80**
Chapter 16. Implementation of the executable architecture in Dexon BPM ..... 81
**16.1 To implement is to materialize architecture ..... 82**
**16.2 From methodological design to executable configuration ..... 82**
**16.3 Configuration as a governance contract ..... 83**
Chapter 17. End-to-end validation and testing ..... 83
**17.1 Nature of Validation ..... 84**
**17.2 Functional Validation ..... 85**
**17.3 Data and Rule Validation ..... 85**
**17.4 Validation of objectives and evidence ..... 85**
Chapter 18. Change Management and Organizational Adoption ..... 85
**18.1 Exchange as a condition of value ..... 86**
**18.2 Training by role ..... 86**
**18.3 Role governance and accountability ..... 86**
Chapter 19. Post-exit government ..... 87
**19.1 The process in production as living architecture ..... 87**
**19.2 Versioning and Reapproval ..... 87**
**19.3 Changes in Services, Capabilities, and Data ..... 88**
Chapter 20. Evidence-based continuous improvement ..... 89
**20.1 Evidence as a Learning Engine ..... 89**
**20.2 Improvement Cycle ..... 90**
**20.3 Iterative Redesign ..... 90**
**20.4 Process mining: evidence of execution as an object of analysis ..... 90**
Chapter 21. Dexon Maturity Model ..... 91
**21.1 Purpose of the model ..... 91**
**21.2 Advancement criteria ..... 93**
**21.3 Anchoring in Reference Maturity Models ..... 93**
Chapter 22. Interoperability with ITIL, certified ITSM practices, and artificial intelligence ..... 94
**22.1 The ITIL 4 Service Value System and Methodology ..... 94**
**22.2 Certified practices as a verified functional reference ..... 94**


All Rights Reserved – Dexon Software-2026

**22.3 Artificial Intelligence in Goal-Oriented Process Design ..... 96**

Chapter 23. Methodological conclusion ..... 97

References ..... 98


All Rights Reserved – Dexon Software-2026

# Chapter 1. Purpose, Nature, and Scope of the Methodology

*This chapter establishes the central thesis of the methodology: to transform strategy into an executable business architecture, rather than limiting process management to documentation or BPMN modeling. To this end, it proposes a traceability chain that connects strategy, capabilities, services, offers, processes, data and configuration elements, and indicators, so that each artifact has a business justification and a verifiable relationship with the others. The scope integrates business architecture, AS-IS and TO-BE design, data modeling, automation, measurement, governance, and improvement, underpinning fundamentals of strategy, enterprise architecture, BPM, data management, governance, and maturity. For top management, the value of the approach lies in turning strategic priorities into observable operational conditions: what the organization must know how to do, what value it offers, how it delivers it, what data it operates on and with what evidence it demonstrates results. The methodology is not intended to formulate an independent theory, but rather to integrate recognized frameworks into an applicable and governable method within Dexon BPM.*

## 1.1 Purpose of the methodology

This methodology establishes the conceptual and practical framework for modeling, redesigning, automating, measuring, and governing business processes in Dexon BPM from an integrated enterprise architecture vision. Its purpose is not only to produce BPMN diagrams or document data entities; Its purpose is to enable an organization to translate its strategic intent into operational capabilities, requestable services, executable processes, governed data, and verifiable indicators.


All Rights Reserved – Dexon Software-2026

*Figure 1. Proposed framework to turn the process objective into the design axis: strategy defines objectives, objectives are deployed into capabilities and results, BPMN models operational trajectories, and data/KPIs close.*

The methodology is organized around a chain of business traceability:
Strategy → Capabilities → Services → Offerings → Processes → Data/CIs → Indicators

This chain constitutes the ordering principle of the whole method. Each artifact must exist because it fulfills a function within it: the strategy defines intention; capabilities express what the organization must know how to do; services materialize value; the offers make that value accessible to users and customers; processes coordinate work; data and CIs support execution, decision-making and control; and the indicators make it possible to verify whether the expected result was achieved. This chain is the axis that orders the sections of this document and defines the order of implementation that the *Dexon Process Implementation Guide* develops operationally.

It is convenient to name precisely the problem that this chain aims to solve, because the discipline of architecture has identified it and has not closed it. In his reading of the state of the countryside, Villalobos (2026) points out four recurrent fractures: business and strategy are enunciated in a


All Rights Reserved – Dexon Software-2026

language that does not compromise anything; the business and the operating model are connected informally; the operating model breaks down into three independent worlds – organization, processes and resources; and artificial intelligence enters as a tool without criteria to distinguish its contribution from noise. The most structural of the four is the second. Between the map of capabilities and the operating model that must implement them, there is usually no declared mechanism: mapping is left to the discretion of each project, and that is why it is lost in each team rotation. This methodology exists to close that gap. The traceability chain, the meta-model in Chapter 4 and the process objective in Chapter 8 are, together, its answer.

Consequently, this methodology should not be understood as a process drawing manual. It should be understood as an **executable business architecture** methodology. A process is not considered complete because it has a diagram; It is considered complete when it can demonstrate traceability with a capability, a service, an offer, a data model, a set of rules, those responsible, indicators and a governance mechanism.

## 1.2 Methodological scope

The scope of this methodology comprises five integrated domains.


All Rights Reserved – Dexon Software-2026

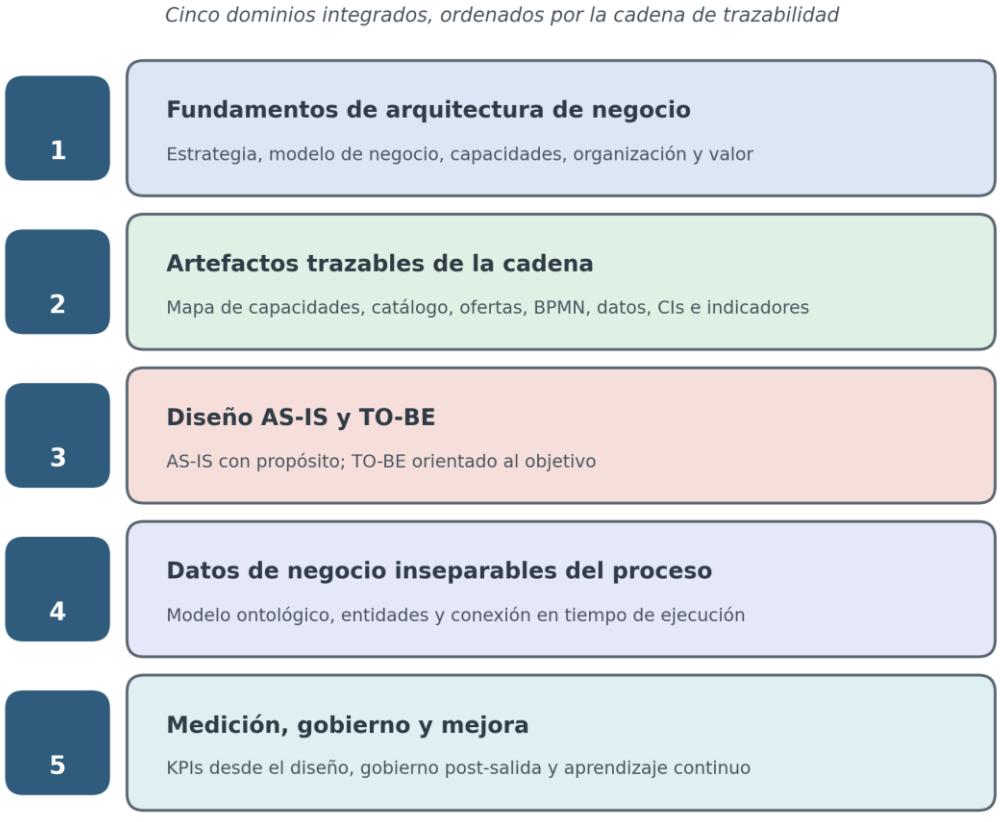


*Figure 2. The five integrated domains of methodological scope, ordered by traceability chain: fundamentals, traceable artifacts, AS-IS/TO-BE design, business data, and measurement-governance-improvement.*


All Rights Reserved – Dexon Software-2026

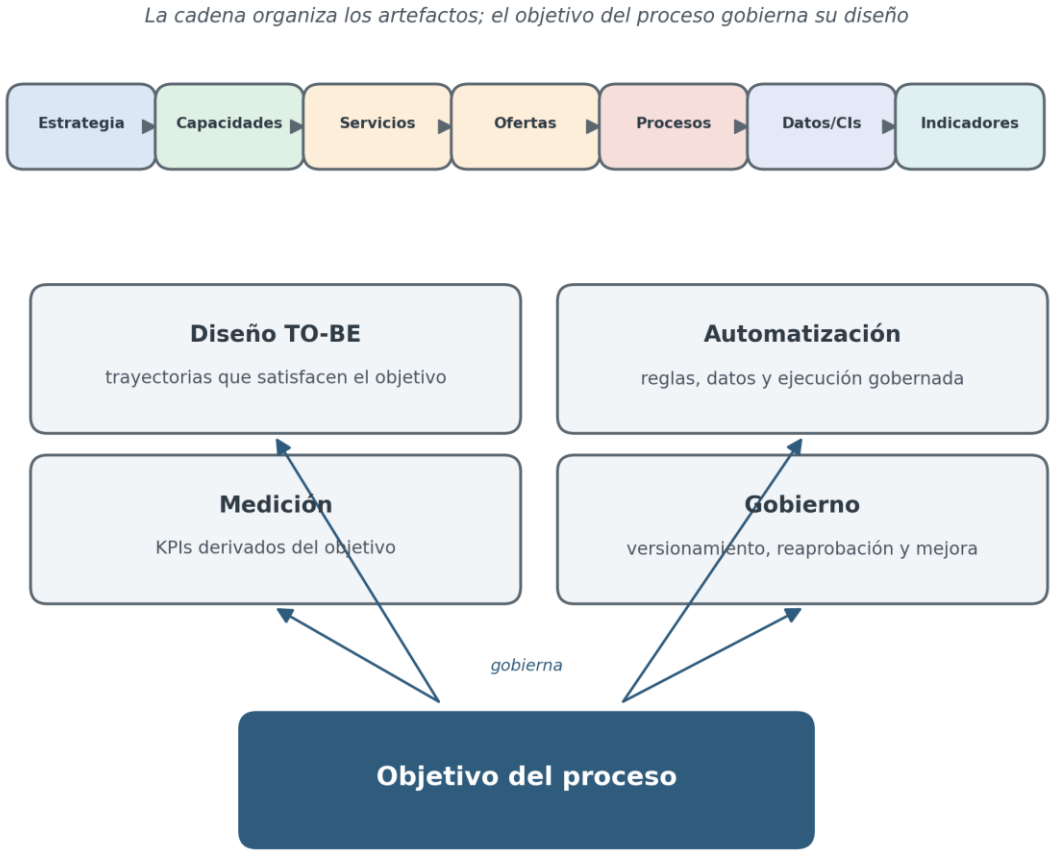


*Figure 3. The Dexon enterprise traceability chain organizes artifacts and process goal governs TO-BE design, automation, measurement, and governance.*

First, it defines the business architecture fundamentals necessary to connect strategy, business model, capabilities, organization, processes, resources, data, and indicators. This methodology adopts eight fundamental models – business model, financial, strategic, capabilities, organization, processes, resources and indicators – whose alignment is essential for the effectiveness, efficiency and transformation of the organization; this framework of eight models comes from *Villalobos' Readings for Business Architects* (2019).

Second, it establishes criteria for building and relating the capability map, service catalog, service offerings, BPMN processes, data model, CIs, and indicators. This relationship should not be implied in external presentations or documents; it must be recorded as formal traceability in Dexon BPM artifacts, where each link has its own editor and explicit relationships.

Third, it defines the AS-IS and TO-BE process modeling approach. The AS-IS is used to understand


All Rights Reserved – Dexon Software-2026

the operational reality at the level needed to redesign; TO-BE is used to build the future process that best meets the process objective. The operational guide already establishes that the process file must be filled out before drawing BPMN and includes identity, objective, classification, responsibility, relationships and SIPOC.

Fourth, it incorporates business data modeling as an inseparable component of process design. At Dexon, processes aren't just represented; they run on data, rules, forms, CIs, integrations, and indicators. The operational guidance defines that entities should be discovered from the ontological model, created in the table designer, and attached to processes using Related Data, which connects data and processes at runtime.

Fifth, it establishes the basis for measuring, governing, and improving processes after they are released to production. The operational guide requires defining KPIs with goal, meaning, frequency and owner, linking them to the real process and, when applicable, to strategic objectives.

## 1.3 Nature of the methodology

This methodology adopts BPMN as the process modeling language, but does not reduce process design to BPMN. BPMN is the standard language for representing business processes in a notation that is understandable for business users and sufficiently accurate for technical users; its purpose is to create a bridge between process design and its implementation (Object Management Group [OMG], 2013). At Dexon, BPMN is retained as a language of representation, communication, and automation, but is complemented by strategic traceability, capability architecture, data modeling, business rules, measurement, and governance.

The methodology is supported by a synthesis of business architecture, BPM, strategy and goal-oriented design. From TOGAF, he adopts the notion of capacity as the ability of the business to do something and the separation between what the business does and how it does it (The Open Group, 2022a, 2022b). From business architecture value streams, it adopts the idea of representing how value is created for a stakeholder and how that value is related to capabilities and processes (The Open Group, 2022c). From BPM CBOK, it adopts the need for common terminology, recognized best practices, and a professional vision of BPM not tied to a vendor (ABPMP, 2019). From GoalBPM, it adopts the need to explicitly relate BPMN models to stakeholder objectives and to verify whether the processes meet those objectives (Koliadis & Ghose, 2006).

The following figure summarizes the proposed methodological architecture. The traceability chain organizes the document and the process goal becomes the center of TO-BE design, automation, measurement, and improvement. This logic is consistent with the existing Dexon chain, with the materialization of its native publishers and with the literature of business architecture and Goal-Oriented BPM.

## 1.4 Theoretical foundations of the method

The soundness of a methodology does not depend on the number of frameworks it cites, but on the


All Rights Reserved – Dexon Software-2026

consistency with which it integrates them. This methodology is deliberately supported by four complementary theoretical pillars, each of which answers a different question of the problem of turning strategy into a governed operation.

The first pillar is the theory of strategy and organizational capabilities. De Porter (1996) adopts the distinction between strategy and operational effectiveness; Mintzberg (1987) and Mintzberg and Waters (1985), the recognition of the deliberate and emergent strategy; resource-based vision and dynamic capabilities (Barney, 1991; Teece et al., 1997), the explanation of why capacities—and not tasks—are the stable unit of analysis; and Kaplan and Norton (1996, 2004), the technique for translating strategy into causally linked objectives and indicators.

The second pillar is enterprise and business architecture. The TOGAF Standard and its guides to capabilities and value streams (The Open Group, 2022a, 2022b, 2022c) provide the definitions of capacity, value stream, and contract architecture; the BIZBOK Guide (Business Architecture Guild, 2026) provides the capacity-value mapping and the architectural knowledge base; and Ross, Weill, and Robertson (2006) provide the notion of operating model as an explicit decision to standardize and integrate processes. Within this pillar, the work of Jorge Villalobos (Universidad de los Andes) deserves special mention: his *Readings for Business Architects* were the foundational conceptual influence of the first versions of this methodology, his framework of eight interrelated models remains the basis of chapter 3, and his decomposition of the business model —in the current formulation— is partially adopted in section 3.2 (Villalobos, 2019, 2026). His most recent reading of the field also provides two elements that this document incorporates: the diagnosis of the gap between the skills map and the operating model that section 1.1 proposes, and the criterion for assigning work between people, algorithms and agents that section 12.3 develops (Villalobos, 2026).

The third pillar is the BPM discipline. The BPM CBOK (ABPMP, 2019) provides terminology and professional practices; the BPM life cycle (Dumas et al., 2018) provides the structure for identification, discovery, analysis, redesign, implementation, and monitoring; the BPMN and DMN standards (OMG, 2013, 2024) provide the formal languages of processes and decisions; and the goal-oriented line of research—KAOS and GoalBPM (Koliadis & Ghose, 2006)—provides the foundation for the goal-oriented design that this method operationalizes.

The fourth pillar is governance, data and maturity. COBIT (ISACA, 2018) and ITIL 4 (Axelos, 2019) provide the separation between governing and managing and the theory of service as co-creation of value; DAMA-DMBOK (DAMA International, 2017) provides the treatment of data as governable assets; and the maturity models BPMM (OMG, 2008) and PEMM (Hammer, 2007) anchor the maturity model in Chapter 21.
The contribution of this methodology is not to propose new theory, but to integrate these bodies of knowledge into an executable method on Dexon BPM, where traceability is not documented in static diagrams, but is configured, executed and measured on the platform. This positioning is consistent with the principles of good process management—context awareness, continuity, holism, and institutional ownership—formulated by vom Brocke et al. (2014).


All Rights Reserved – Dexon Software-2026

## 1.5 Scope of application and assumptions

This document is the **methodology for implementing executable business architecture on Dexon BPM**. The distinction matters to read it correctly: it is not presented as a general-purpose enterprise architecture framework that competes with TOGAF, nor as a platform-independent body of knowledge. It is the method with which implementations are designed, built, measured and governed on a specific platform, and that is why its rules can be more demanding and more verifiable than those of a general framework: it can be supported by mechanisms that it takes for granted.

The principles behind it—traceability as a condition of governance, objective as a design contract, measurement derived from the objective, government by design—are conceptually independent of the platform and could be applied in other environments. Operational rules are not: they presuppose an environment with five features that Dexon BPM implements natively.

| Assumption | What it demands from the environment |
|---|---|
| Related Object Repository | Architectural artifacts exist as objects with queryable relationships, not as documents |
| Running processes on the same repository | The process that is modeled is the one that is executed, without translation to another tool |
| Activity-Linked Data and Rules Engine | Mandatory decisions and data are configured, not programmed externally |
| Audit trail by instance | Each execution leaves a record of who, what, when, and with what data |
| Measurement linked to the real process | The indicators are fed by the operation, not by reports built separately |

Outside of an environment with these characteristics, a large part of the method is still applicable as a design discipline, but **it loses its distinctive property**: conformity is no longer verifiable by consultation and depends again on documentary review. It is advisable to be explicit about this rather than leaving ambiguity open.

The methodology also assumes three organizational prerequisites. First, that there is or can be established an **identifiable owner** by process and by critical entity: without named responsibility, R6 is inapplicable and the post-exit government becomes nominal. Second, that the organization agrees to **declare verifiable objectives**, which in practice means accepting that some processes will show that they do not meet what was expected of them. Third, there must be a **willingness to decide with data**: an organization that measures and does not use what is measured can implement the method and not obtain its benefit.


All Rights Reserved – Dexon Software-2026

## 1.6 Limits of applicability

Stating where a method does not apply does not weaken it: it is the most reliable sign that it has been used in real conditions. This methodology yields unevenly and there are contexts where its cost exceeds its benefit.

*Figure 4. Where the method performs, where it partially performs, and where its cost outweighs its benefit.*

**Where he performs fully.** Critical, repeatable processes with a stable objective; regulated or audited environments where evidence is required; operations with sufficient volume for measurement to be meaningful; transversal processes where coordination between areas is the main source of value loss; and organizations that have already assumed the cost of having process owners.

**Where it partially performs.** Knowledge processes with high variability, where the path is not predictable but the result is assessable: here the formal objective and satisfaction criteria retain all their value, but the design of trajectories loses usefulness and it is convenient to model the process as case management with verifiable milestones instead of as a sequence. It also partially performs in low-volume processes, where indicators take time to accumulate significant evidence.

**Where it is not convenient to apply it.** In genuinely exploratory or creative work, the outcome of which cannot be specified in advance without destroying its value: formalizing a verifiable objective


All Rights Reserved – Dexon Software-2026

for a research or conceptual design activity produces a fiction of control. In processes whose objective is unstable – because the business is still discovering what it wants to achieve – where the method must wait for the intention to stabilize or limit itself to implementing the exploration. In organizations without any data discipline or defined responsibility, where methodology is not the first problem to be solved. And in marginal processes of low risk and low frequency, where the effort of formalization is not recovered: applying the complete method to an internal procedure that is executed three times a year is a wrong government decision, even if it is methodologically impeccable.

**The cost of the method.** Rigour has a price and it should be said. The formulation of a well-formed objective, with declared results, criteria, thresholds, constraints and risks, is the work of analysis that consumes the time of scarce people – process owners, architects, data managers – before there is anything executable. The return on that investment comes later: in fewer design reprocesses, in changes that do not break the architecture and in government discussions that are resolved with evidence. But it comes later. An organization that needs visible results in weeks should start with the light version of the next section and not the full method.

## 1.7 Light Version of the Method

Not all processes justify the entire method. The methodology admits a minimum subset, applicable to processes of low risk, low volume or limited scope, which preserves what makes the architecture verifiable and omits what is only justified when there is criticality involved.

| Element | Complete method | Lightweight version |
|---|---|---|
| On-chain location | Mandatory, with declared capacity, service and offer | Mandatory: capacity and service. Offer only if requestable |
| Objective of the process | Nine components with well-formed rules | Purpose, a stakeholder, an expected result and a criterion with its threshold |
| AS-IS Analysis | Formal, with identification of gaps against the objective | Structured conversation, without modeling |
| Trajectories | Normal and exceptional, all governed | Main path plus critical exceptions |
| Data and rules | Full specification per activity | Required data by state and decision rules |
| Indicators | Five families, GQM referral | An achievement indicator and a flow indicator |
| Validation | Five-dimensional E2E with signed minutes | Functional testing of the main path and critical exceptions |


All Rights Reserved – Dexon Software-2026

| Element | Complete method | Lightweight version |
|---|---|---|
| Government | Versioning, Reapproval, and Impact Matrix | Owner and Approval Date |

The rule that governs the choice is not the size of the process but its **exposure**: if a failure of the process generates relevant regulatory, financial or reputational consequences, the complete method corresponds even if the process is small. A process can be three activities and be critical.

What the lightweight version **never omits** are the four conditions that make the architecture still searchable: location in the chain, identified owner, at least one threshold satisfaction criterion, and at least one indicator linked to the actual process. A process that renounces these four is not applying a light version of the method: it is outside of it, and must be recorded as such so that the maturity model does not count it as compliant.

# Chapter 2. Guiding principles

*The chapter defines the rules that protect method consistency and prevent an automation initiative from being reduced to digitizing existing inefficiencies. Their starting point is that architecture must precede automation: before setting up a flow, it is necessary to understand its link to capabilities, services, data, risks, and indicators. Traceability is presented as a condition of governance, not as a documentary requirement, and the objective of the process becomes the unit that guides design, evaluation and measurement. BPMN retains its function as a formal language of representation and execution, but it ceases to be the starting point; First, you define why the process exists and what verifiable result it should produce. The chapter also states that data is indispensable for execution, decision, auditing, and integration, and that measurement must be designed from the outset. For an executive committee, these principles offer a practical investment criterion: not approving isolated automations, but solutions whose contribution to value, control, and learning can be demonstrated.*

## 2.1 Architecture before automation

Every implementation project at Dexon must start with understanding the business architecture that makes sense of the process. Automating a process disconnected from capabilities, services, data, and indicators produces apparent efficiency, but not necessarily business value. The Dexon operational guide expresses this principle directly by warning that links should not be skipped: a process without service is an orphan flow, a service without capabilities cannot be analyzed and an offer without an initial layout is not requestable.

Automating doesn't mean moving current work faster. Automating means executing in a controlled manner an operational design that has been justified from the business model, the capabilities required, the service that is delivered, the data that is needed and the indicators that will measure the result. Davenport argues that formulating strategy is not enough: the processes that effectively implement it must also be designed, and technology is a central enabler of process innovation


All Rights Reserved – Dexon Software-2026

(Davenport, 1993).

## 2.2 Traceability as a condition of rigor

Traceability is not a documentary attribute; it is a condition of government. Each process must be able to answer: what strategic objective or business need it addresses, what capabilities it orchestrates, what service it materializes, what offer activates it, what data it consumes and generates, what ICs support it, what rules it uses and what indicators demonstrate its performance.

TOGAF states that capabilities should be mapped to other business perspectives—organization, value streams, information, applications, and plans—to produce a coherent vision and support better decisions (The Open Group, 2022b). The BIZBOK Guide v15 reinforces this approach by highlighting capability and value mapping, business architecture knowledgebase and results orientation to connect strategy, execution and investment (Business Architecture Guild, 2026). At Dexon, this traceability must materialize in explicit relationships within the platform, not in single files or subsequent interpretations.

## 2.3 The goal of the process as a design unit

Every process exists to achieve a result. Therefore, the objective of the process should not be a rhetorical phrase on the file; It must become the central criterion for designing, evaluating, automating, and measuring the process. GoalBPM states that stakeholder objectives must be explicit, communicated and traceable to the proposed, approved or implemented changes in the processes (Koliadis & Ghose, 2006).

In this methodology, the process objective functions as a design specification. It should state the expected result, the beneficiary stakeholder, the conditions of success, the relevant constraints, the necessary data and the indicators that will demonstrate compliance. A process can be fast, cheap, and automated, but if it doesn't meet its goal, it's still a bad process.

## 2.4 BPMN as a formal language, not as a starting point

BPMN remains a modeling language because it allows representing events, activities, gates, flows, roles, and collaborations with a standard that is understandable for business and technology (OMG, 2013). However, in this methodology BPMN is not the starting point. Before modeling the flow, the process must be characterized, its objective must be formulated, it must be located in the traceability chain and its conditions of success must be determined.

The Dexon operational guide already establishes this order: the process sheet must be prepared before drawing BPMN, including objective, classification, responsibility, relationships with value chain and service, and SIPOC inputs/outputs. The BPMN diagram is, therefore, the executable representation of a previous methodological decision.


All Rights Reserved – Dexon Software-2026

## 2.5 Data as a condition for execution, decision and measurement

Data is not a byproduct of the process. They are a condition for executing it, deciding within it, integrating it with other systems, auditing and measuring it. In Dexon, entities must be derived from the ontological model of the business and connected to processes through Related Data, configuring which tables are captured or queried, their enforceability by status, scope filters, and on-site editing.

This methodology adopts the thesis that there can be no rigorous TO-BE design without data design. Each BPMN activity must be clear about what data it requires, what data it produces, what rules it applies, what evidence it leaves and what indicator it feeds.

## 2.6 Measurement by design, not at the end

Measurement should not be added after automating. Indicators should be derived from the objective of the process. The operational guidance states that KPIs should have a goal, meaning, frequency, and ownership, and should be linked to the actual process and strategic objectives where they exist.

This principle is consistent with Khadem, for whom alignment requires connecting vision, strategy, roles, responsibilities, objectives, and indicators in the same execution logic (Khadem & Khadem, 2017). It also coincides with Rummler-Brache, whose approach seeks to strengthen the relationship between strategy, process redesign and a top-down measurement system as a basis for continuous improvement (Rummler & Brache, 2012).

# Chapter 3. Business Architecture Fundamentals

*This chapter presents business architecture as a structured, governable representation of the system by which the organization creates, delivers, and captures value. The view goes beyond the organizational chart and articulates eight interrelated models: business, finance, strategy, capabilities, organization, processes, resources and indicators. The business model connects the service portfolio, business structure and ontological model; the strategy incorporates both the deliberate intention and the emergent learning derived from the operation; and capabilities are used as a stable abstraction to decide investment, redesign and automation. The chapter precisely differentiates capabilities, services, and processes: capabilities express what the organization must know how to do, services materialize value for stakeholders, and processes coordinate activities, decisions, roles, data, and rules to produce results. Its main managerial implication is that transformation should not be organized around isolated areas or applications, but around critical capabilities, value streams, and an explicit operating model that determines which processes to standardize and which data to integrate.*


All Rights Reserved – Dexon Software-2026

## 3.1 Business architecture as a governable representation of the organization

Business architecture is a structured representation of how an organization creates, delivers, and captures value. Its function is not to describe the organization as an organizational chart, but to make visible the system by which strategic intent is converted into operation, information, measurement, and learning. This methodology defines business architecture as a coherent representation of how the organization creates, delivers, and captures value, supported by eight interrelated models. This representation – the organization understood as a complex but governable machine, described through business, financial, strategic, capabilities, organizational, process, resource and indicator models – comes directly from the work of Villalobos, whose readings on the business model, the strategic analysis, the capability map and the strategic model constitute the most direct conceptual influence of this methodology since its first versions (Villalobos, 2019, n.d.-a, n.d.-b, n.d.-c).

A rigorous business architecture allows answering questions that a simple documentation of processes cannot answer: what capabilities are critical for the strategy?, what services materialize the value proposition?, what processes execute those capabilities?, what data are necessary to control the operation?, which CIs support the service?, what indicators show that the value was delivered?

## 3.2 Business model, value proposition and service portfolio

The business model is the starting point because it defines how the organization creates, transforms, delivers, and transfers value. Villalobos treats it as the business in its purest state—before committing to a specific operating model—and breaks it down into seven pieces: customer paths, product architecture, customer architecture, portfolio of business services, business structure with its components, channels and activities, business scenarios, and business topology (Villalobos, 2019, 2026).

This methodology does not adopt the seven, and it is worth saying which ones it takes and why. Take three, because they are the ones that have a direct impact on the traceability chain: the **portfolio of business services**, which becomes the Services link and gives rise to the catalog and offers; the **business structure**, which provides the components, channels and activities on which the processes are based; and **the customer** routes, which feed into the design of value streams and offers from the perspective of the recipient of the value. The remaining four—product architecture, customer architecture, scenarios, and business topology—are legitimate business analysis tools that this method does not require to design an executable process; omitting them is preferable to quoting them without using them.

It is also necessary to correct an inaccuracy in previous versions of this document. The **ontological model** is not part of the decomposition of the business model in the frame of reference: it is a contribution of this methodology, developed in chapters 10 and 11, and its function is different. It does not describe how the organization creates value, but rather what concepts must share


All Rights Reserved – Dexon Software-2026

processes, data, and indicators for execution to be semantically consistent.

Osterwalder provides a complementary perspective: the Business Model Canvas describes how an organization creates, delivers, and captures value, and the Value Proposition Canvas allows you to design, test, build, and manage products and services that are integrated with the business model (Osterwalder & Pigneur, 2010; Osterwalder et al., 2014). At Dexon, this idea translates into the fact that the service portfolio should not be an administrative list, but an operational expression of the value proposition.

Porter complements this view by distinguishing strategy from operational effectiveness: strategy does not only consist of doing the same thing better, but also of doing different things or doing them differently to maintain a differentiated position (Porter, 1996). Therefore, the Dexon methodology must prevent the service catalog from becoming an inventory of requests; You should treat it as a strategic decision about what value is delivered, to whom, through what channels, and with what capabilities.

## 3.3 Deliberate strategy, emergent strategy and learning

The methodology recognizes that the strategy has deliberate and emergent components. Mintzberg defines strategy from multiple perspectives—plan, ploy, pattern, position, and perspective—and distinguishes between intentional, realized, and emergent strategy (Mintzberg, 1987; Mintzberg & Waters, 1985). This distinction is important because an implementation in Dexon not only automates an already defined strategy; It also generates operational data that uncovers patterns, gaps, exceptions, and opportunities for adjustment.
Salazar Yusti, from the perspective of emergent strategy, questions the obsolete ideas of strategic planning and proposes a theory on how strategy is practiced within organizations, embracing its emergent nature (Salazar Yusti, 2021). In this methodology, that perspective translates into a rule: automated processes must produce evidence to learn. A process doesn't just execute a strategy; It also reveals whether the strategy works in practice.

## 3.4 Capabilities, Processes, and Services

Capabilities represent what the organization must know how to do; Processes represent how that capability is executed in the operation. This methodology establishes that capabilities are stable, independent, and measurable skills, classified as missional, strategic, and operational, and that the capability map connects intention with operation, a structure inherited from Villalobos (n.d.-b).
TOGAF reinforces this definition by pointing out that a business capacity represents the business's ability to do something and separates the "what" from the "how", the "who" and the "where" (The Open Group, 2022b). Therefore, a process should not be confused with a capacity. "Manage claims" can be a capability; the BPMN process describes how a claim is received, classified, analyzed, decided, communicated, and closed.

Villalobos formulates this separation in more demanding terms, and more useful for design: a capacity is an individual and independent activity that the operating model must implement, and it


All Rights Reserved – Dexon Software-2026

is **a contract without any operational detail** —in his expression, the *APIs of the business* (Villalobos, 2026). The practical consequence is a criterion of good training that this methodology adopts: a capability is well enunciated if its contract can be formulated without naming a process, an area or a system. If when describing it it is inevitable to mention how it is done, who executes it or with what tool, what is being described is not the capacity but one of its implementations. This criterion complements the R5 constraint in section 4.6 with a test that the reviewer can apply directly to the statement, before looking at the repository.
Services, on the other hand, are the way in which the organization materializes value for a stakeholder. At Dexon, every service has an owner, value proposition, criticality, cost/price, capabilities that implement it, and CIs that support it. Offers are the commercial face of the service: name, description, visibility, validity and process that executes the request.

## 3.5 Processes as coordinated implementation of capabilities

A business process is a coordination of activities, decisions, roles, data, and rules aimed at producing a valuable result. Davenport defines process as a structured, measured set of activities designed to produce a specific outcome for a customer or market (Davenport, 1993). Rummler-Brache complements this vision by pointing out the importance of managing the "white space" of the organizational chart, that is, the interfaces where work crosses areas and where delays, reprocesses and loss of responsibility often appear (Rummler & Brache, 2012).

At Dexon, this idea is fundamental: a process does not belong only to one area, even if it has an organizational owner. A process materializes a service, orchestrates capabilities, consumes and generates data, uses CIs, complies with rules, and produces indicators. Its design must resolve the transversal coordination of the work.

## 3.6 Strategic Capability Foundation: Resources, Dynamic Capabilities and Operating Model

The centrality of capabilities in this methodology is not a modeling convention; It is based on strategic theory. The resource-based view holds that sustainable competitive advantage comes from valuable, rare, difficult to imitate, and organizationally leveraged resources and capabilities, rather than from product-market position alone (Barney, 1991). This thesis explains why the capability map—and not the organizational chart or the application inventory—is the stable abstraction on which investment, redesign, and automation are decided.

The theory of dynamic capabilities adds the temporal dimension: in changing environments, the advantage lies not only in existing operational capabilities, but in the organization's ability to detect opportunities and threats, capture them, and reconfigure its resource base (Teece et al., 1997; Teece, 2007). In terms of this methodology, the traceability chain and evidence-based measurement constitute the institutional detection mechanism; the goal-oriented TO-BE design, the capture mechanism; and the Dexon platform – where processes, data, rules and indicators are governed configuration and not dispersed code – the reconfiguration mechanism. An executable business


All Rights Reserved – Dexon Software-2026

architecture is, in this sense, a materialized dynamic capability.

Finally, the notion of operating model provides the cross-cutting design criterion: organizations must explicitly decide which processes to standardize and which data to integrate between business units, because that decision defines their foundation for execution (Ross, Weill, & Robertson, 2006). In Dexon practice, this decision is reflected in which processes are modeled as corporate, which entities of the ontological model are master and shared, and which indicators are consolidated at the enterprise level.

## 3.7 The company in production and the company that transforms it

The three types of capability—missionary, strategic, and operational—aren't just a map ranking. They separate two companies that coexist in the same organization and that the design should not confuse. The company in production—the one that serves customers, executes the business, and consumes resources—is sustained by missionary and operational capabilities. The company that transforms it – the one that models, understands, analyzes, diagnoses, designs and intervenes on the first – is based on strategic capabilities. Both have their own people, processes, and resources, and their objectives, rhythms, and indicators are different (Villalobos, 2026).

This distinction has three methodological consequences.

1. The first is of scope. The processes that this methodology designs with Objective-Driven Process Design are, for the most part, processes of the company in production: it is there where the objective of the process, its satisfaction criteria and its indicators make immediate operational sense. When the method is applied to processes of the company that transforms – project demand management, prioritization of initiatives, governance of architecture – it is still valid, but its indicators measure transformation capacity and not current value production. The maturity model in Chapter 21 reflects this difference when evaluating the management system and not just each process.
2. The second is responsibility. A process of the company in production has an owner in the operation; The design of this process has an owner in the company that transforms. Separating them avoids two symmetrical pathologies: the process office that designs without anyone in the operation being responsible for the result, and the operational area that modifies the process without anyone sustaining architectural coherence. The R6 restriction in section 4.6 requires sole proprietorship of the process; This distinction specifies where that owner resides.
3. The third is reflexive, and it is worth stating it bluntly: **this methodology is itself a process of the company that transforms**. It has one goal—to produce verifiable executable architecture—it has normal trajectories and exceptions, it consumes and produces data, and it can be measured. The seven phases of Chapter 9 are his TO-BE model; the repository is their source of evidence; The maturity model in Chapter 21 is your indicator of achievement. A methodology that demands of business processes what it does not demand of itself does not withstand the scrutiny of whoever must adopt it.


All Rights Reserved – Dexon Software-2026

# Chapter 4. The Dexon business traceability chain

*The chapter formalizes the architectural core of the methodology through the Strategy → Capabilities → Services → Offers → Processes → Data/CIs → Indicators chain. His decisive contribution is to turn that sequence into a verifiable metamodel, composed of objects, relationships, cardinalities, and integrity constraints, rather than leaving it as a conceptual statement. The methodology defines what should exist in the repository, how objectives, goals, initiatives, capabilities, services, offers, processes, activities, decisions, rules, workspaces, entities, ICs, measurement elements, roles, risks, and evidence are related, and what configurations are considered invalid. The fifteen integrity constraints prevent, among other defects, orphan processes, services without capabilities, unverifiable objectives, decisions without data, without criteria or without a declared question, work without a declared executor, poorly formed metrics, work families without a results contract and changes without impact analysis. For management, the chapter makes strategic alignment a demonstrable property of the business model: it allows you to query dependencies, demand unique responsibility, evaluate impact before changing, and establish methodological conformity with evidence extracted from the repository, not with subsequent interpretations.*

## 4.1 Definition

The Dexon enterprise traceability chain is the mechanism by which an organization connects intent, capability, service, operation, information, and measurement. Its function is to prevent models from becoming isolated pieces. The chain is:
Strategy → Capabilities → Services → Offerings → Processes → Data/CIs → Indicators

This chain is not just an ordering idea: it is an obligatory methodological principle: no process is approved if it cannot demonstrate its position in the chain. Sections 4.3 to 4.6 formalize this principle as a metamodel: they define the architectural objects in the chain, their relationships and cardinalities, the process categories, and the integrity constraints that any conformal model must satisfy.

## 4.2 Methodological reading of each link

**Strategy.** It defines what the organization intends to achieve, what position it wants to build, what value it wants to deliver, and what renunciations it accepts. Porter reminds us that strategy is not simply operational effectiveness, but doing different things or doing them differently (Porter, 1996).

**Capacities.** They express the skills that the organization needs to fulfill its intention. TOGAF states that capabilities are a stable abstraction of the business and can be mapped to organization, value streams, information, applications, and plans (The Open Group, 2022b).

**Services.** They materialize value to internal or external stakeholders. At Dexon, the catalog records owner, value proposition, criticality, cost/price, capabilities and associated ICs.

**Offers.** They make the service requestable. At Dexon, the offer is the commercial side of the catalog


All Rights Reserved – Dexon Software-2026

and can share the process with other offers; An offer is requestable when it is visible, in force and the process has an initial layout.
**Processes.** They run the service using BPMN, rules, forms, states, transitions, and mappings. The operational guidance states that the BPMN must have a clear beginning, explicit purposes, gates with named conditions, and a happy path legible from left to right.

**Data/CIs.** They support execution, decision-making, traceability and integration. In Dexon, data is connected to the process using Related Data, and governable elements can be promoted to CIs in the CMDB.

**Indicators.** They verify whether the process meets its objective. They must distinguish strategic achievement indicators and operational process indicators, and be linked to the actual process and strategic objectives when applicable.

### On the order of the chain and other decompositions

A reader familiar with other process architecture decompositions will notice that the order of this string is not universal. Some reference architectures read the sequence as a business model → business services → missional processes → business capabilities → procedures, that is, with capabilities *below* processes and not above services (Villalobos, 2026). The difference is not a fundamental disagreement; It is a change in the height of the term *capacity*, and it is convenient to make it explicit so that the chain is not read as inverted.

In these decompositions, the capacity occupies the level of the fine implementable unit: what a set of procedures performs, already within the operating model. In this methodology, *capability* designates the unit of the capability map—the stable, timeless skill that the organization must master, formulated as a contract without operational detail (section 3.4)—that predates any decisions about services, processes, or systems. Read in this way, both readings coincide: the reference framework itself places the capability map *before* the operating model and makes it necessary for the latter to implement each capability.

This methodology deliberately preserves the order of Strategy → Capabilities → Services → Offers → Processes for a practical, not doctrinal, reason: it is the order in which business planning actually occurs. An organization decides first what it should know how to do, then what value it packages into services, then how it makes it requestable, and finally what processes execute it. Reversing the sequence would force the design of processes before knowing what capacity they respond to, which is precisely the defect that the R1 constraint prevents.

## 4.3 The metamodel of the executable architecture

Up to this point, the traceability chain has been presented as a conceptual principle. A methodological framework of reference demands more: a formal definition of what types of objects exist, how they can be related, and what configurations are invalid. That definition is the metamodel: the model to which all models built with this methodology must conform. The discipline of enterprise


All Rights Reserved – Dexon Software-2026

architecture has a direct precedent in ArchiMate, whose expressive power comes precisely from defining a formal metamodel of concepts and relationships (The Open Group, 2023). The Dexon metamodel differs in two ways: it is domain-specific to the executable business architecture – it does not claim to cover the entire enterprise – and it is executable, because each type of object is materialized on the platform and each relationship is a real repository relationship, not a diagramming convention.

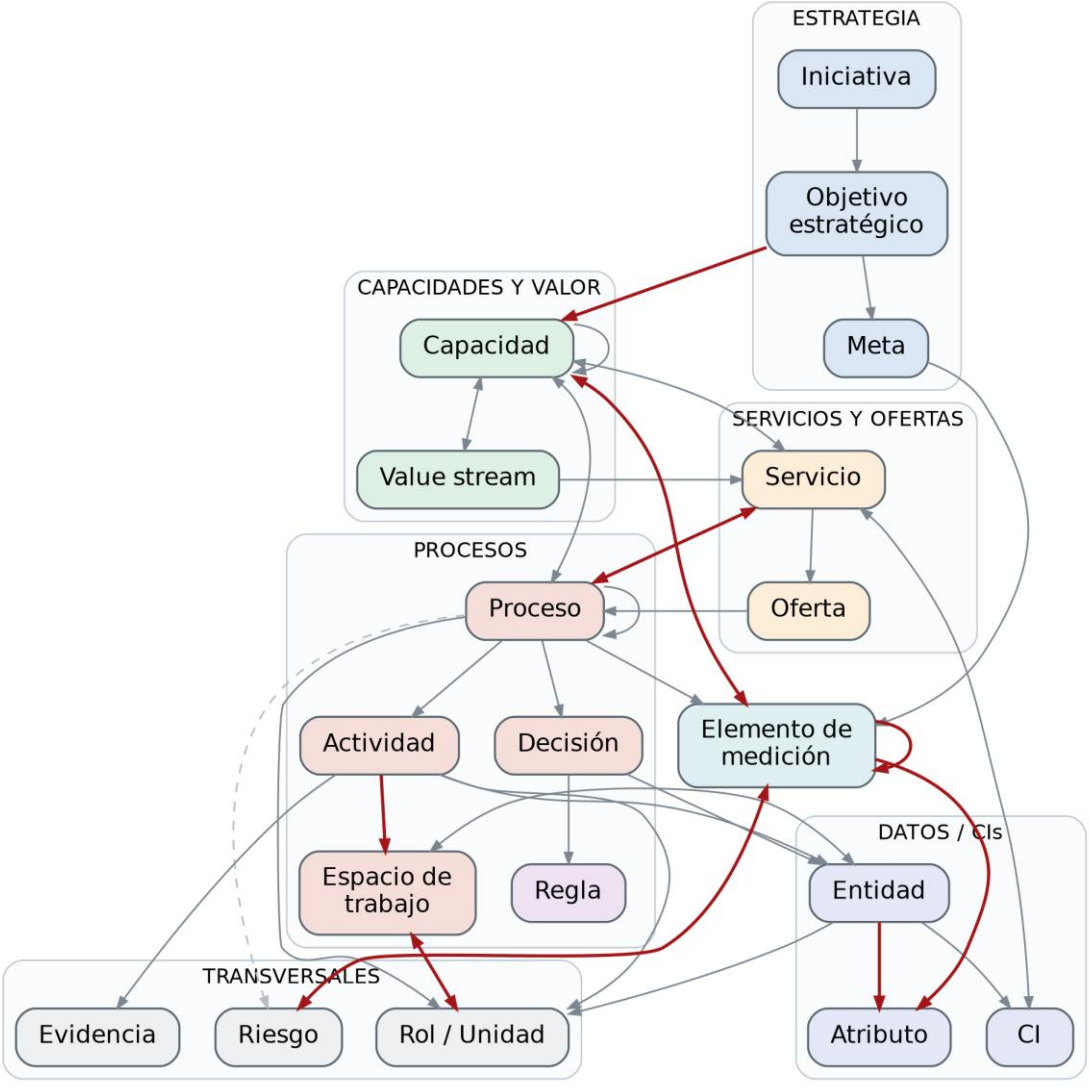


*Figure 5. Metamodel of the chain: structural objects and their relationships. In red, the relationships incorporated in the latest revisions: the workspace, the composition of the measurement element and its links to capacity and risk. In dotted line, the relationship between process and risk, which is established through the objective. The cardinalities of each relationship are detailed in the tables in Section 4.4, and the design contract relationships in the following illustration.*

The metamodel distinguishes three kinds of element, and that distinction is deliberate. Structural **objects** are the artifacts that your organization creates, approves, and versions: they have their own identity and persist in the repository. The **target specification objects** are the components of the design contract that Chapter 8 develops: they exist as typed records and not as free text, because


All Rights Reserved – Dexon Software-2026

otherwise the integrity constraints could not be assessed. Derived **concepts and roles** are not stored as independent objects: they are calculated from others or are roles that an object plays in a relationship. Separating them avoids the mistake of treating as data what is the result of a calculation.

This version incorporates a structural object that previous versions left implicit: the **workspace**. The chain went all the way to the data and indicators, but the point at which a person—or an agent—actually does the work had no name in the model. Without this purpose, three legitimate questions of a regulated organization remained unanswered: where each activity is carried out, who or what executes it, and where the operational load accumulates. Declaring it closes the chain to the place where the operating model becomes observable.

## Structural objects in the chain

| Purpose | Definition | Link |
|---|---|---|
| Strategic objective | An outcome that the organization is committed to achieving within a defined horizon | Strategy |
| Goal | Quantification of a strategic objective in a period, with a threshold and responsible | Strategy |
| Initiative | Temporary effort that mobilizes resources toward one or more goals | Strategy |
| Capacity | Stable skill that the organization must master; typified (missionary, strategic, operational) and with criticality declared on the sheets | Capabilities |
| Value stream | Sequence of stages that add value from end to end to produce an outcome for a stakeholder | Capabilities / Services |
| Service | Unit of value offered to a stakeholder, with owner, value proposition and criticality | Services |
| Offer | Requestable form of consumption of a service, with visibility and validity | Offers |
| Process | Executable coordination of activities, decisions, data and rules, with objective and owner; Classified as requestable, unrequestable, or reusable thread (see 4.5) | Processes |
| Activity | Human or automatic unit of work within a process, with data, rules, responsible parties and evidence | Processes |


All Rights Reserved – Dexon Software-2026

| Purpose | Definition | Link |
|---|---|---|
| Decision | Point in the process where a course of action is selected based on explicit data and criteria | Processes |
| Workspace | Environment where an executor performs the work of an activity: tray, form, portal, console, or integration interface. It is the point at which the operating model becomes observable and where its quality attributes are measured (see 15.7) | Processes |
| Business Rule | Restriction, calculation, validation, or policy governing activities and decisions | Processes / Data |
| Business Entity | Concept of the stabilized domain in the ontological model and structured as data | Data |
| Attribute | Declared ownership of an entity or other object in the metamodel. It is the unit on which all measurement rests (see 15.5) | Data |
| CI | Governable element that supports services and processes, subject to configuration management | Data/CIs |
| Measuring element | A declared magnitude that quantifies an attribute of a measuring object. Its anchor is a triplet—type of object, concrete record, and attribute (see 15.5). It has unity, source and periodicity, and formula if derived. It plays the role of variable or indicator depending on whether or not it declares decision criteria (see 15.6) | Indicators |
| Organizational Role/Unit | Institutional responsibility for artifacts and activities. When the role executes work, its type of executor is also typified: human, algorithmic or agent (see 12.3) | Transversal |
| Risk | Condition of uncertainty that requires a design decision in the process | Transversal |
| Evidence | Verifiable record of execution: data, documents, signatures, traces | Transversal |

## Target Specification Objects

These five types shape the design contract that Section 8.6 specifies as a tuple. They are declared here as objects of the metamodel because the constraints R4 and R7, and the derivation of indicators in Chapter 15, require individual consultation.


All Rights Reserved – Dexon Software-2026

| Purpose | Definition | Link |
|---|---|---|
| Objective of the process | Contract for the design of a process. It groups purpose, stakeholders, results, criteria, thresholds, restrictions, risks, required capabilities and evidence | Processes |
| Expected Result | A business state that must be achieved when the process completes successfully, expressed on ontological model entities and not on interface artifacts | Processes |
| Satisfaction criterion | An assessable predicate that makes an expected result verifiable; Attribute Reference with Identified Data Source | Processes |
| Threshold | Quantification of the permissible tolerance of a criterion: goal, band and direction (maximize, minimize or range) | Processes / Indicators |
| Restriction | Non-negotiable condition of the design, typified as time, cost, compliance, quality, risk or experience | Processes |

## Derived concepts and roles

| Concept | Nature | Definition |
|---|---|---|
| Career | Derived from the BPMN model | Possible path of an instance from a start event to an end event. It is calculated from the process model; It is not stored as a separate record |
| Governed exception | Classification of a trajectory | A trajectory whose purpose resolves, compensates, scales, rejects or closes with evidence a deviation foreseen in the design |
| Measuring object | Role, not type | The role played by any structural object when a measuring element quantifies one of its attributes. Does not add a type to the metamodel (see 15.5) |
| Variable | Role, not type | A measurement element that is declared, captured and accumulates values over time, but does not have a decision criterion: it has no goal, threshold or deployment. It is an input for other measurement elements (see 15.6) |
| Indicator (KPI) | Role, not type | A measurement element that also declares decision criteria – goal and threshold with meaning – and owner, and therefore enables a management decision (see 15.6) |

The practical consequence of this organization is that the design contract of Chapter 8 and the metamodel of this chapter are no longer two separate formalizations: the nine components of the goal tuple correspond to objects and relations declared here, and can therefore be verified with the same mechanism with which the rest of the chain is verified.


All Rights Reserved – Dexon Software-2026

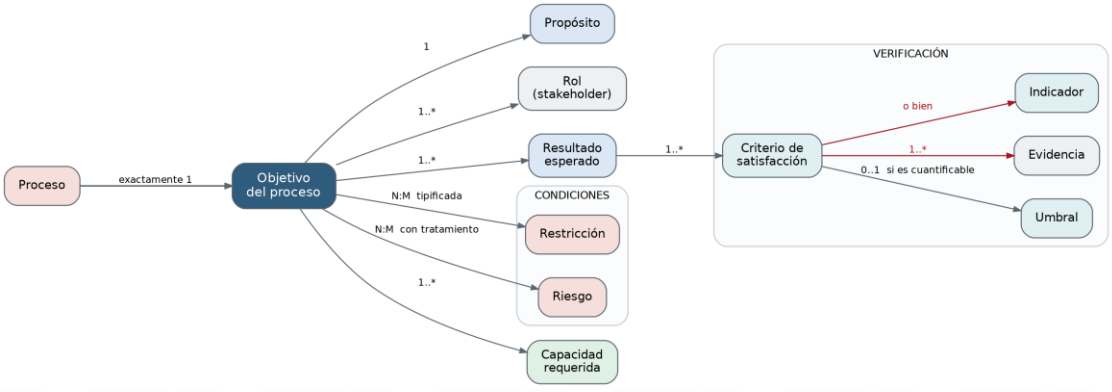


*Figure 6. The objective of the process as the object of the metamodel: its nine typed components and the cardinalities that make good training verifiable.*

## 4.4 Relationships and cardinalities

Metamodel relationships are not lines in a diagram; are the structural constraints that make traceability verifiable. Each one has a precise methodological reading.

### Chain Relationships

| Relationship | Cardinality | Methodological reading |
|---|---|---|
| Strategic Objective → Goal | 1 to 1..* | An objective without quantified goals is not governable |
| Goal → Measurement Element (Indicator Role) | 1 to 1..* | Every goal is verified with at least one indicator |
| Initiative → Strategic Objective | N:M | Every investment declares what objectives it mobilizes |
| Strategic ↔ Objective Capacity | N:M, with a declared role | Closes the first link in the chain: each link declares whether the capability is required, enabler, or impacted by the objective |
| Capacity → Capacity | 0..1 father | Hierarchy of composition; Type and criticality only in sheets |
| Value stream capability ↔ | N:M | Each value stage declares which capabilities enable it |
| Value stream → Service | N:M | The value stream is materialized in one or more services |


All Rights Reserved – Dexon Software-2026

| Relationship | Cardinality | Methodological reading |
|---|---|---|
| Service Capacity ↔ | N:M, min. 1 per serving | A service without capabilities has no delivery support |
| Process Capacity ↔ | N:M, min. 1 per process | No process is approved without declaring which orchestral capabilities |
| Process Service ↔ | N:M, min. 1 service per approved process | Direct link and not inferred by transitivity: also covers processes that do not have an offer |
| Service → Offer | 1 to 0..* | The service exists without an offer, but only the offer makes it requestable |
| Offer → Process | 0..* to 1 | Several offers can share process; Every offer executes exactly one |
| CI Service ↔ | N:M | Basis of the impact analysis between value and resources |
| Process → Process (Invocation) | 0..* | A process can invoke reusable threads |
| Process → Activity | 1 to 1..* | The process is the unity of government; the activity, the unit of work |
| Decision Process → | 0..* | The decisions of the process are declared individually |
| Activity ↔ Entity | N:M with Consume/Spawn role | Each activity declares what you read and what you write |
| Activity → Evidence | 1 to 0..* | Execution leaves verifiable record |
| Workspace → Activity | 1 to 0..1; mandatory if the executor is a human or agent | Work happens in a declared place, not in an anonymous system |
| ↔ Workspace Role | N:M, with Executor Type Declared | It is known who – or what – operates in each space |
| Entity workspace ↔ | N:M with role exposes/captures | The space declares which data it presents and which it collects |
| Decision → Rule, Entity | 1..* and 1..* | No decision without explicit criteria and data |


All Rights Reserved – Dexon Software-2026

| Relationship | Cardinality | Methodological reading |
|---|---|---|
| Entity → Attribute | 1 to 1..* | Attributes are the measurable unit and must be declared |
| CI → Entity | 0..1 (promotion) | What is governable inventory is promoted to the CMDB |
| Process → Measuring Element (Indicator Role) | 1 to 1..*, min. 1 achievement | A process without an achievement indicator cannot demonstrate its objective |
| Measuring Element → Measuring Object | exactly 1, with specific type and record declared | The anchor identifies *which* object is being measured, not just what kind it is (see 15.5) |
| Measuring Element → Attribute | exactly 1, of the measuring object | Every magnitude – variable or indicator – quantifies a declared attribute; There are no floating metrics (see 15.5) |
| Measuring Element Capacity ↔ | N:M | Allows multiple breaches to be read as a single capability gap (see 15.8) |
| Measuring Element Risk ↔ | N:M | The link that makes an element KRI: the KRI is a relationship, not a box (see 15.8) |
| Measuring Element → Alignment Board | 1 to 0..* | Each assignment declares responsibility, alignment role, goal, and weight of its own (see 15.9) |
| Measuring Element → Measuring Element | 0..*, acyclic graph | The formula of one element can reference others; Dependence is declared and the calculation respects it (see 15.6) |

## Design contract relationships

| Relationship | Cardinality | Methodological reading |
|---|---|---|
| Process → Process Objective | Exactly 1 | Every process has a single design contract |
| Objective of the process → Role (stakeholder) | 1..* | The value is defined by who receives or evaluates it |
| Process Objective → Expected Result | 1..* | A goal without a stated result is not verifiable |
| Process Objective ↔ Capacity | 1..* Required | The objective states what skills it requires from the business |



All Rights Reserved – Dexon Software-2026

| Relationship | Cardinality | Methodological reading |
|---|---|---|
| Purpose of the process ↔ Restriction | N:M | Non-negotiable conditions are declared before modeling |
| Objective of the Risk process ↔ | N:M, with declared treatment | Every critical risk has a visible response in the TO-BE |
| Expected Result → Satisfaction Criterion | 1..* | Each result is made verifiable by evaluable predicates |
| Satisfaction Criterion → Threshold | 0..1; mandatory if the criterion is quantifiable | Tolerance is declared, not negotiated after non-compliance |
| Satisfaction Criterion → Evidence or Indicator | 1..* | No criterion is left without a way of being verified |

### Transversal relationships of responsibility

| Relationship | Cardinality | Methodological reading |
|---|---|---|
| Process → Role (owner) | Exactly 1 | Sole and institutional responsibility |
| Activity → Role (responsible) | 1..* | No activity without identifiable responsible |
| Decision → Role (responsible) | 0..1; mandatory if the decision is human | When a person decides, you know who |
| Entity → Role (data owner) | exactly 1 in critical entities | Data quality has a person in charge appointed |
| Measuring Element (Indicator Role) → Role (Owner) | Exactly 1 | Every indicator has someone to answer for its reading |

## 4.5 Process categories and traceability required

Not all processes are requested from a catalog. The methodology recognises three categories and requires them to be traceable in a differentiated way, because applying the same requirements to all of them would produce either rules that are impossible to comply with or generalised laxity.

| Category | How to get started | Traceability required |
|---|---|---|
| Requestable process | Request a user from the catalog | Service, at least one capacity, visible and current offer, process objective and achievement indicator |
| Non-Solicitable Process | Event, integration, time schedule, or | Service, at least one capability, process objective and achievement indicator, and statement of trigger mechanism. No offer required |


All Rights Reserved – Dexon Software-2026

| Category | How to get started | Traceability required |
|---|---|---|
| | change to a data source | |
| Reusable thread | Invocation from another process | At least one invoking process and its own objective subordinate to that of the process invoking it. Inherits service and capabilities from the summoner; No bid required or self-achievement indicator required |

*Aplicar los mismos requisitos a las tres categorías produce reglas imposibles o laxitud*

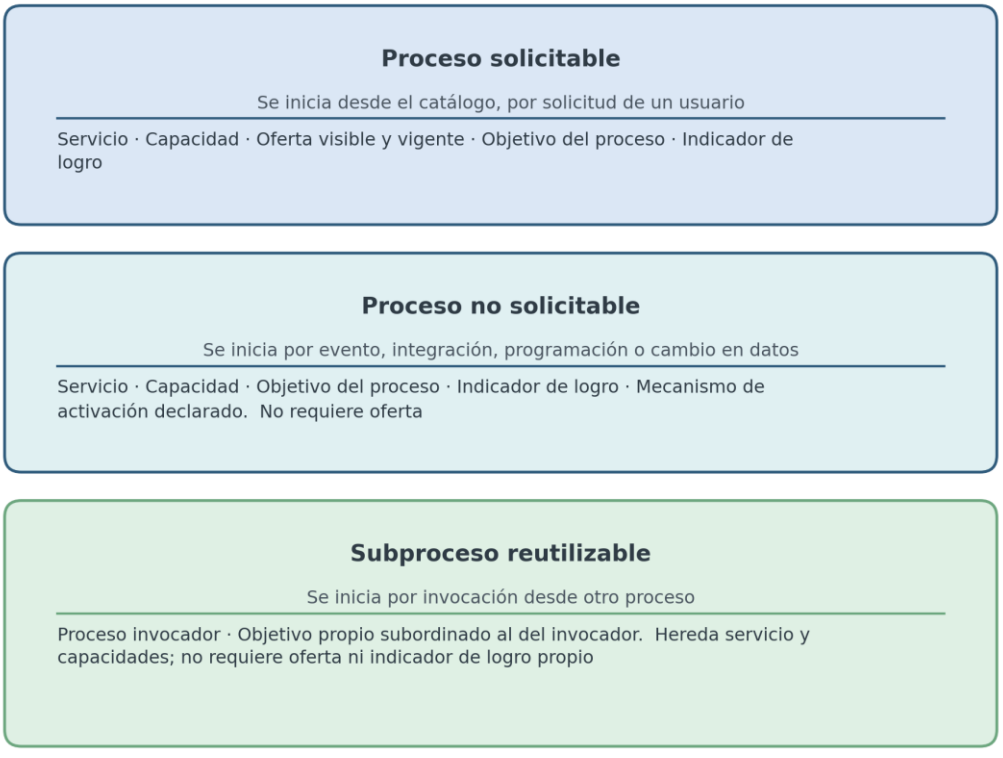


*Figure 7. The three categories of process and the traceability required of each one.*

This distinction resolves an ambiguity of previous versions, in which certain rules seemed to require supply to every process despite the fact that the method itself supports event-driven starts, integration, and automation.

## 4.6 Architectural integrity restrictions

Cardinalities define what is structurally possible; Integrity constraints define what is methodologically valid. A model can be constructible on the platform and still violate the method. The following fifteen restrictions, scattered until now as rules in different chapters, are consolidated


All Rights Reserved – Dexon Software-2026

as the formal criterion of conformity:

- **R1 — No orphan proceedings.** Every process – requestable or not – is linked to at least one service that materializes. Reusable threads inherit that binding and must declare at least one invoking process. A flow without a service or invocator process does not enter the approved repository.
- **R2 — No empty service.** Every service declares at least one capability that implements it and, if operable, the CIs that support it.
- **R3 — Complete Requestable Offer.** Every requestable process has at least one offer, and an offer is only requestable if it is visible, in force and its process has an operational entry point. Non-requestable processes are exempt and declare the activation mechanism instead.
- **R4 — Well-formed target.** Every approved process has exactly one process objective that satisfies the good training rules of section 8.6: at least one stakeholder, at least one expected outcome, at least one satisfaction criterion per outcome, threshold stated in the quantifiable criteria, typified constraints, treatment declared for each critical risk, at least one required capability, and at least one evidence or indicator per criterion.
- **R5 — Well-trained skills.** Every leaf capability has type and criticality; packets are grouped and not executed.
- **R6 — Sole liability.** Every process has exactly one owner with a current approval date; every critical entity has a data owner; every indicator has an owner.
- **R7 — Closed trajectories.** Every trajectory of the TO-BE model ends at an end that corresponds to a declared expected outcome or a governed exception.
- **R8 — Data with purpose.** All captured data declares its function: to execute, decide, fulfill, evidence, measure or integrate.
- **R9 — Explicit decisions.** Every decision in the process identifies its input data, its rule or its responsible, and the evidence it leaves.
- **R10 — Change with assessed impact.** No changes to an object in the metamodel are approved without querying its relationships in the live repository.
- **R11 — Work with a declared place and executor.** Every executable activity declares its class of executor—human, algorithmic, or agent—and, when the executor is a human or agent, the workspace where it is performed. An activity without a typified executor prevents deciding whether its automation was a design decision or a configuration accident, and leaves unanswered the question that every audit asks in a regulated organization: what part of this process is executed by a person and what part is not.
- **R12 — Well-formed measurement.** Every measuring element declares its role. If the role is indicative, the goal, the threshold with declared meaning and the owner are mandatory; If the role is variable, so are the data source and the frequency of capture. A magnitude without a declared role leaves undefined whether its absence of a goal is a design decision or an oversight.
- **R13 — Acyclic measurement dependencies.** The reference graph between measurement elements does not contain cycles. A formula that depends, directly or indirectly, on its own result is not calculable, and no order of evaluation solves it.
- **R14 — Decisions explained.** Every decision gate of a trajectory declares the **question** that is


All Rights Reserved – Dexon Software-2026

resolved in it, and its exits are the **possible answers**. A branching without question documents *where* it is decided, but not *what* is decided: it is the modeled ambiguity of section 12.1 surviving within the artifact. R14 is R9 verifiable fact on the diagram — R9 requires data and rule; R14 requires that the intention of the decision be written where it is drawn.

- **R15 — Contract of the working family.** Any activity that **outputs results** —that is, the completion of which decides which activity follows in a case family—provides the person who closes it with the corresponding outcome element. Without it, no one can choose the outcome and the trajectory never advances: it is an open trajectory disguised as an incomplete configuration. R15 derives from R3 and R7 for the concurrent execution mode that Section 12.4 develops.

## How each constraint is verified

These restrictions are not all checked in the same way, and it should be said precisely: to affirm that the fifteen are verified with a consultation would be inaccurate and would weaken the argument. The following classification indicates what the platform can automate, what requires analysis of the process model, and what still requires human judgment.


All Rights Reserved – Dexon Software-2026

*Figure 8. The fifteen integrity restrictions according to their mode of verification.*

| Restriction | Verification Mode | What is actually tested |
|---|---|---|
| R1, R2, R3, R5, R6, R11, R12, R15 | Consult the repository | Existence and cardinality of relations between declared objects. Fully automatable. R15 is assessed as a violation: without a performance contract there is no possible execution |
| R14 | Consult the repository | Presence of the question declared at each decision gate. It is evaluated as a warning, because the question is design documentation and its quality is judged by the reviewer |
| R4 | Consult the repository | Presence and good training of the components of the objective. The query verifies the structure; The relevance of the content corresponds to the approval of the owner |


All Rights Reserved – Dexon Software-2026

| Restriction | Verification Mode | What is actually tested |
|---|---|---|
| R10 | Consult the repository | Record that the impact analysis was executed before approving the change |
| R7 | Model Analysis | Reachability property on the process graph: it requires going through the TO-BE model, not consulting records. In models with cycles and concurrency, the analysis must be limited to the trajectories declared as relevant in the design |
| R13 | Model Analysis | Absence of cycles in the graph of dependencies between measurement elements: it requires going through the graph, not consulting records. It is the same check that the platform solves when ordering the calculation |
| R8, R9 | See more expert judgment | The query verifies that the purpose, data, and rule are declared; that the declaration is substantive and not formulated, it is up to the design reviewer |

This classification has two deliberate consequences. First, it honestly delineates the scope of government by design from Chapter 16: the platform can prevent or alert the violation of verifiable constraints by query, while R7 and R13 require a check on a graph, R14 is issued as a warning, and R8 and R9 retain a criterion component. Second, it defines conformity: an implementation conforms to this methodology when its repository satisfies R1 through R15 and its model respects the cardinalities of section 4.4, which makes methodological conformity a demonstrable property and gives the maturity model in Chapter 21 its basis for objective evaluation.

## Relationship with the notion of reference architecture

The practice of enterprise architecture usually interposes, between the capability map and the operating model, a layer of *reference architectures*: pre-established design decisions that delimit how capabilities will be implemented in the organization, in processes, and in technology (Villalobos, 2026). This methodology does not introduce that layer as an additional structure, because it already has it under another name. The metamodel in section 4.3, the cardinalities in 4.4, and the fifteen constraints in this section **are** the process reference architecture of the method: they do their job—predetermine which configurations are permissible before a particular process is designed—and add a property that a document reference architecture does not have, which is to be verifiable on the repository. Naming it that way facilitates dialogue with enterprise architecture teams that work with that vocabulary, without duplicating the structure of the method.

## The scope of the word "formal"

In this version, *formal* means typed structure, relationships with declared cardinality, rules of good formation and verifiable criteria. It does not yet mean complete mathematical semantics: the metamodel does not define domains of interpretation, inference rules, exhaustive treatment of


All Rights Reserved – Dexon Software-2026

cycles and BPMN concurrency or a demonstrably correct verification algorithm. That formalization is a possible development, not a property that this document claims.

# Chapter 5. Introduction to Objective-Driven Process Design

*This chapter introduces Objective-Driven Process Design as an approach correction to TO-BE design that is limited to optimizing the current process. The central question is no longer how to execute faster or automate more activities and becomes whether the future process, when executed, satisfies the objective for which it exists. The objective is treated as a design specification that must declare outcome, stakeholder, success conditions, constraints, data, decisions, exceptions, and indicators. BPMN remains indispensable, but every activity, gate, and event must justify its contribution to the objective, to a rule, to a risk, to an obligation of evidence, or to measurement. The chapter also establishes five conditions for approval of the TO-BE: cumulative linkage with capacity and service – and with the offer when the process is requestable; normal outcome-oriented trajectories; governed exceptions; sufficient data to execute, decide and audit; and indicators of achievement, quality, timeliness, compliance and improvement. For senior management, this approach avoids confusing local efficiency with business effectiveness.*

## 5.1 Definition

**Objective-Driven Process Design** is the methodological approach by which a TO-BE process is designed, evaluated, automated, measured, and governed based on an explicit, verifiable, and traceable objective. This approach maintains BPMN as the modeling language, but changes the core of the design: the starting point is no longer the sequence of activities, but the result that the process must achieve. This chapter presents the motivation of the approach and the way to formulate process objectives; its formal specification as a design contract, together with the detailed criteria for approval of the TO-BE, is developed in Chapter 8, and its phased deployment in Chapter 9.

The goal of the process becomes a design specification. It must define what outcome is expected, for whom value is produced, under what conditions it is considered fulfilled, what constraints must be respected, what data is necessary, what decisions need to be made, what exceptions need to be controlled, and what indicators will demonstrate achievement.

## 5.2 Difference from traditional TO-BE

Traditional TO-BE often starts with AS-IS and asks how to make it faster, cheaper, or more automatable. That approach is useful, but insufficient. As the tradition of process reengineering warned, automating a fragmented structure can accelerate a bad design instead of solving it; the redesign must begin with the client, the result and the complete workflow required to produce it (Hammer, 1990; Hammer & Champy, 1993).

Objective-Driven Process Design starts from another question: **if this process is executed as designed, does it satisfy the objective for which it exists?** Only then are efficiency, automation, task


All Rights Reserved – Dexon Software-2026

assignment, forms, integrations, and technology evaluated. GoalBPM provides the conceptual basis for this decision: processes must relate to stakeholder objectives and it must be verified whether their trajectories meet those objectives (Koliadis & Ghose, 2006).

## 5.3 Relationship to BPMN

BPMN is still indispensable. It allows representing events, tasks, gates, participants, messages, exceptions, and flows with a common notation for business and technology (OMG, 2013). However, in this methodology, each BPMN element must be justified by its contribution to the objective, constraint, rule, data need, governed exception, or indicator.

An activity that does not contribute to the objective, does not capture necessary evidence, does not execute a rule, does not reduce risk, does not deliver value or enable measurement must be questioned. A gate without explicit data or rules is not a decision; it is modeled ambiguity. A BPMN end that does not correspond to an expected result or a controlled exception does not methodologically close the process.

## 5.4 Formulation of the process objective

The objective of the process must be formulated with a minimum structure:
[Outcome verb] + [business object] + [beneficiary stakeholder] + [success condition] + [relevant constraints] + [measurement evidence].

Example:
"Manage return requests so that the customer receives a correct, traceable and timely response, complying with commercial policies, controlling fraud risks and generating reliable information for root cause analysis and service improvement."

This formulation avoids weak goals such as "manage returns" or "automate requests". The robust objective declares value, quality, timeliness, compliance, risk, and learning. Thus, the process is not designed to move cases; It is designed to produce a verifiable result.

## 5.5 TO-BE approval criteria

A TO-BE model should only be approved when it demonstrates five conditions:

1. The process has a well-formed objective and is linked to at least one capability and to the service it provides and, when it is requested, to the offer that activates it.
2. Every normal BPMN trajectory leads to a result that satisfies the goal.
3. Exceptional trajectories are governed and do not remain as informal ruptures.
4. The data required to decide, execute, audit, and measure are defined.
5. The indicators allow the evaluation of achievement, quality, timeliness, compliance and improvement.

This logic transforms the TO-BE into an architectural artifact and not just an operational diagram.


All Rights Reserved – Dexon Software-2026

The process ceases to be a sequence of tasks and becomes a governed mechanism for producing value, generating reliable data, and learning from its own execution.

# Chapter 6. Strategy, capabilities and value creation

*This chapter explains how to turn strategy into a governable, deployable intent about the operation. Strategy is not understood as a separate statement of execution, but as a system of objectives, goals, capacities, services, processes, data and indicators that must be formulated, executed, measured and adjusted with evidence. The business model defines how value is created, delivered and captured; capabilities express what the organization must master to fulfill that intention; Value streams show, from the perspective of the stakeholder, how value is generated from end to end; and services and processes materialize and execute that promise. The chapter differentiates these abstractions to prevent capacity maps, catalogs, and diagrams from becoming disconnected inventories. It also links strategic maps and indicators with causal hypotheses verifiable through real operating data. Its value to CEOs, CIOs, and chief strategy officers is to provide a concrete bridge between corporate priorities and decisions about investment, standardization, integration, automation, and performance.*

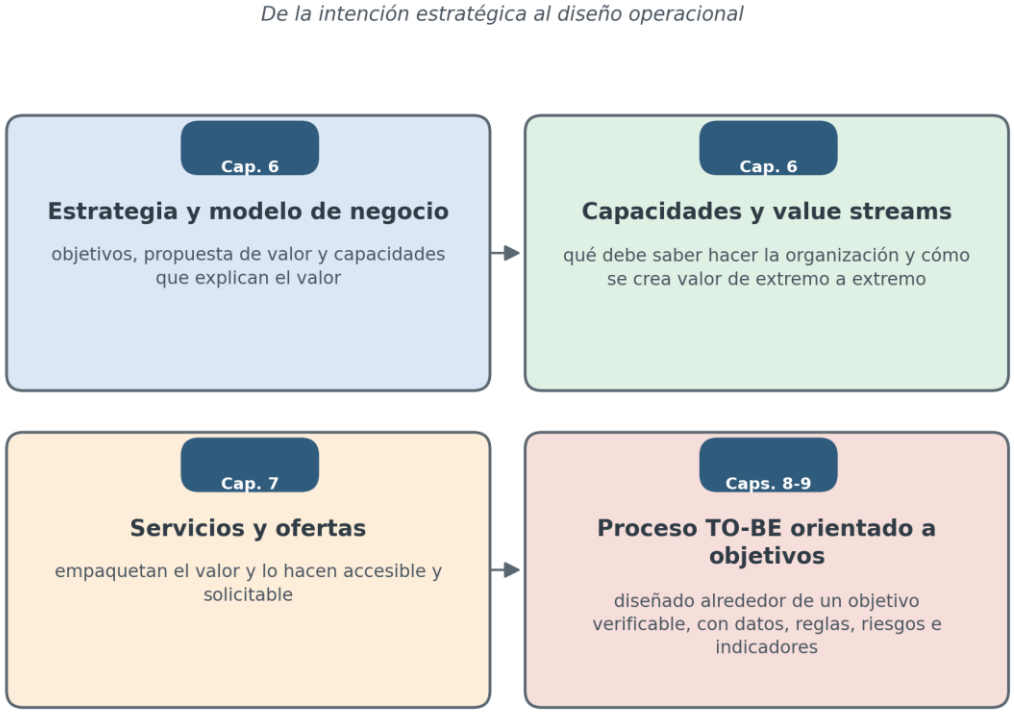


*Figure 9. From strategy to operational design (chapters 6 to 9): capabilities and value streams explain value, services and offerings make it accessible, and the TO-BE process is designed around verifiable objectives, with data, rules, risk.*


All Rights Reserved – Dexon Software-2026

## 6.1 Strategy as a governable intention

The strategy defines the organization's intention: what value it wants to create, for whom, through what differentiating proposition and with what priorities. In a process-oriented methodology, strategy cannot remain an abstract statement separate from the operation. It must become an execution architecture that allows for the drawing of objectives, goals, capabilities, services, processes, data and indicators.

At Dexon, this connection is expressed through the enterprise traceability chain. Strategy is modeled as the first link because it provides direction, prioritization criteria, and business sense to all subsequent artifacts. the *Dexon Process Implementation Guide* indicates that, when the scope includes strategy, it is structured as **objectives → goals → KPIs**, along with initiatives, fiscal year and strategic classifiers that can then be reused in the file of each process. This decision is methodologically important: it allows the process not to be an isolated flow, but an operational unit connected to higher objectives.

Strategy, however, should not be understood only as deliberate planning. It must also recognize learning, adaptation, and evidence. Mintzberg helps to understand that strategy manifests itself as both intention and emergent pattern; Porter recalls that strategy requires choosing a position and making resignations; Khadem emphasizes the need to align vision, responsibilities and indicators; and Osterwalder provides tools to translate value proposition and business model into observable components (Mintzberg, 1987; Porter, 1996; Khadem & Khadem, 2017; Osterwalder & Pigneur, 2010). This methodology takes an integrative view: the strategy is formulated, deployed, executed, measured, and adjusted based on operational evidence.

## 6.2 Business model and value proposition

The business model describes how the organization creates, delivers, and captures value. In this methodology, the business model is the starting point and three pieces are adopted from it – portfolio of business services, business structure and customer routes – to which is added the ontological model, which is the method's own contribution (section 3.2). This combination is particularly suitable for Dexon because it connects four fundamental questions: what value does the organization offer, by what paths its customers receive it, how it organizes itself to produce it, and what business concepts should be shared by processes, data, and indicators.

Osterwalder provides useful language here: the Business Model Canvas allows describing how an organization creates, delivers, and captures value, and the Value Proposition Canvas allows designing, testing, building, and managing products and services integrated with the business model (Osterwalder et al., 2014). At Dexon, this means that the service portfolio should not be an administrative list of requests, but the operational expression of the value proposition. Each service must make sense in the face of a stakeholder, a need, a capacity and an expected result.

The value proposition must be expressed in sufficiently concrete terms to guide the design of processes. It is not enough to say that a service "attends to requests" or "manages requirements".


All Rights Reserved – Dexon Software-2026

It must be indicated what result the stakeholder expects, what condition of success allows the value delivered to be considered and what indicators will allow it to be verified. This clarity is the basis of goal-oriented TO-BE design.

## 6.3 Capabilities as a bridge between intent and execution

Capabilities are the bridge between strategy and operation. A capability describes what the organization must know how to do to fulfill its intention, without yet describing how it does it, by whom, or with what system. In this methodology, the capabilities model is defined as the backbone of the business architecture: it describes what the organization must know how to do to implement its intention, and classifies capabilities as missional, strategic, and operational (Villalobos, 2019, n.d.-b).

TOGAF reinforces this vision by presenting capability-based planning as a business planning technique focused on business outcomes, useful for coordinating efforts across functional domains and achieving strategic capabilities. It also states that architectures should be expressed in terms of outcomes and business value, not just as technology deliverables (The Open Group, 2022a, 2022b). Therefore, in this methodology, a capability is not modeled to complete a map; It is modeled to make decisions about investment, redesign, automation, data, governance, and improvement.

A capability should have a clear name, stable scope, consistent granularity level, type, criticality, and explicit relationships with services, processes, roles, data, applications, and CIs. the *Dexon Process Implementation Guide* sets out practical rules for capability mapping: packages are named with nouns, capabilities with action verbs, type, and criticality are marked only on sheets, and each capability must be linked to services, processes, and roles. These operating rules must be preserved, because they protect the architectural quality of the model.

## 6.4 Value streams and value creation

Value streams allow you to observe how the organization creates value from the perspective of the stakeholder. TOGAF defines the business architecture value stream as an end-to-end collection of activities that add value and produce an outcome for a customer, stakeholder, or end user; in addition, it points out that value must be defined from the perspective of the stakeholder who receives the product, service, or result (The Open Group, 2022c). This idea is essential to prevent processes from being designed solely from the internal structure of the organization.

The Dexon methodology must clearly distinguish value stream, capacity and process. The value stream shows how value is produced from end to end. Capability describes what the organization must know how to do to produce it. The BPMN process describes how work is executed, coordinated, controlled, and automated. TOGAF points out that capabilities, value streams, processes, information, and systems are distinct concepts within business architecture, and that the process operationalizes the capability and translates the value stream to the operational level (The Open Group, 2022a).


All Rights Reserved – Dexon Software-2026

This distinction changes the quality of the TO-BE design. The architect does not first ask "what activities does the area do today?", but "what value should the stakeholder receive, what stages generate that value, what capabilities enable each stage and what processes must exist to execute them?". Thus, the process is not born from an organizational chart or an operational custom, but from a logic of value.

## 6.5 Relationship between capabilities, services and processes

A capability can be implemented by multiple processes, and a process can orchestrate multiple capabilities. This methodology establishes that capacities are stable and timeless, while processes are dynamic and sequential, and that the map of capacities is the structural basis and processes are the way in which that structure comes to life. This relationship must become a methodological rule: no process is approved if it does not declare which capacities it orchestrates.

Services appear as the unit of value that allows the materialization of capabilities to stakeholders. A service is not a capability or a process. Ability expresses a skill; service expresses a promise of value; the offer expresses a form of access; The process expresses the mechanism of execution. At Dexon, this separation is native: the service catalog records owner, value proposition, criticality, cost/price, capabilities that implement it, and CIs that support it.

The right relationship is as follows: the strategy defines priorities; capabilities express what the organization must know how to do; value streams show how value is created; services package that value; the offers make it requestable; and BPMN processes execute the required work. This logic avoids two common mistakes: designing processes without explicit value and creating services without real delivery capacity.

## 6.6 Strategy Maps and Verification of Value Assumptions

The Strategy → Indicators link of the Dexon chain has a direct antecedent in the Balanced Scorecard and strategy maps. Kaplan and Norton demonstrated that a strategy is only executable when it is translated into concrete objectives, connected by explicit cause-and-effect relationships, and accompanied by indicators and goals (Kaplan & Norton, 1996, 2004). A strategy map is, at its core, a set of hypotheses: if the organization strengthens certain internal capabilities and processes, then it will deliver a certain value proposition and obtain certain results.

The structure, objectives → goals → KPIs that the *Dexon Process Implementation Guide* defines for the strategic stage is a deployment of this logic. The methodology extends it at a critical point: between the strategic objective and the indicator, the Dexon chain interposes capabilities, services, offers and executable processes. This turns the hypotheses of the strategy map into verifiable hypotheses: as the processes are executed on the platform and generate real operational data, the organization can contrast whether the improvement of a process effectively moved the strategic indicator to which it is traced. Alignment ceases to be an annual planning exercise and becomes a permanent circuit of formulation, execution, measurement and adjustment: the synthesis between deliberate and emergent strategy argued in chapter 3 (Mintzberg & Waters, 1985; Salazar Yusti,


All Rights Reserved – Dexon Software-2026

2021).

# Chapter 7. Services and offers as a materialization of value

*The chapter places the service as a unit of business value and the offer as the concrete mechanism through which this value is made accessible and requestable. A service must express what result it delivers, to whom, under what conditions, through what capacities and with what operational support; Therefore, the catalog ceases to be an inventory of procedures and becomes an operational view of the business model. The methodology distinguishes capacity, service, supply and process to avoid duplication and design errors: capacity is an organizational skill, service is a promise of value, supply is a specific form of consumption and process is the executable mechanism that produces the result. In Dexon BPM, this separation allows you to reuse processes for various offerings and connect catalog, user experience, data, rules, workflow, and indicators. Executive engagement is relevant: an organization does not transform its relationship with customers or users by publishing forms, but by ensuring that each offering activates a complete, consistent, measurable, and governed delivery architecture.*

## 7.1 Service as a unit of business value

A service is a unit of value offered by the organization to an internal or external stakeholder. Its function is to convert organizational capabilities into useful results for someone. Therefore, a service must have a clear value proposition, an owner, a criticality, a relationship with capabilities and a relationship with the resources or ICs that support it.

This definition is convergent with ITIL 4's service theory, which defines a service as the means to enable the co-creation of value by facilitating the outcomes that consumers want to achieve, without consumers having to manage specific costs and risks, and which distinguishes service from service *offering*, understood as the formal description of one or more services designed to meet the needs of a target group of consumers (Axelos, 2019). Dexon's separation between service – a promise of value with an owner and capabilities – and an offer – a concrete and requestable form of consumption – implements exactly this distinction, and allows the catalogue to function simultaneously as an instrument of architecture and as a consumer experience.

At Dexon, the service catalog should not be understood as an inventory of procedures. It should be understood as an operational view of the business model. the *Dexon Process Implementation Guide* establishes that for each service, identity, status, visibility, description, owner, value proposition, tier, criticality, cost/price, capabilities that implement it and CIs that support it are documented. This structure is sufficient to treat the service as an architectural artifact, not just a portal element.

A well-defined service answers five questions: what value does it deliver, to whom, under what conditions, through what capabilities, and with what operational support. If any of these questions cannot be answered, the service is not yet mature to conduct a TO-BE process.


All Rights Reserved – Dexon Software-2026

## 7.2 Difference Between Capacity, Service, Offering, and Process

The Dexon methodology adopts the following definitions:

- **Capacity.** The organization's stable ability to do something necessary to fulfill its intention. It is relatively independent of specific organizational charts, systems, and processes.
- **Service.** Unit of value that the organization offers to a stakeholder, supported by capabilities and resources.
- **Offer.** The specific way in which a service is presented, requested or consumed. It is the visible face of the service for the user.
- **Process.** BPMN coordination of activities, decisions, events, roles, data, and rules that executes the service and produces a verifiable result.
- These definitions avoid methodological confusion. "Managing access requests" can be a process; "Identity and access management" can be a capability; "Access support" can be a service; "Request access to an app" can be an offer. Architecture becomes governable when each element takes its place.

*Cuatro preguntas distintas sobre el mismo dominio: cada elemento ocupa su lugar*

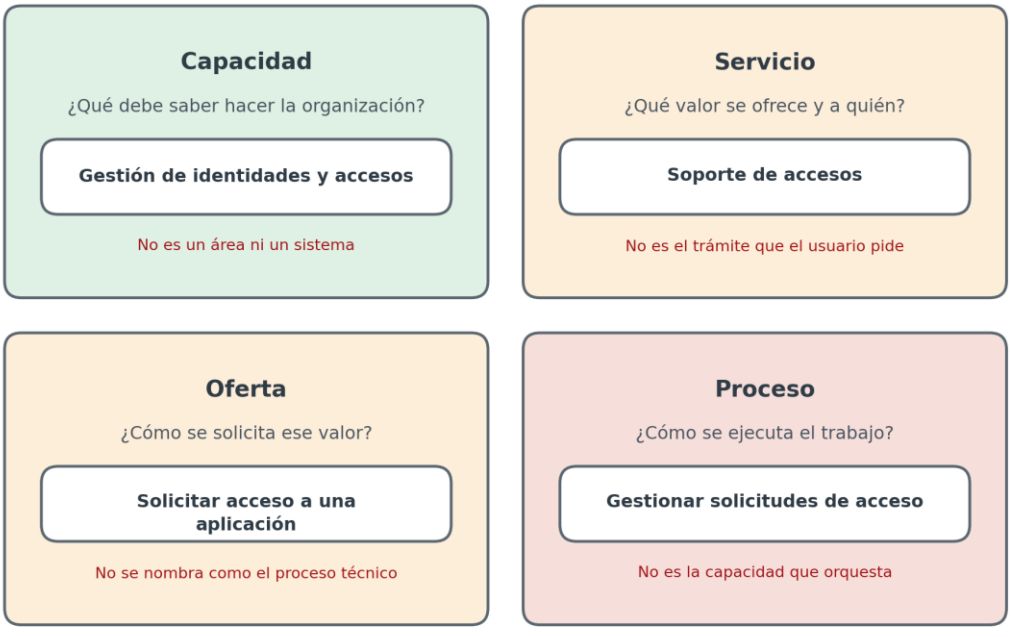


*Figure 10. The four concepts applied to the same domain, with the most frequent classification error at each level.*

## 7.3 Supply as a mechanism for accessing value

The service offering turns value into something accessible. At Dexon, the offer is explicitly the commercial face of the catalog: it has its own name and description, the process that executes it,


All Rights Reserved – Dexon Software-2026

order, visibility and validity; Several offers can share the same process. This separation allows the end user to see a clear and understandable promise, while the organization maintains reusable and governed processes.

The offer should not be called like the technical process. It should be named as the request, need, or result that the user recognizes. This distinction improves user experience and architecture. It allows the same capacity and the same process to support multiple forms of consumption, without duplicating flows unnecessarily.

The operational guide also establishes that an offer is requestable when it is visible, in force and the process has an initial layout; If it's gray, the end user won't see it or won't be able to order it. In conceptual methodology, this should be interpreted as an architectural condition: value only materializes when the offer, process, data, and user experience are integrated.

## 7.4 Relationship with Dexon BPM

Dexon BPM supports this architecture through integrated modules: it is a solution for optimizing complex processes and managing large volumes of data, consisting of Dexon Designer, Dexon Workflow, Dexon ETL, Dexon ECM and Dexon BI. Dexon Workflow allows you to design and implement business processes, model activities and data needed for their execution, and integrate existing systems.

Therefore, the service catalog and offers are not simple forms. They are points of coupling between architecture, experience, processes, data and indicators. At Dexon, the process executes the offer; the data model structures the capture and query; the rules engine governs decisions; and the reports/KPIs show if the value was delivered. This integration between catalog, offer, and workflow is also the central requirement of the service catalog and request management practices under which Dexon BPM was certified (Pink Elephant, 2026; see chapter 22).

# Chapter 8. Objective-Driven Process Design

*This chapter converts Objective-Driven Process Design into a formal design contract. The objective of the process ceases to be a descriptive text and is specified as an architectural object composed of purpose, stakeholders, results, satisfaction criteria, thresholds, constraints, risks, capacities and evidence. This structure makes it possible to determine whether a target is well-formed and, subsequently, to evaluate each trajectory of the TO-BE against explicit conditions. The methodology distinguishes strong satisfaction, weak satisfaction and dissatisfaction: only the first allows the design to be approved, because it requires that all normal trajectories achieve results and that exceptional ones be governed, preserve restrictions and leave evidence. The chapter also introduces rules to break down complex objectives and make visible the conflicts between criteria such as opportunity, control, cost, experience or compliance. For management, the contribution is to turn ambiguous discussions about "the ideal process" into verifiable decisions: what value should be achieved, what tensions are accepted, what thresholds express the balance and what evidence will*


All Rights Reserved – Dexon Software-2026

*allow the design to be confirmed or refuted.*

## 8.1 Formal definition

Formally, **Objective-Driven Process Design** is defined as the methodological approach by which a TO-BE process is designed, evaluated, automated, measured, and governed based on an explicit, verifiable, and traceable objective with strategy, capabilities, services, data, and indicators. While Chapter 5 introduced the motivation of the approach, this chapter specifies it as a design contract: the structure of the objective, its mandatory components, and the conditions under which a TO-BE can be considered complete.

This approach maintains BPMN as the modeling language. OMG defines BPMN as a graphical notation for specifying business processes, understandable for business users and sufficiently accurate for technical users, creating a standardized bridge between design and implementation (OMG, 2013). The Dexon methodology retains that bridge, but adds a design rule: the process is not modeled to represent activities; it is modeled to satisfy a goal.

GoalBPM offers the closest conceptual foundation: it proposes to relate process models in BPMN to stakeholder objectives, and argues that objectives must be explicitly stated, communicated, and traceable to proposed, approved, or implemented changes in processes (Koliadis & Ghose, 2006). This methodology adopts that principle and makes it Dexon practice: each TO-BE process must demonstrate how it satisfies its objective through BPMN trajectories, data, rules, indicators, and satisfaction criteria.

## 8.2 Objective of the process

The goal of the process is to specify the result that the process should produce. It is not a decorative phrase or a description of activities. It is the criterion that allows us to decide which activities exist, which decisions are automated, which data are mandatory, which exceptions are governed and which indicators are measured.

A robust goal should include:

- **Beneficiary stakeholder:** who receives or evaluates the value.
- **Expected result:** what should be obtained at the end of the process.
- **Success criterion:** how the result is recognized as satisfactory.
- **Restrictions: conditions** of time, cost, compliance, risk, quality or experience.
- **Evidence:** data and indicators that allow the achievement to be verified.

Sections 8.6 and 8.7 elevate this structure to formal specification: mandatory components, rules of good formation, and satisfaction semantics.

Example:
Manage supplier incorporation requests so that the requesting area has an authorized, validated,


All Rights Reserved – Dexon Software-2026

and registered supplier in a timely manner, complying with risk policies, documentary requirements, and contracting rules, and leaving traceable evidence for audit and measurement.

## 8.3 Expected Outcomes, Stakeholders and Satisfaction Criteria

The expected result is the business state that should be achieved when the process ends successfully. It should not be confused with a documentary output. An output can be "approved form"; A result is "supplier authorized to contract under verified conditions". The result expresses value; the output expresses evidence or artifact.

The stakeholder is the one who initiates, receives, participates in, or evaluates the value. TOGAF Value Streams emphasizes that value is defined from the perspective of the stakeholder, not from the internal cost of producing it. Therefore, the TO-BE process must specify who receives the value and what condition allows it to recognize it as useful.

Satisfaction criteria turn the objective into assessable conditions. They must answer: what must be true at the end of the process?, what cannot happen?, what exceptions are acceptable?, what data proves compliance?, what indicator will show performance? Without satisfaction criteria, the objective cannot govern the design.

The satisfaction criteria also require a post-closing verification instrument: if the value is defined by the stakeholder, the design must provide for the mechanism by which that stakeholder confirms – or denies – that the result received was as expected. Satisfaction measurement is not a courtesy of service; It is the final link in the chain objective → evidence, and its results directly feed into the improvement cycle of Chapter 20.

## 8.4 Restrictions, Risks, and Decisions

Every process operates under constraints. Some come from strategy; others related to regulation, internal policy, security, budget, operational capacity, user experience, technological availability or data quality. The TO-BE design must make them explicit before modeling the flow, because constraints determine gates, controls, validations, rules, SLAs, escalation, and exceptions.

Risks should be treated as design conditions. A risk isn't just something that can fail; it is a possibility that requires architectural decision. If the risk is fraud, the process should include verification, segregation of duties, or automated rules. If the risk is SLA non-compliance, the process should incorporate monitoring, alarms, or escalation paths. If the risk is poor data quality, the process should include validation, enrichment, or integration.

In Dexon, this logic is supported by the rules engine, layouts, Related Data, SLAs, notifications, reports and CIs. The operational guidance states that the BPMN design is implemented through statuses/activities, transitions, and rules, and that each activity must define form, related data, assignment, and notifications.


All Rights Reserved – Dexon Software-2026

### Risk Register as a Governed Object

Treating risk as a design condition requires that risk be an object and not an annotation. Each log entry goes through a full cycle, and the three stages must be declared separately, because confusing them is what produces logs that no one uses:

**Inherent risk** —type, impact, and likelihood prior to any treatment—→ **treatment** —type of mitigation (avoid, reduce, transfer, or accept), countermeasure, and management summary—→ **residual risk** —impact and likelihood after treatment. A record that only keeps one rating does not allow us to know if the current level is the result of a decision or luck.

### Two elements connect the register to the rest of the frame.

Early **warning signs** are what is observed *before* materialization. They are the natural hotbed of KRIs: every signal that can be quantified is a candidate for a risk-linked measurement element (section 15.8). A risk register with declared signals and without indicators to measure them is describing, without saying so, the work that is missing.

The **uses of risk** are the processes whose design it conditions, the portfolio elements analyzed and the KRIs that anticipate it. From this follows a rule of integrity that should be stated as such: **a registry entry that is in use is not deleted.** First, their uses are disassociated, leaving a trace of the decision. A record from which referenced risks are erased is a record that no one can trust, and the impact analysis in section 19.3 is no longer reliable with it.

## 8.5 KPIs derived from the objective

Indicators should be derived from the objective, not added at the end. The *Dexon Process Implementation Guide* states that KPIs should have goal, meaning, frequency, and ownership, and be linked to the actual process and strategic objectives if the strategic stage was performed. In Objective-Driven Process Design, this rule is expanded: each KPI must have an explicit relationship with outcome, quality, timeliness, compliance, risk, or experience. Chapter 15 formalizes this derivation using the Goal-Question-Metric approach (Basili et al., 1994).

It is recommended to distinguish four types of indicators:

1. **Achievement indicators:** measure whether the expected result was achieved.
2. **Flow indicators:** measure time, backlog, volume, cycle, and capacity.
3. **Quality indicators:** measure rework, errors, incomplete data or satisfaction.
4. **Control indicators:** measure compliance, risk, exceptions and audit.

A process that only measures volume or time may seem efficient and still fail to meet its goal. A goal-oriented process measures value achieved.

## 8.6 Formal specification of the objective of the process

According to the metamodel in Chapter 4, the objective of the process is not a text: it is an


All Rights Reserved – Dexon Software-2026

architectural object with obligatory components and typed relationships. Each of its nine components corresponds to a type stated in section 4.3 — purpose as an attribute of the objective itself; stakeholders as roles; results, criteria, thresholds, and constraints as objects of specification; risks, capacities, and evidence as structural objects—and to a relationship with cardinality declared in section 4.4. This correspondence is what allows good training to be verified by consultation and not by reading. Formally, a process objective is specified as the tuple:

**O = ⟨ purpose, stakeholders, outcomes, criteria, thresholds, constraints, risks, capabilities, evidence, ⟩**

*Figure 11. The formal specification of the objective: nine mandatory components around a single architectural object – intent (blue), verification (water green), conditions (red) and traceability (green).*

| Component | Definition | Good Formation Rule |
|---|---|---|
| Purpose | Why the process exists in terms of value | Unique, brief, traceable to a service or strategic objective |
| Stakeholders | Who receives or evaluates value | At least one, identified in the organizational model or typified if external |


All Rights Reserved – Dexon Software-2026

| Component | Definition | Good Formation Rule |
|---|---|---|
| Expected results | Business states to be reached upon completion | At least one, expressed on ontological model entities, not on interface artifacts |
| Satisfaction criteria | Evaluable predicates that make each result verifiable | Each result has at least one criterion; Each criterion references attributes of entities with identified data source |
| Thresholds | Quantification of tolerance by criterion | Each quantifiable criterion declares goal, tolerance band, and sense (maximize, minimize, or range) |
| Restrictions | Non-negotiable design conditions | Typified: time, cost, compliance, quality, risk or experience |
| Risks | Possibilities that degrade the result | Every critical risk has a visible design treatment in the TO-BE |
| Required Skills | Skills that the process orchestrates | At least one, drawn to the map of capabilities |
| Evidence | Records Demonstrating Compliance | Each criterion has at least one piece of evidence or an indicator that feeds it |

The nine-component tuple and its well-formed rules are a **proposal of this methodology**, built on the basis of goal-oriented requirements engineering (van Lamsweerde, 2009), the specification of architectural constraints, risk management and the measurement information model in Chapter 15. It does not come as such from any of these separate sources.

A goal is **well-formed** when it satisfies all the rules in the third column. The practical consequence is the same as that pursued by the entire metamodel: good training is verifiable by query—it can be checked against the repository, field by field, and relationship by relationship—and is therefore a candidate for government by design in Chapter 16. A process whose objective is not well formed should not enter into the design of trajectories, because no trajectory can be demonstrated satisfactorily against criteria that do not exist or are not evaluable.

## 8.7 Semantics of Satisfaction, Decomposition, and Conflict

The formal specification allows you to define precisely what it means that a process satisfies its objective. Based on the strong, weak and unsatisfied classification proposed by GoalBPM (Koliadis & Ghose, 2006), the Dexon methodology **redefines its semantics** to distinguish a governed exception from a trajectory that does not meet the objective: while the original approach assesses satisfaction on the set of trajectories without attributing a status of its own to the expected exceptions, this methodology admits strong satisfaction in the presence of exceptional trajectories as long as they are governed, they preserve critical constraints and leave evidence. The classification that follows is


All Rights Reserved – Dexon Software-2026

therefore a Dexon extension built on the conceptual foundation of GoalBPM, not a literal application of it. Let T be the set of trajectories of the TO-BE model – the possible paths from a beginning to an end – divided into normal and exceptional. The objective is stated with **strong satisfaction** when every normal trajectory achieves at least one expected result by meeting all criteria and thresholds, and every exceptional trajectory ends in a governed exception that preserves constraints and leaves evidence. It is declared with **weak satisfaction** when normal trajectories achieve results but violate non-critical thresholds, or when there are exceptions that are not yet formally governed. It is declared **dissatisfied** when some normal trajectory does not lead to any expected result, or when some criterion lacks data to be evaluated. The approval criteria of the TO-BE (sections 5.5 and 9.7) thus acquires an exact formulation: only a design is approved with strong satisfaction; weak satisfaction forces us to redesign or renegotiate thresholds explicitly; Dissatisfaction returns design to formulation.

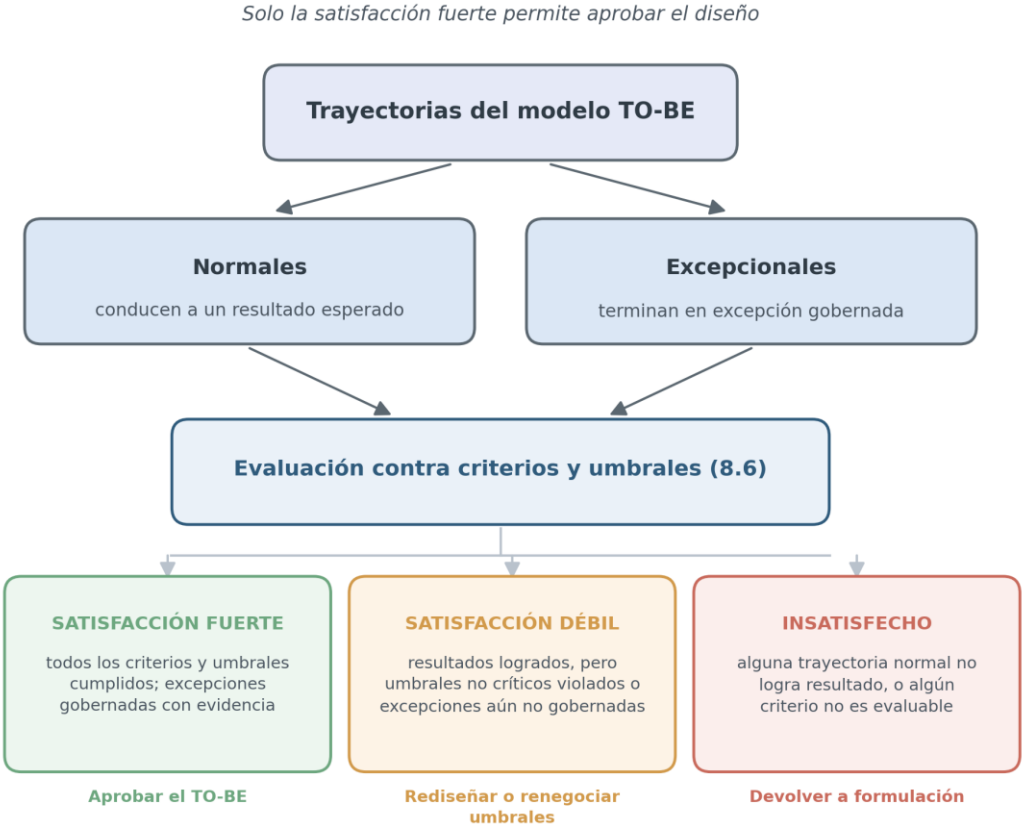


*Figure 12. Satisfaction semantics: Every TO-BE trajectory is evaluated against the criteria and thresholds of the objective, and only strong satisfaction approves the design.*

Complex objectives break down, and goal-oriented requirements engineering provides the discipline to do so (van Lamsweerde, 2009). In an AND decomposition, the parent goal is satisfied only if all subgoals are satisfied; in an OR decomposition, the sub-objectives are design alternatives and it is enough to satisfy one, leaving the choice and its justification recorded. The decomposition ends


All Rights Reserved – Dexon Software-2026

when each objective is assignable —there is a trajectory, activity or rule responsible for satisfying it— and measurable —there is a criterion with a threshold and source—. This stopping condition avoids both abstract goals that no one executes and pulverization on tasks with no sense of value.

Finally, objectives can conflict: opportunity versus control, cost versus expertise, speed versus compliance. The same formal tradition teaches that conflicts are not resolved by hiding them in the design, but by declaring them: when two criteria are stressful, the objective must register the arbitration—which criterion prevails, or which pair of thresholds expresses the accepted balance—so that the decision is governable and revisable. An arbitrated conflict is an architectural decision; A hidden conflict is an informal exception waiting to occur.

# Chapter 9. Goal-oriented TO-BE design methodology

*The chapter operationalizes the approach in seven phases that subordinate the BPM cycle to the objective of the process. First, the intention is formulated and the process is located in the traceability chain; then objective, results, criteria, thresholds, risks and evidence are defined. The AS-IS is analyzed in selective depth, only to identify what prevents the goal from being achieved, what needs to be conserved, and what needs to be transformed. From there, normal and exceptional TO-BE trajectories are designed, followed by data, rules, forms, integrations, CIs, SLAs, and automation. Before passing, the satisfaction of the objective in real scenarios is validated and the execution, measurement and governance are prepared. The method avoids two extremes: exhaustively documenting the past without the ability to redesign, or automating a future vision without sufficient data or controls. For an executive sponsor, the seven phases constitute a value assurance scheme that allows for clear deliverables and step criteria to be required before committing the output to production.*

## 9.1 Methodological principle

The TO-BE design is not about "improving the AS-IS". It starts by formulating the objective of the process and designing the most appropriate mechanism to satisfy it. The AS-IS is studied to the extent that it allows us to understand restrictions, existing data, risks, actors, exceptions and opportunities; but it should not limit the imagination of the TO-BE.

The difference is profound: traditional TO-BE tends to optimize existing activities; goal-oriented TO-BE designs satisfaction trajectories. GoalBPM refers to the possible paths of a process and assesses whether they meet normative objectives, classifying strong, weak, or dissatisfied satisfaction (Koliadis & Ghose, 2006). This methodology adopts this classification and extends its semantics in the terms specified in section 8.7.

The seven phases that follow are consistent with the BPM lifecycle consolidated in the literature—process identification, discovery, analysis, redesign, implementation, and monitoring/control (Dumas et al., 2018)—but introduce a correction of emphasis: the cycle is not traveled around the current process, but around the goal. Identification and discovery are subordinated to the formulation of the intention (phases 1 and 2); the AS-IS analysis is instrumental and limited (phase


All Rights Reserved – Dexon Software-2026

3); the redesign is design of satisfaction trajectories (phase 4); and implementation and monitoring are conditional on the validation of the objective (phases 5 to 7). This subordination of the life cycle to the objective is the central operational difference between a conventional BPM program and Objective-Driven Process Design.

*Figure 13. The seven phases of goal-oriented TO-BE design and the artifact that each one engages.*

## 9.2 Phase 1 — Formulating the Intent of the Process

The first phase consists of locating the process within the traceability chain. The team should identify the related strategy, core capability, value stream, service, offering, and stakeholder. The result of this phase is an intention sheet that answers: why the process exists, what value it delivers, and what capacity it operationalizes.

At Dexon, this phase is connected to the process file, which must be filled out before the BPMN and include identity, objective, classification, responsibility, relationships with value chain and service, and SIPOC inputs/outputs.


All Rights Reserved – Dexon Software-2026

## 9.3 Phase 2 — Define objective, results and satisfaction criteria

The second phase turns intent into verifiable goal. The objective is broken down into expected results, sub-objectives, constraints, risks, required data and preliminary KPIs. Here you define what "success" means before drawing activities.

Minimal Artifacts:

1. Process objective sheet.
2. Target–Outcome–KPI Matrix.
3. List of restrictions and risks.
4. Stakeholder map.
5. Satisfaction criteria.

## 9.4 Phase 3 — Analyzing AS-IS with Purpose

The AS-IS is modeled only with the depth needed to redesign. It is not intended to document everything; The aim is to understand why the current process does not fully satisfy the objective. The analysis should identify delays, reprocesses, dataless decisions, redundant controls, missing data, informal exceptions, diffuse liabilities, and non-value-contributing activities.

The AS-IS must answer three questions: what prevents achieving the objective, what should be retained, and what should be eliminated or transformed.

## 9.5 Phase 4 — Design TO-BE trajectories

The fourth phase designs the future process in BPMN. Every normal trajectory should lead to an expected outcome. Every exceptional trajectory must be governed: resolve, compensate, escalate, reject, or close with evidence. No critical exception should be left out of the model.

The operational guidance summarizes good BPMN practices: clear start, explicit purposes, gates with named conditions, legible happy path, and absence of unnecessary crossings. Objective-Driven Process Design adds a rule: each BPMN element must justify its contribution to the objective, constraint, data, rule, risk or indicator.

## 9.6 Phase 5 — Designing Data, Rules, and Automation

The TO-BE is not complete if it only has flow. You must specify data, rules, forms, integrations, mappings, and events. the *Dexon Process Implementation Guide* establishes that the data is connected to the processes through Related Data, associating tables that each category of activity captures or queries, with mandatory status filters, scope filters and on-site editing.

In this phase, the following are defined:

1. Mandatory data per activity.
2. Decision rules.


All Rights Reserved – Dexon Software-2026

3. Forms and layouts.
4. Validations and calculations.
5. ETL or API integrations.
6. ICs involved.
7. Documentary evidence.
8. Alarms, notifications and SLAs.

The characterization of each activity—**type of task, responsible organizational unit taken from the actual structure and not as free text, expected effort, and deadline**—is part of the persisted design, with the same rank as the diagram. It survives the reload, the administrator's report is consulted and feeded. An activity characterized in margin notes is not characterized: these four attributes are the ones that later allow the load to be dimensioned, slack to be calculated and a deadline missed without reconstructing it from memory.

## 9.7 Phase 6 — Validate Goal Satisfaction

Before approving the TO-BE, the team must validate that the process meets its goal. Validation must cover normal, alternate, and exceptional scenarios. For each scenario, the following are verified: expected result, available data, rule applied, assignee assigned, evidence generated and indicator fed. When the constraint is time-bound – SLAs, regulatory deadlines, windows of attention – validation should include simulation against actual or observed behavior: contrasting the designed thresholds with the effective distribution of cases before triggering them, rather than adjusting them after the first breach. Ex ante simulation is to time constraints what E2E testing is to trajectories.

A TO-BE is approved when:

1. It has a verifiable objective.
2. It is traced to capacity and service, and to the offer if it is requested.
3. Their normal trajectories meet the target.
4. Its exceptions are governed.
5. Its data and rules are sufficient.
6. Its KPIs allow you to measure achievement and control.
7. Its automation at Dexon is viable.

When the process has segments that run as a case family (section 12.4), the condition of closed paths is extended: in addition to every path ending in an expected outcome or a governed exception, the R15 contract must be fulfilled—every activity that outputs results offers them to the closer—and the completion conditions of each join must be declared. A family without a contract does not fail to be approved: it fails to be executed, and it does so in silence, standing still.

## 9.8 Phase 7 — Prepare for Implementation, Measurement, and Governance

The last phase prepares the passage from design to execution. At Dexon, this implies that the process


All Rights Reserved – Dexon Software-2026

has an initial layout, that the offer is requestable, that the data is hung up on the process, that the SLAs are simulated, that the KPIs are linked to the real process and that the E2E test goes from request to report.

The TO-BE does not end when it is published. It ends when you can govern. To govern means to version, measure, revise, correct, reapprove and learn. A goal-oriented process must produce enough evidence to know if the organization is creating the promised value.

# Chapter 10. Processes and data as an integrated system

*This chapter argues that process and data are seen as inseparable from the same operational reality. The process transforms inputs into results through activities, decisions and rules; Data represents the actors, objects, events, states, and transactions that allow that transformation to be executed, controlled, integrated, and measured. For this reason, the TO-BE design cannot consist of first drawing the flow and then adding fields or forms: each activity must indicate what data it consumes and generates, which rule it applies, what evidence it leaves and what indicator it feeds. The ontological model precedes and governs the data model by stabilizing the business vocabulary, a critical requirement for organizations in which different areas employ incompatible concepts. Finally, data is treated as assets, with explicit purpose and governance responsibilities. For senior management, this integration reduces the risk of technically correct but business-inconsistent automations, and creates a common foundation for interoperability, analytics, compliance, auditing, and consistent decisions at the corporate scale.*


All Rights Reserved – Dexon Software-2026

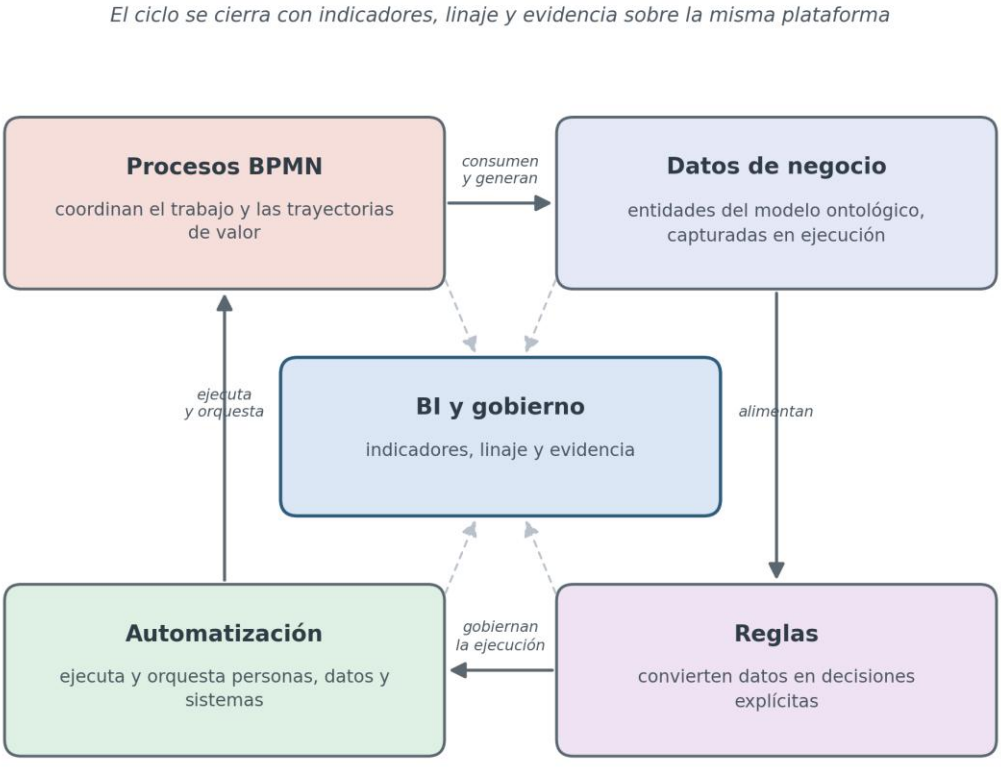


*Figure 14. The integrated process-data system (chapters 10 to 15): BPMN processes consume and generate data, rules convert data into decisions, automation executes and orchestrates, and BI/government close the loop with indicators, lineage, and evidence.*

## 10.1 The structural unity between process, data and result

In this methodology, a process is not defined solely as a sequence of activities, nor is the data defined as a record stored in a table. The process is the mechanism by which the organization transforms inputs into results; Data is the structured representation of the facts, objects, states and decisions that make this transformation possible. Therefore, process and data are two complementary views of the same operational reality.

A goal-oriented process requires data to initiate, classify, decide, assign, control, audit, and measure. In turn, data acquires organizational value when it is captured, validated, transformed, and used within real processes. Therefore, this methodology describes data as the "common language of business" and as a mechanism for measuring, analyzing, personalizing, automating, and complying with regulations.

Therefore, the TO-BE design should not first produce a BPMN and then "add fields". It must produce an integrated model where each activity indicates what information it requires, what information it


All Rights Reserved – Dexon Software-2026

generates, what rule it applies, what evidence it leaves and what indicator it feeds. A flow that doesn't define your data is an incomplete flow; A data model that does not identify the processes that use it is an abstraction without operation.

## 10.2 Ontological model and semantic coherence

The ontological model defines the vocabulary of the business: actors, objects, events, transactions, contracts, services, results and indicators. Their role is to ensure that processes, data, capabilities and indicators speak the same language. The ontological model is a contribution of this methodology and is not part of the decomposition of the business model from the reference framework (section 3.2). It is defined as the set of concepts that the organization handles and the relationships between them.

This semantic coherence is critical in large organizations. When one area calls the buyer "customer", another calls the consumer "user" and another calls the contractor "third party", automation can work technically and still fail business-wise. The Dexon methodology requires that relevant concepts be stabilized before they are converted into entities, forms, rules, reports, or integrations.

The ontological model does not replace the data model. It precedes and governs it. Define what concepts exist in the business domain; The data model defines how those concepts are structured into entities, attributes, relationships, domains, keys, and referential integrity. TOGAF reinforces this orientation by pointing out that data architecture must define the types and sources of data necessary to support the business in a way that is understandable to stakeholders (The Open Group, 2022a).

This requirement has an operational consequence: semantic control is exercised at the time of creation, not in a subsequent revision. Creating an entity must first involve verifying whether the concept already exists in the model (reuse precedes creation), and the structural quality of the model (concepts without definition, isolated entities, broken references, missing domains) must be treated as a measurable condition for progress, not as a documentary aspiration. A vocabulary is stabilized by protecting its entry point, and the operational guide implements both controls directly on the platform.

## 10.3 Data as Business Assets

Data is a business asset because it allows you to operate, decide, control, measure and learn. DAMA-DMBOK organizes data management around areas such as governance, architecture, modeling, integration, interoperability, master data, metadata, BI, and data quality, and locates governance as a discipline that establishes policies, roles, and standards for managing data as valuable assets (DAMA International, 2017).

At Dexon, this notion must be translated into a methodological rule: all data captured in a process must have a purpose. It can be captured because it is necessary to execute an activity, apply a rule, make a decision, fulfill an obligation, leave evidence, feed a KPI, integrate another system or


All Rights Reserved – Dexon Software-2026

preserve traceability. If a piece of data does not fulfill any of these functions, it must be questioned.

The practical consequence is that the form is not a capture screen; It's a data governance point. Each field in a layout must correspond to an entity, attribute, rule, or evidence. Each mandatory value must be justified by operational, regulatory, analytical or control necessity.

# Chapter 11. Information architecture and business data model

*The chapter translates the vocabulary of the business into information structures that can be used by processes. Entities represent objects, actors, events, or transactions that the organization needs to know about and govern, and must be discovered from the ontological model with business user validation. A mature entity is not limited to a table: it requires definition, conceptual responsibility, attributes, rules of uniqueness, relationships, domains, quality criteria, sensitivity, volatility, historicization, and processes that consume or generate it. The chapter highlights that cardinalities, obligatoriness and referential integrity are business conditions of coherence, while business rules must be explicit, legible and governable. Dexon Designer materializes these definitions and Related Data connects entities to every activity at runtime, including capture, query, enforceability, and scope. For CIOs and data managers, the chapter offers a discipline to prevent forms, integrations, and reports from creating parallel definitions, and to ensure that the executable process shares consistent corporate semantics.*

## 11.1 Business entities

A business entity represents an object, actor, event, or transaction that the organization needs to know, manage, or evaluate. It can be a Customer, Request, Contract, Product, Vendor, Case, Asset, Document, Payment, Incident, or Service. Current methodology defines entities as real-world objects relevant to the business domain and links them to attributes, relationships, forms, reports, and information systems.

Entities must be discovered from the ontological model and validated with business users. The operational guidance explicitly states that, in the data model stage, entities from actors, business objects, events, and transactions that processes are going to read or write must be discovered.

A well-defined entity must have a business name, description, conceptual responsible, main attributes, uniqueness rules, relationships, value domains, quality criteria, level of sensitivity, volatility, required historicization and processes that consume or generate it.

## 11.2 Relationships, Integrity, and Business Rules

Relationships express how entities connect. A Client may submit many Requests; a Request may require multiple Documents; an Asset can be associated with an IC; a Service can be supported in


All Rights Reserved – Dexon Software-2026

multiple ICs; a Process can query multiple entities and update others. Cardinality, obligatoriness and referential integrity are conditions of coherence, not technical details.

Business rules define constraints, calculations, validations, derivations, policies, and decision criteria. OMG DMN defines Decision Model and Notation as a language and notation for specifying business decisions and business rules, designed to work together with BPMN and CMMN in the modeling of decisions associated with processes and cases (OMG, 2024). The Dexon methodology does not require replacing BPMN with DMN, but it does adopt its principle: **decisions must be explicit, readable and governable**.

In Dexon, rules appear in calculations, validations, transitions, assignments, SLAs, integrations, reports, and alerts. The platform's calculation engine (Calculus Engine) is the basis for calculated fields, validation rules, DB Mappings calculations and rules for creating BPM activities.

## 11.3 Relationship with Dexon Designer and Related Data

Dexon Designer and the table designer materialize the data model into structures that can be used by processes. The operational guide defines non-negotiable conventions: implicit PK <TABLE>_ID, FK using <NAME>_ID, hierarchies with PART_OF, lists of values such as *_LST tables, and well-defined single fields to support upserts and bulk loading.

Related Data is the mechanism by which the data model connects to the execution of the process. The tables it captures or queries are associated with each activity category, with mandatory status filters, scope filters, and on-site editing. The guidance expressly states that this connects "data" to "processes" at runtime.

This relationship changes the nature of BPMN at Dexon. Activity isn't just a box in a diagram; It is a point of interaction between human or automatic work, business data, rules, permissions, evidence, notifications, and measurement.

# Chapter 12. Data for decision-making

*This chapter addresses the decision as an explicit object of operational design. Each decision must identify input data, applicable rule, responsible or driver, result, evidence, exceptions and related indicator; otherwise, a BPMN gate represents uncertainty, not government. Automation is reserved for decisions that are structured, repeatable, traceable, and supported by reliable data, while human judgment is maintained when there is contextual ambiguity, negotiation, exception, or professional responsibility. The criterion is not to automate in principle, but to determine whether the decision improves timeliness, consistency, compliance, control, scalability or experience without degrading the quality of the result. The chapter further introduces the three zones of execution of a process—augmented people, algorithmic zone, and agents—with an explicit allocation rule, distinguishes the two ways in which a path is executed—state transitions and case family—and stresses that data quality must be designed from the source through validations, Domains, Enforceability, Uniqueness,*


All Rights Reserved – Dexon Software-2026

*Integrity, Synchronization, and Auditing. For digital transformation leaders, the implication is straightforward: automating decisions with incomplete or inconsistent data does not eliminate risk, but rather amplifies and accelerates their operational consequences.*

## 12.1 Operational decisions

An operational decision is a determination made within a process to select a path, assign a resource, approve or reject a request, calculate a priority, trigger an exception, escalate a case, generate an alert, or initiate an integration. In a goal-oriented TO-BE process, decisions should not be hidden in free text or in the user's informal judgment.

Each decision should document: input data, applicable rule, responsible or deciding driver, expected output, evidence generated, possible exception, and related KPI. BPM CBOK promotes common terminology and generally accepted practices for BPM, which reinforces the need to separate modeling, design, measurement, governance, and technology as connected disciplines, not as isolated activities (ABPMP, 2019).

A BPMN gate without explicit data and rules is not a decision; it is drawn uncertainty. An automated decision without data quality is not automation; it is accelerated operational risk.


All Rights Reserved – Dexon Software-2026

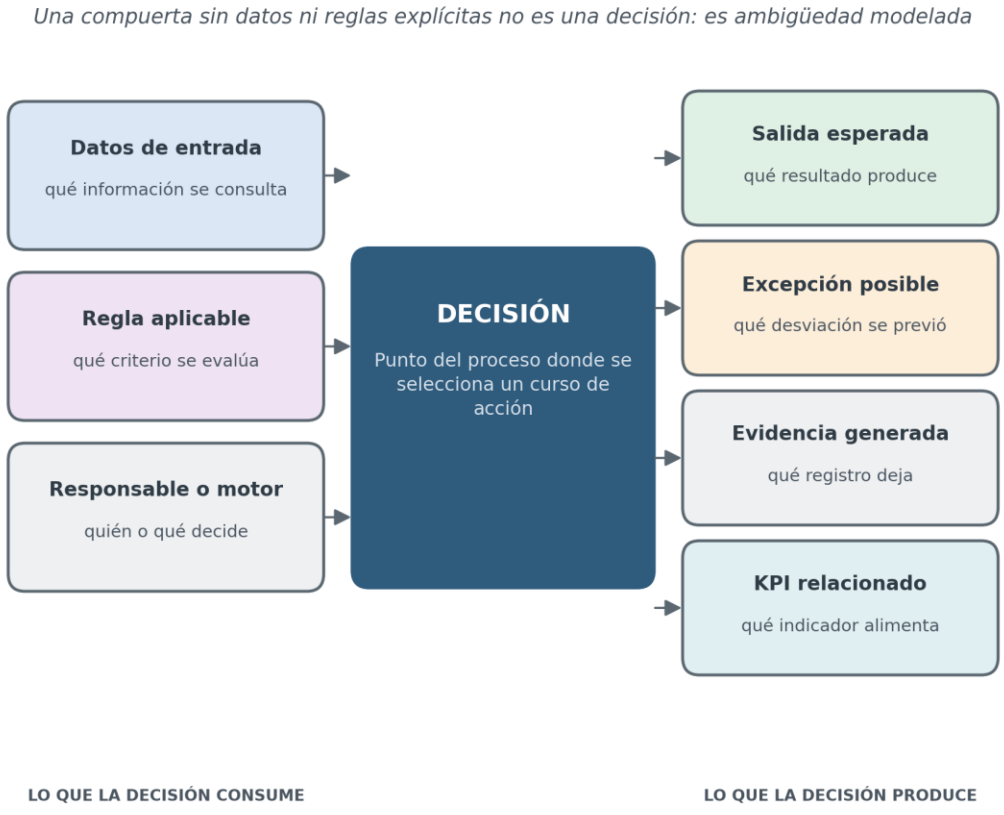


*Figure 15. Anatomy of a decision: what it consumes and what it must produce to be governable.*

## 12.2 Decision Automation

Decision automation should be reserved for decisions that are sufficiently structured, repeatable, traceable, and based on reliable data. When a decision depends on expert judgment, contextual ambiguity, or negotiation, it can remain human, but the minimum information that the responsible party needs and the evidence that must be left behind must be documented.

Objective-Driven Process Design demands that every decision be evaluated against the process goal. An automated decision is justified if it improves timeliness, consistency, compliance, control, scalability, or experience without degrading the quality of the outcome. A human decision is justified if it provides contextual criteria, professional judgment, exception management or explicit responsibility. Section 12.3 refines this binary alternative and Chapter 22 extends the discipline to decisions assisted or executed by artificial intelligence.


All Rights Reserved – Dexon Software-2026

## 12.3 Execution Zones: People, Algorithms, and Agents

The previous section framed the decision as a binary alternative: automate it or keep it human. The emergence of software agents capable of executing tasks with limited autonomy forces us to refine it, because an agent is neither a deterministic algorithm nor a person. It decides probabilistically, it gets it wrong in ways that an algorithm doesn't, and yet it executes without moment-to-moment supervision. Treating it as ordinary automation underestimates its risk; treating it as a human decision underestimates its scope.

This methodology therefore distinguishes three execution zones within the same process, and requires each activity to declare in which it operates.

| Area | Executor | Nature | What the design demands |
|---|---|---|---|
| Augmented people | Tool-assisted and AI-assisted human | Judgment, negotiation, professional responsibility | Minimum information that the person in charge needs, evidence that he must leave and what the assistance suggests without deciding for him |
| Algorithmic | Rule, calculus, or deterministic integration | Reproducible and auditable by construction | Explicit rule, declared input data, and verifiable result |
| Agents | Software agent with limited autonomy | Probabilistic; runs without continuous supervision | Stated scope, escalation criteria, irreversible limit of action, and evidence for each intervention |


All Rights Reserved – Dexon Software-2026

*Figure 16. The three execution zones of a process and the criteria that assigns work between them.*

The allocation criterion is enunciated in a single rule, which this methodology adopts from the practice of enterprise architecture: **as algorithmic – deterministic – as possible, as probabilistic as necessary** (Villalobos, 2026). Its strength is in the order of terms. The probabilistic zone is not chosen because it is available, but because the deterministic zone is not enough. Applied to the design of a process, it forces each intervention of an agent to be justified by what the rule cannot resolve – ambiguity of natural language, non-modelable variability, volume of cases irreducible to a decision tree – and not by the novelty of the technique.

The executor class is declared **by executable activity**, and how it executes determines how far that granularity goes. When the segment runs through state transitions, the activities share a template, and the class is applied to the process; When running as a case family (section 12.4), each activity has its own template and therefore **its own type of executor**. The granularity that this section demands is then fully achievable, and the statement is edited from the design itself: the diagram is the door, not a separate screen.

This typification is not a descriptive label. The R11 constraint in section 4.6 makes it a condition of conformance, and the Workspace ↔ Role relationship in Chapter 4 makes it searchable. Its practical


All Rights Reserved – Dexon Software-2026

consequence is that the question "what part of this process does an artificial intelligence execute?" is no longer answered by the memory of the implementing team and is answered with a query to the repository, which is exactly what an audit in a regulated organization will require.

## 12.4 The two ways of executing a trajectory

The previous sections dealt with the decision and the executor. A question that V7 left implicit and that design cannot avoid is missing: **how a trajectory is materially executed when the work is no longer sequential.**

The same TO-BE path supports two forms of execution, and the designer **does not choose a modality**: he draws a single path and the shape of each segment is derived from the diagram itself. **State transitions** is the default shape. A single case advances through the stages of the process, and it is the right way as long as the work is sequential: at each moment there is exactly *where it is*.

**Case family** is the form of concurrent work. Each activity runs as its own case, with its squad and category, chained to its predecessors by results. It is the correct way when two or more activities are carried out at the same time with different managers, because a single case cannot be in two places.


All Rights Reserved – Dexon Software-2026

*Figure 17. The two ways of executing a trajectory and the rule that determines which one applies to each segment.*

## The conversion rule is declarative, not configurative

A segment runs as a family when one of two things happens: it is fed by a **parallel gate** —the concurrency BPMN signal—or the designer **explicitly declares** that the activity runs as its own case. The second is justified even if there is no parallelism: an activity with service agreements, data controllers or own data deserves its own case.

The condition **spreads to successors**. Once the trajectory becomes a family, it continues in a family until its ends. The return of family to states does not exist as an execution mechanism, and to fake it in the diagram would be to promise a semantics that the engine does not have. It should be said as a property of the model and not as a pending limitation: the competition, once opened, closes in its entirety, not in return.

## The vocabulary of the decision is the results

In the family segment, the label of each arc is the **outcome** chosen by the person who closes the


All Rights Reserved – Dexon Software-2026

activity, expressed in business language – "Approved", "Rejected", "Returned for correction" – and visible on the case screen. An unlabeled arc receives the generic continuation result; An arc that reaches an end of the trajectory is a result of **closure**, and the organization's policy decides whether the entire family or just that branch ends.

Together with R14, this makes up the minimum unit of a well-documented decision: the **gate question** tells what is asked, and the **results** of its outputs tell what can be answered.

### The two formal seams

The passage between shapes occurs at two points, and both have a ruler.
The **fork from the state flow**: the result that triggers the family is anchored to the state where it was drawn and is only offered there. The root case waits while the family works, and cannot be closed with open children.

The **join for completeness**: An activity fed by several concurrent predecessors is created when **they all** close. The strictest variants—uniqueness by family, completeness expressions—are advanced settings and are not part of the design.

### Consequence of traceability

The arches actually executed in each family are recorded on a case-by-case basis. The designed trajectory and the executed trajectory are no longer comparable only by inspection: they are compared by consultation, which makes the family of cases a direct input to the mining process of chapter 20.

## 12.5 Data Quality as a Condition of Automation

Data quality is a condition of automation. DAMA-DMBOK defines data quality management around dimensions such as accuracy, completeness, consistency, timeliness, validity, and uniqueness, and links data management with governance, architecture, integration, metadata, and BI (DAMA International, 2017).

At Dexon, quality must be designed from the source: forms with validations, closed domains when applicable, state-mandated, uniqueness rules, referential integrity, enrichment, synchronization, auditing, and change controls. If quality is corrected only at the end, the process will have already generated rework, dubious decisions, and unreliable indicators.

# Chapter 13. Integration between processes, data and automation

*The chapter shows how the designed architecture becomes coordinated execution between people, data, rules, and systems. Forms and events are not isolated interface components, but rather capture, query, update, and trigger points within the process. ETL integrations and flows should be*


All Rights Reserved – Dexon Software-2026

*defined from the TO-BE, specifying participating systems, data exchanged, transformations, events, errors, evidence, and indicators, rather than being relegated to a subsequent technical task. The CMDB and the CIs provide the relationship between services, processes and governable resources, which allows the analysis of impact and dependency beyond the inventory. The chapter distributes these functions among Dexon's integrated modules: Designer for architecture, Workflow for execution and rules, ETL for interoperability, ECM for unstructured information, and BI for measurement and analytics. For the CIO, the value lies in replacing one-off integrations and fragmented automations with an orchestrated architecture where all technological interaction preserves business context, traceability, and evidence.*

## 13.1 Forms, Events, and Execution

The form is the point where the process captures, queries, or updates data. In Dexon, executable BPMN design is implemented through states or activities with layouts, transitions, rules, Related Data, mappings, and notifications.

Events are trigger or change points. They can start a process, modify its state, trigger a rule, receive external information, trigger an integration, or generate an alert. The Dexon architecture allows process instances to be started or updated manually, by events such as receiving mail or changes in external sources, programmatically or from external web pages.

## 13.2 Integrations, ETL, and Services

Integration makes Dexon an orchestrator between people, processes, and information systems. Dexon ETL is the module that executes data flows, integrates them into workflows and allows interoperability with internal and external systems.

Integration should not be seen as a downstream technical task. It must be designed from the TO-BE: which systems participate, what data is exchanged, what rules transform the data, what events trigger synchronization, what errors can occur, what evidence remains and what indicator measures the result. The platform also allows real-time processing through data services, SOA/EDI and Change Data Capture that evidences changes in information sources and triggers integration processes.

## 13.3 CMDB and ICs as operational support

CIs represent governable elements that support services and processes: applications, assets, technology components, critical resources, or infrastructure elements. The operational guidance indicates that what is governable inventory should be promoted to CI in the CMDB, including business applications with the "Business application" type.

The methodological value of the CMDB is not only in inventorying. It is about connecting services, processes, capacities, data, risks and indicators with the resources that support them. Thus, when a service fails, the organization can analyze the impact on capabilities, processes, users, and indicators. This architecture is consistent with certified IT asset and configuration management practices,


All Rights Reserved – Dexon Software-2026

whose criteria require precisely distinguishing service components from assets, linking them, and sharing common data between both repositories (Pink Elephant, 2025, 2026).

## 13.4 Role of Dexon Modules

Dexon Workflow executes processes and ensures that business rules are applied; it also allows the design and implementation of processes, activities, business data and associated organizational structure. Dexon Designer models business architecture, including processes, strategic objectives, data model, organizational structure, information systems, people, actors, and roles.
Dexon ETL integrates external data streams and systems, Dexon BI produces reports, dashboards, indicators, maps, and analytics, and Dexon ECM manages unstructured data such as documents, OCR/ICR, digital signatures, and automatic document creation. Together, these modules allow the process to be not just flow, but **executable architecture with data, rules, evidence and indicators**.

# Chapter 14. Data governance and operational governance

*This chapter defines the system of responsibilities and controls necessary to support the executable architecture. It distinguishes the owner of the process, responsible for objective, design, performance, approval and improvement, from the owner of the data, responsible for definition, quality, integrity, security, lineage and evolution. Coordination between the two is mandatory because processes create and modify data, and data condition decisions, auditing and indicators. Lineage, historicization and traces make it possible to reconstruct the origin and transformation of information and to demonstrate who acted, when and on what object. Post-exit governance requires evaluating any changes to processes, data, rules, indicators or ICs for their impact on the entire chain. The chapter further adopts COBIT's distinction between governing and managing: management evaluates, prioritizes, and monitors, while management plans, configures, operates, and supports. For senior management, this separation avoids committees without evidence and technical modifications without a mandate or analysis of consequences.*

## 14.1 Process Owners and Data Owners

Operational governance requires distinguishing responsibilities. The process owner is responsible for the objective, design, performance, approval, changes and improvement of the process. The owner of the data is responsible for definition, quality, use, integrity, security, lineage and evolution of the data. Both must be coordinated because no process can be governed without data and no critical data can be governed without knowing the processes that create or modify it.

The operational guide requires that the process file record objective, owner organizational unit, approval, service that materializes and SIPOC inputs/outputs. That file should be methodologically expanded to include relationship with critical entities, automated decisions, evidence, indicators, and data responsibilities.


All Rights Reserved – Dexon Software-2026

## 14.2 Lineage, historicization and auditing

Lineage allows us to know where a piece of data comes from, what process captured or transformed it, what rule modified it, which system received it and which indicator used it. Historicization allows us to know how a piece of data changed over time. The audit allows you to show who did what, when, from where and on what object.

Dexon BPM has auditing functionalities through logs that identify the date, time and user associated with the operation, and maintains traceability on actions of creation, updating, modification, non-repudiation or deletion of information components. In addition, the platform maintains a detailed trace of activities performed by users within a process, including action, date, time, IP, device, and user, and that this information is saved for active and closed instances.

## 14.3 Post-departure government

The government does not begin after going into production; it is designed from the ground up. However, it becomes visible when the process starts operating. The operational guidance states that, after the live output, the capability map and catalog are revised when the business model changes; the processes have an owner and date of approval; they must be reapproved when they change; and the data model evolves by designer, never by direct SQL.

This principle must remain a methodological rule: any change in process, data, rule, indicator or IC must be evaluated by impact on the traceability chain. Changing a field can affect a form, rule, integration, KPI, audit, and automated decision.

## 14.4 Governing and Managing – The COBIT Distinction

The operational governance described in this chapter adopts the central distinction of COBIT: to govern is to assess the needs and choices of stakeholders, to lead through prioritization and decision-making, and to monitor performance and compliance against the agreed direction; To manage is to plan, build, execute, and monitor activities aligned with that direction (ISACA, 2018). In Dexon's practice, process owners, data owners, and approval bodies govern that set goals, approve TO-BE designs, evaluate evidence, and decide on changes; They manage the roles that configure, operate, integrate and support the platform.

This separation avoids two common pathologies. The first is government without evidence: committees that approve processes they cannot observe, on hand-built reports. The second is management without a mandate: technical teams that modify processes, data or rules without anyone evaluating the impact on the objective, service or indicator. COBIT's cascade of goals—from stakeholder needs to business, alignment, and governance goals—is structurally analogous to the Dexon traceability chain, making it easier for organizations already operating under COBIT to integrate this methodology into their existing governance system (ISACA, 2018).


All Rights Reserved – Dexon Software-2026

*Figure 18. The two planes of responsibility and the pathologies that appear when they are confused. Chapter 15. Indicators, measurement and evidence*

The chapter states that an indicator is a test of the objective and not a decorative reporting element. The measurement must cover strategic contribution, capacity, service, process and data quality, and each KPI must declare goal, meaning, frequency, owner, source and link with an object and attribute of the metamodel. The methodology uses Goal-Question-Metrics to derive questions and metrics from satisfaction criteria, and links strategic measurement to objectives and key results when this framework is used. It also adopts the logic of ISO/IEC/IEEE 15939 to differentiate between measurement object, attribute, base measure, derived measure and indicator with decision threshold. This eliminates floating metrics calculated only because the data is available and objectives that lack evidence. For CEOs and planning directors, the chapter proposes a measurement system capable of connecting strategy and operation without informal leaps, making it possible to distinguish activity, performance and value actually achieved, and to use evidence as a basis for learning and redesign.


All Rights Reserved – Dexon Software-2026

## 15.1 Indicators as proof of the objective

In Objective-Driven Process Design, an indicator is not a decorative report. It is evidence of compliance with the objective of the process. If the objective defines outcome, stakeholder, quality, opportunity, constraints, and evidence, indicators must measure exactly those dimensions.
The operational guide establishes that KPIs must have a goal, meaning, frequency and owner, and must be linked to the real process and strategic objectives if the strategic stage was carried out. This rule should be maintained as a minimum criterion, but extended to five indicator families:

- **Strategic indicators:** measure contribution to objectives, goals and initiatives.
- **Capacity indicators:** show whether a capacity is available, mature, supported, or at risk.
- **Service indicators:** measure value delivered, compliance, experience, criticality and availability.
- **Process indicators:** measure cycle, backlog, productivity, SLA compliance, rework, and exceptions.
- **Data indicators:** measure completeness, consistency, duplicity, timeliness, traceability, and usage.

## 15.2 Operational Evidence

Evidence is the set of data, documents, logs, statuses, decisions, signatures, comments, times, integrations and reports that allow demonstrating what happened in a process instance. Without evidence, measurement becomes interpretation; With evidence, the organization can audit, explain, improve, and learn.

Dexon BI and Report Designer allow you to exploit information through reports, dashboards, graphs, maps and indicators; reports can be used as a source for alerts, notifications, ETLs and ad-hoc queries on process data. This connects measurement with automation: the indicator not only informs, it can also trigger corrective or preventive actions.

## 15.3 Measurement for Learning and Improvement

A goal-oriented process must produce learning. If the data shows that the process meets deadlines but does not satisfy the stakeholder, the objective is poorly measured or the design is incomplete. If the process meets the result but with high rework, there are problems of quality, rules or experience. If KPIs are correct but not used to make decisions, the government is weak.

Recent versions of the BIZBOK Guide (Business Architecture Guild, 2026) highlight enhancements in capability and value mapping, dynamic rules-based routing, and performance metrics to clarify value creation, align capabilities with value delivery, and enrich performance management. This orientation is fully compatible with Dexon: measuring is not closing the cycle; To measure is to open the next cycle of redesign.


All Rights Reserved – Dexon Software-2026

## 15.4 Disciplined derivation of indicators: GQM and OKR

The rule that indicators are derived from the target needs a procedure, not just a conviction. The Goal-Question-Metric approach offers this procedure: each goal is formally specified—object of study, purpose, quality focus, point of view, and context—questions that characterize its fulfillment are derived from the goal, and metrics are derived from each question that answer it with data (Basili et al., 1994). The structure of the process objective defined in Chapter 8 – stakeholder, outcome, success criterion, constraints, evidence – is deliberately isomorphic with this template: each satisfaction criterion must generate at least one assessable question, and each question at least one indicator with a data source identified in the business model.

This procedure prevents the two most common defects of measurement systems: orphan metrics, which are calculated because the data exist but do not answer any relevant questions, and hypotheses without evidence, success criteria that no one can verify because no data supports them. Applied in Dexon, the method also requires that each metric declare its source: which entity, which field, what state of the process and which rule feeds it.

For the strategic link, the OKR framework is complementary: the objectives express qualitative intent and the key results make it measurable and time-bound (Doerr, 2018). In organizations that manage their strategy with OKRs, the achievement and service indicators defined in this methodology are the natural source of key results, which integrates the operation of processes with the rhythm of strategic management without duplicating measurement systems.

## 15.5 The object of measurement: every measurement rests on an attribute of the metamodel

The GQM derivation responds to where an indicator comes from; it remains to formalize what it rests on. This methodology adopts the measurement information model of the ISO/IEC/IEEE 15939 standard: the entities have attributes; Base measures quantify attributes using a measurement method and scale; derivative measures combine base measures; and the indicators apply decision criteria – thresholds – on these measures to respond to an information need (ISO/IEC/IEEE, 2017). In terms of the metamodel in Chapter 4, the rule is only one: **every measured quantity is anchored to an object of measurement and to one of its declared attributes**, where the object of measurement is any object of the metamodel.

The anchor is made up of **three** statements, not two, and the third is the one that V7 implied:

1. the **type of object** measured – process, service, offer, capacity, entity, risk, workspace;
2. the **concrete record** of that type: *which* process, *which* service. Without this statement, the indicator measures "some process", which is almost as floating as measuring none;
3. The **measured attribute** : opportunity, backlog, completeness, availability.

For processes, the direct link between the measuring element and the process satisfies the second statement. The golden rule is then accurately restated: **a measurement element without a declared**


All Rights Reserved – Dexon Software-2026

**type, record, and attribute is not an indicator; it is a number with a name.** Section 15.6 specifies a point where this methodology deliberately departs from the structure of the standard: base measure, derived measure and indicator are not treated as three distinct types, but as roles of the same object.

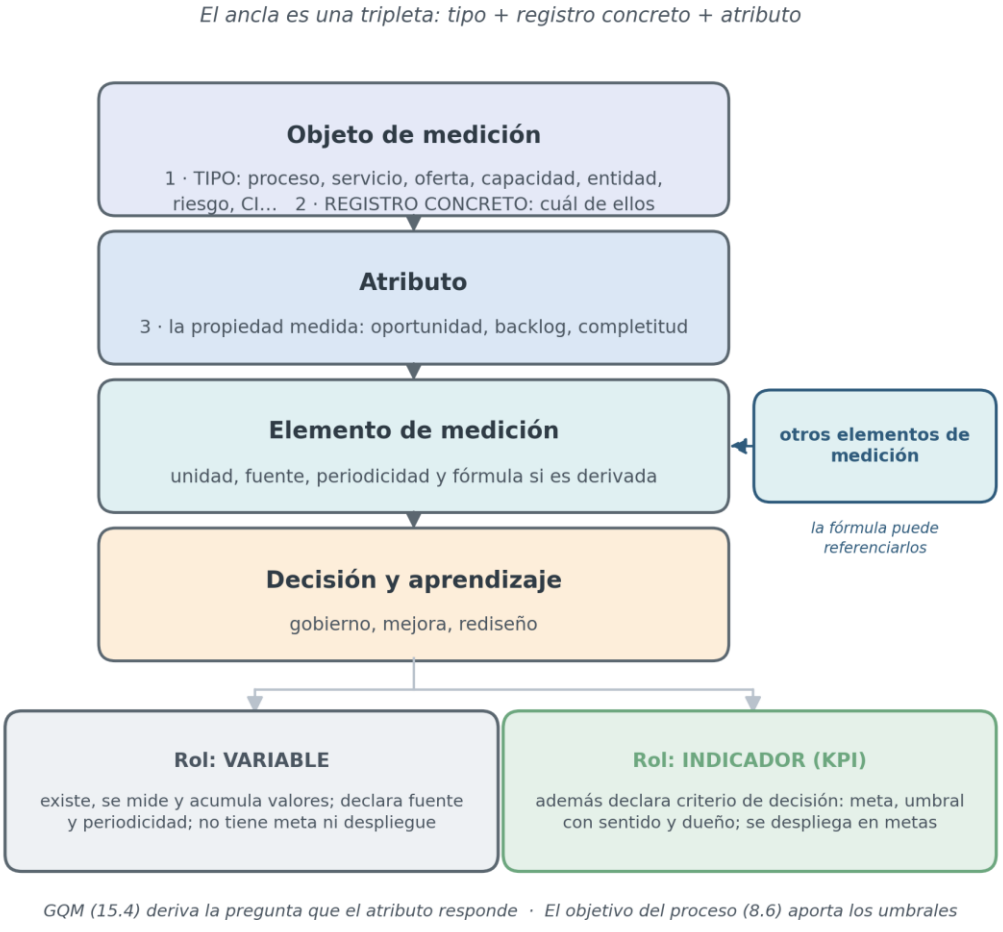


*Figure 19. The measurement chain and the two roles of the measurement element: of the object and its attribute to the decision, with the composition of formulas that the metamodel declares. GQM provides the question; The formal objective provides the thresholds.*

This formalization unifies the five indicator families in Section 15.1, which are revealed as the same structure applied to different measurement objects:

- strategic indicators quantify attributes of objectives and goals;
- those of capacity, attributes of capabilities —availability, maturity, support, risk—;
- those of service, attributes of services and offers —compliance, experience, demand—;
- those of process, attributes of processes and activities —cycle, backlog, reprocess, exceptions—; y


All Rights Reserved – Dexon Software-2026

- those of data, attributes of entities —completeness, consistency, opportunity—.

The thresholds of the formal specification of the objective (section 8.6) are, in this model, the decision criteria applied to the attributes measured:
The objective circle → criterion → threshold → attribute → data is closed without informal links.

It is worth specifying the relationship with the attributes of IQs, because the analogy is tempting but inaccurate. Both are typical properties of governed objects in the repository, and that's how they're similar. But configuration attributes describe the state that an object must have and are controlled (its deviation is a nonconformity), while measurement attributes quantify the behavior that an object exhibits and are observed (its deviation is information). The configuration is audited; Performance is measured. The IC is not the general model of which the measurement objects are a case: it is just another measurement object, whose configuration attributes can also be measured (effective availability, associated incidents, obsolescence). The correct generalization is that of the metamodel: any object in the chain—capacity, service, supply, process, activity, entity, CI—can be measured, and the relationship table in Chapter 4 incorporates this relationship as a formal part of the method.

## 15.6 Variables and indicators: one object, two roles

The ISO/IEC/IEEE 15939 model distinguishes three successive constructs: the base measure, which quantifies an attribute using a measurement method; the derived measure, which combines base measures using a function; and the indicator, which applies a decision criterion on those measures. This methodology adopts the chain, but **does not adopt the trichotomy**, and it is worth explaining why, because it is a design decision of the metamodel and not a simplification.

All three types of the standard are not stable. The same magnitude is a base measure in one calculation and a derived measure in another; and it becomes an indicator the instant someone attaches a goal to it. "Requests received" is a baseline measure when it feeds a resolution rate, and is an indicator of the day the organization decides to set an expected demand threshold for you. Classifying them separately forces objects to be reclassified each time the context of use changes, and makes the lineage between them depend on which type each one was assigned to.

That is why the metamodel declares **only one type, the measuring element**, and two **roles** that this type can play. This is the same criterion with which section 4.3 treats the object of measurement: role and not type.

| | **Variable** | **Indicator (KPI)** |
|---|---|---|
| What it declares | Unit, data source, capture periodicity and, if derived, formula | All of the above, plus goal, threshold with meaning and owner |


All Rights Reserved – Dexon Software-2026

| | Variable | Indicator (KPI) |
|---|---|---|
| What it does | It exists, is measured and accumulates values over time | It also enables a decision: the result is compared against a criterion |
| What the method requires | Declared source and periodicity (R12) | Declared goal, meaningful threshold and owner (R12) |
| Deployment | Not deployed | It is displayed in goals by person responsible and period |

The difference, then, is not of nature but of **what is stated above**. A measurement element without a decision criterion is a variable: it exists to be measured and to feed others. An element with a decision criterion is an indicator: it exists for someone to decide based on it.

## The composition and its dependency graph

From this unification follows the property that the metamodel incorporates in section 4.4: **the formula of a measurement element can reference other measurement elements**, whether variables or indicators. A compliance indicator can be composed of two catch variables; an aggregate index can be composed of three indicators already published; and a strategic indicator can consolidate process indicators without any of them ceasing to be what it is.
These references form a graph of dependencies, and that graph has two consequences that the methodology explicitly states.

The first is calculation: the order of evaluation is not arbitrary. An element cannot be calculated before those on which it depends, so the graph must be traversed in topological order. Dexon BPM manages this dependency structure and resolves the order of calculation, which transfers to the environment a correction that would otherwise have to sustain the procedure: it is another case of the principle of section 16.3, according to which native control is preferred to the procedure when the rule can be implemented.

The second is integrity: the graph must be acyclic. A formula that depends, directly or indirectly, on its own result is not calculable, and no order of evaluation solves it. The R13 constraint declares this property and classifies it, along with R7, among those that are verified by graph analysis and not by querying the repository.

The third consequence is of **government**, and it is not minor. As the dependencies are declared, the impact analysis of chapter 19 also covers measurement: before modifying the formula, source or periodicity of a variable, it is possible to consult which indicators consume it – directly or transitively – and which decisions would be affected. Without the declared graph, that question is only answered by reading formulas one by one, which is how indicator batteries degrade over time.

## An Implementation Warning

The fact that variable and indicator are the same object is a virtue of the model, but it opens a door:


All Rights Reserved – Dexon Software-2026

nothing prevents, by construction, attaching a goal to what should be a magnitude of catch, nor leaving without a goal what is reported as a management indicator. When the platform does not force distinction, discipline cannot remain in the habit of the implementing team. That is why R12 exists: it forces the role to be declared and makes it verifiable, by consultation, that each element complies with what its role requires.

## 15.7 Operating Model Outcome Attributes and Quality Attributes

The previous two sections established that every measurement element quantifies a declared attribute of an object in the metamodel. It is necessary to distinguish two families of attributes that this formulation groups together and that design must treat differently.

Result **attributes** describe whether the process achieves what it exists for: cases resolved within the threshold, requests decided correctly, value effectively delivered to the stakeholder. They are born from the objective of the process and are those that chapter 8 makes verifiable through satisfaction criteria.

The **quality attributes of the operating model** describe how the system that produces that result behaves, regardless of the particular case: throughput, response time, unit cost of execution, reliability, scalability, and load on each workspace. A quality attribute is, in the formulation of enterprise architecture, an observable property of the operating model that must be quantifiable (Villalobos, 2026). If it cannot be quantified, it is an aspiration and not an attribute, and it does not belong to the model.

The distinction matters because the two families fail in different ways and correct each other in different places. A process can fully satisfy its objective in each case and be unfeasible in volume. You can meet all your outcome thresholds and concentrate the load in a workspace that becomes a bottleneck. None of these defects appear in an achievement indicator, and that is why the TO-BE design must declare quality attributes as well as satisfaction criteria.

Formally, both families are treated equally—they are attributes of metamodel objects, measured according to the string in section 15.5—and this uniformity is deliberate: it allows a single measurement mechanism to cover the performance of the process and the health of the system that runs it. What changes is the object of measurement. The result attributes are placed on the process, its activities and the entities it produces; quality attributes are preferentially placed on workspaces, CIs, and capabilities, which is where the operating model becomes observable.

## 15.8 Measurement alignment is relationships, not fields

Indicator systems typically record institutional alignment as **fields on a token**: one field for the strategic objective, one for the initiative, one for the process. The consequence is known: the alignment is read, not consulted; it degrades without anyone noticing; and it does not admit its own attributes.


All Rights Reserved – Dexon Software-2026

In this metamodel, each alignment is a **queryable relation** of the measurement element, and several of them carry attributes that a field could not hold.

| Measuring Element Ratio | Attributes of the relationship |
|---|---|
| Strategic objective | Weight of the element inside the lens |
| Initiative | Weight |
| Portfolio of plans, programs and projects | **Own Goal and Weight** by Portfolio Item |
| Organizational Unit (Deployment) | Weight |
| Strategic theme and axis | — |
| Capacity | — |
| Risk | — |

The last two deserve comment, because they are the ones that this version incorporates.

The relationship with **capacity** answers a question that without it has no answer: when several processes fail to comply at the same time, do they share a weak capacity? A system of indicators that only aligns with objectives and processes reads three related breaches as three independent problems.

The relationship with **risk** is what converts a measurement element into a **KRI.** And it is worth stating it clearly because it contradicts the usual practice: **the KRI is not a box, it is a relationship.** The same element can have a goal —and therefore be a KPI— and a link to risk —and therefore be KRI—, without duplicating the object or forcing an artificial choice.

From this follows a consequence that simplifies the design: the classification of the measurement element – strategic, process, service, data, capacity – is derived from the anchor and its relationships. It is not written. A classification field that is filled in by hand is an opportunity for contradiction with the structure; a derived classification cannot contradict it.

## 15.9 Alignment Boards

The vertical alignment of an organization – so that everyone knows which indicator it responds to and to which goal – is materialized in **dashboards**: rows that assign a measurement element to the person who answers for it, by fiscal year, with **alignment role**, **goal** and **weight** of the assignee.

The **role of alignment** distinguishes three forms of responsibility that practice confuses: **management** (leading and coordinating resources for the result), **success** (directly generating the result) and **influence** (contributing indirectly, through other areas). Without that distinction, accountability becomes a discussion about who is to blame; with it, it becomes a conversation about


All Rights Reserved – Dexon Software-2026

what corresponds to each person.

There are two boards, and they do different things.

The **base board** assigns the item to either a **person** or an **organizational unit**, or both. It is the unit for deploying goals and capture. Two boards of the same element and the same year can coexist for different units: uniqueness is given by element, person, year and unit, not by element.

The **functional role board** is the architectural view: what role the element manages, who it measures its success and who it influences, with its vision element, its theme or axis and its visibility on the strategy map.

A precision that avoids the most common mistake of deployment: **the goal of the board can differ from the institutional goal**, and it must be able to do so – that is deploying. Goals per period come down from the dashboard, not the measurement item.

*De la arquitectura diseñada a la operación gobernada, capítulo a capítulo*

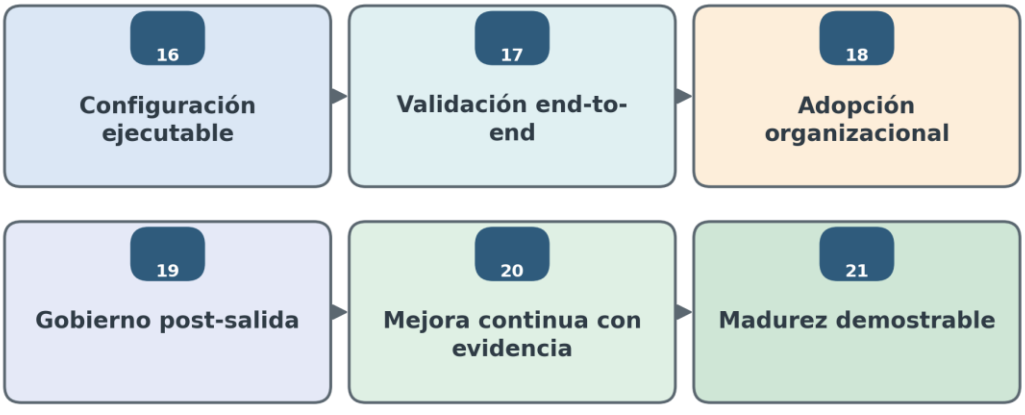


*Figure 20. From Engineered Architecture to Governed Operation (Chapters 16-21): Executable Configuration, End-to-End Validation, Adoption, Post-Output Governance, Continuous Improvement, and Maturity*

# Chapter 16. Implementation of the executable architecture in Dexon BPM

*This chapter redefines implementation as the materialization of an architecture, not as the translation of a diagram into a tool. The executable process integrates BPMN, data, rules, forms, roles, services, offers, SLAs, integrations, reports and indicators, preserving the objective and criteria established during the design. The central rule is that each configuration decision must correspond*


All Rights Reserved – Dexon Software-2026

*to an architectural decision: an activity exists because of its contribution to the objective, a datum because of its operational or evidentiary function, a rule because of the control it exercises, and an indicator because of the question it answers. The configuration is also presented as a governance contract between architecture, operation and technology, with a preference for native platform controls over procedures dependent on memory or manual discipline. For the CIO and transformation managers, this approach bridges the gap between design and execution: compliance is no longer solely dependent on desk reviews and is reinforced by relationships, permissions, mandates, and traces built into the operating environment.*

## 16.1 To implement is to materialize architecture

Implementing in Dexon BPM does not mean translating a diagram into a tool. It means turning a business architecture into an executable system composed of processes, data, rules, forms, roles, services, offers, SLAs, integrations, reports, and indicators. The Dexon platform natively implements the traceability chain through connected editors for capabilities, value chain, service catalog, offers, process sheet, data/CIs and indicators.

The implementation must preserve the methodological intention of the design. If the TO-BE was built under Objective-Driven Process Design, the configuration must preserve the objective of the process, its satisfaction criteria, its BPMN trajectories, its mandatory data, its rules, its risks and its indicators. A process is not implemented when it "runs"; it is implemented when it executes the design that maximizes the fulfillment of its objective and leaves evidence to demonstrate it.

Dexon BPM is comprised of integrated modules – Dexon Designer, Dexon Workflow, Dexon ETL, Dexon ECM and Dexon BI – aimed at automating complex processes, managing large volumes of data and aligning processes, data, people and technology with strategic objectives. This architecture allows the implementation to be a natural extension of the methodology, not a manual translation or a separate technical reconstruction.

## 16.2 From methodological design to executable configuration

The design-to-execution step must follow a central rule: **each configuration decision must correspond to an architecture decision**. The process file, the BPMN TO-BE, the data model, the rules, the roles, the SLAs, the offers and the KPIs are not independent artifacts; they are seen from the same operational solution.

In Dexon, the executable process is configured from the BPM Modeler and is developed as a sequence of states or activities, layouts, transitions and rules; for each activity, forms, related data, state-mandated, assignments and notifications are defined. These elements are, in terms of the metamodel, the materialization of the **workspace** in section 4.3: the layout, the form, the task tray, and the offer portal are the declared places where the work is executed. Configuring them, therefore, is not an interface decision but the implementation of the R11 constraint, and from them the quality attributes of the operating model that section 15.7 distinguishes from the outcome indicators are obtained. Therefore, the configuration must answer methodological questions: what


All Rights Reserved – Dexon Software-2026

objective does this activity satisfy, what data does it capture, what rule it executes, what risk it controls, what evidence it produces and what indicator it feeds.

The deployment must also close the link to services and offers. The operational guide establishes that the initial layout of the workflow makes the offer requestable from the portal and that, when closing the circle, it must be confirmed that the process is linked to capabilities and that the offer is visible and requestable. This rule prevents the process from running technically but disconnected from the user experience and value catalog.

## 16.3 Configuration as a governance contract

The executable configuration should be treated as a governance contract between architecture, operation, and technology. It defines what each role can do, what data is mandatory, what path each case follows, what rules are applied, what evidence is retained, what exceptions are allowed, what SLAs are measured, and what indicators are reported.

TOGAF calls "Architecture Contract" the instrument that governs implementation and deployment to ensure compliance with the defined architecture (The Open Group, 2022a). At Dexon, the equivalent contract is expressed in the combination of approved process token, BPMN TO-BE, data model, rules, Related Data, SLAs, solicitable offer, permits, reports, and exit criteria.

From this contract emerges a principle that runs through the entire methodology: when a methodological rule can be implemented as a native control of the platform, control is preferred to procedure. The chain that prevents the approval of processes without service, the catalog that only makes the complete offer requestable, the mandatory nature of data by state, the verification of prior existence when creating an entity or the automatic audit trail are rules of the method executed by the work environment, not monitored by a document. This government by design responds to the same logic as compliance by design: compliance does not depend on people remembering the rule, but on the system making it difficult to violate it and leaving evidence when it occurs. It is also the practical differentiator of an executable methodology compared to those that live in documents: its rules are not eroded by the rotation of people.

# Chapter 17. End-to-end validation and testing

*The chapter expands on the meaning of validate: it is not enough to verify that the system works, but it must be demonstrated that the process produces the result for which it was designed. End-to-end testing covers five dimensions: functionality, data, rules, objective, and evidence. It must go through main paths, variants, exceptions, rejections, returns, escalations and alternative closures, verifying responsible, forms, available data, conditions, results and error handling. Data validation checks structures, relationships, availability, and protection; the rules review transitions, calculations, assignments, obligatories, SLAs and integrations; and that of the objective confirms that the stakeholder receives a result compatible with the satisfaction criteria. Finally, the evidence must allow us to reconstruct actions, decisions and effects on indicators. For a go-to-production*


All Rights Reserved – Dexon Software-2026

*committee, the chapter provides a more demanding and useful approval rule: a process is not ready because it completes the flow, but because it satisfies the objective, retains control, and can demonstrate what happened.*

## 17.1 Nature of Validation

Comprehensive validation is not only about checking that the platform is not failing. It seeks to verify that the design meets the objective for which it was built. The operational guidance establishes that the live output must include a complete E2E per process: request from portal, execution, SLA, notifications, closing, survey and report, with a minute per process signed by the owner.

In Objective-Driven Process Design, this end-to-end test must validate five dimensions: functional, data, rules, objective, and evidence. Functional validation confirms that the process executes its normal and exceptional trajectories. Data validation confirms that the required data exists, is captured, queried, modified, and protected correctly. Rule validation confirms that decisions are executed under explicit criteria. Goal validation confirms that the output produced satisfies the success criteria. Validation of evidence confirms that the organization can demonstrate what happened, who acted, what was decided and which indicator was affected.

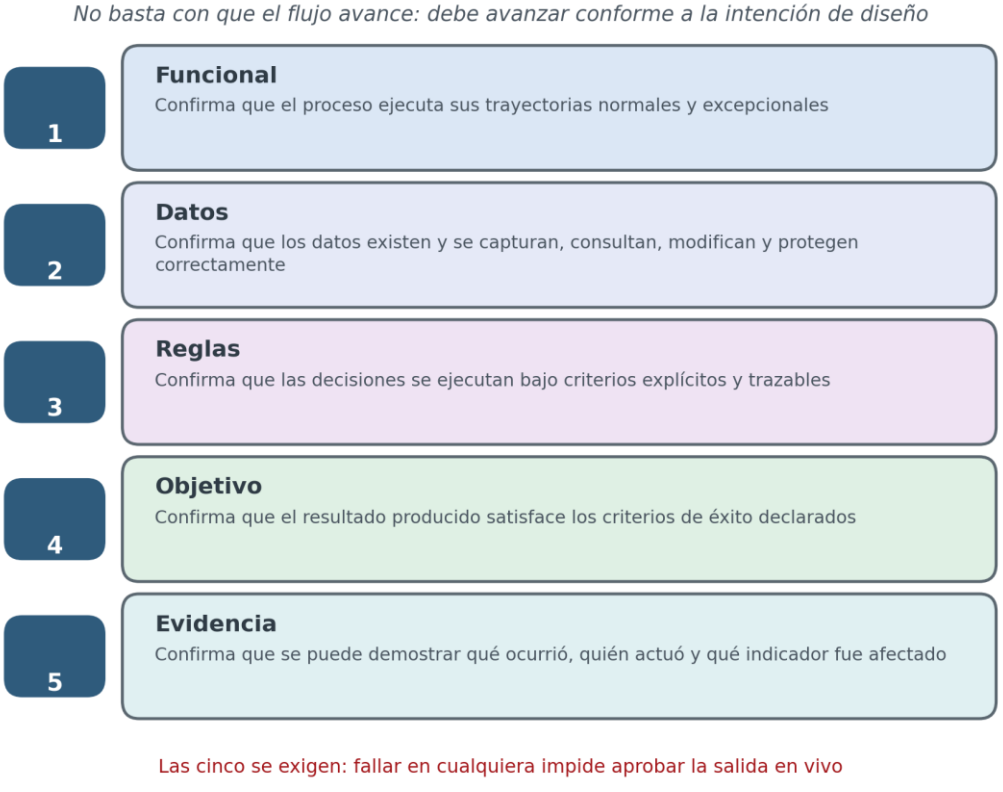


*Figure 21. The five dimensions that end-to-end testing should cover.*


All Rights Reserved – Dexon Software-2026

## 17.2 Functional Validation

Functional validation walks the process from the start event to its end events. You need to test the main path, relevant variants, expected exceptions, rejections, returns, escalations, and alternate closures. It is not enough for the flow to move forward: it must move forward according to the design intent.

Each activity should be checked against its purpose. A human activity must have a responsible party, a form, available data, visible rules, clear instructions and expected evidence. An automatic activity must have activation condition, input data, associated rule or integration, expected output, and error handling. A gate must have complete conditions, mutually exclusive where applicable and based on reliable data.

## 17.3 Data and Rule Validation

Data validation verifies that the entities in the business model are created, related, loaded, and available to the process. The *Dexon Process Implementation Guide* states that entities should be discovered from the ontological model, created in the table designer, connected to processes using Related Data, and promoted to CI when they are governable inventory.

Rule validation checks that transitions, calculations, assignments, enforceability, validations, SLAs, and integrations behave as designed. Dexon has a calculation engine for calculated fields, data validation, DB Mappings and rules for creating BPM activities. Therefore, a rule should not be validated only by its visible result, but also by its traceability: what data it read, what condition it evaluated, what output it produced and what evidence it left.

## 17.4 Validation of objectives and evidence

Goal validation verifies that each tested scenario produces a result that is compatible with the process goal. If the process ends successfully but does not satisfy the stakeholder, the TO-BE should not be approved. If the process complies with the result but does not leave evidence, it is not governable. If it leaves evidence but does not feed indicators, it does not allow learning.

Dexon BPM allows you to attach evidence to the progress of activities – documents, photos, audios, videos – and to sign digitally when it is required to certify compliance with the progress. This capability should be used methodologically to turn the performance into verifiable evidence, not as an incidental documentary repository.

# Chapter 18. Change Management and Organizational Adoption

*This chapter incorporates adoption as a condition for the realization of value. Automation doesn't*


All Rights Reserved – Dexon Software-2026

*transform by itself if users don't understand the change, roles aren't institutionalized, data is captured without quality, or indicators aren't used to decide. The ADKAR model is used to structure the individual transition around awareness, desire, knowledge, capability, and reinforcement, and training is organized by operational responsibilities, not generic functionalities. Process owners, managers, catalog administrators, and analysts require different knowledge about objectives, data, rules, exceptions, indicators, and governance. The chapter also calls for separating permissions by function to avoid dependencies on users with excessive privileges and ensure institutional continuity. For senior management, the message is that adoption must be designed along with the process, with those responsible, expected behaviors, and reinforcement mechanisms. The output to production only creates sustainable value when the organization can operate, interpret and govern the new model in real conditions.*

## 18.1 Exchange as a condition of value

An automated process does not transform the organization if people do not adopt it, if roles are not governed, if data is not captured with quality, or if indicators are not used to decide. Adoption should be designed as part of implementation, not treated as final training.
Prosci's ADKAR model states that organizational change requires individual change and that people must achieve five outcomes: Awareness, Desire, Knowledge, Ability, and Reinforcement (Hiatt, 2006). In the Dexon methodology, this logic translates into five questions: do users understand why the process changes?, do they want to participate?, do they know how to operate?, can they do it in real conditions?, does the organization reinforce the new behavior? This logic is consistent with the classic change management literature, which stresses the need for a sense of urgency, driving coalitions, and institutional reinforcement for change to persist (Kotter, 2012).

## 18.2 Training by role

Training should be organized by operational role, not by generic functionality. The *Dexon Process Implementation Guide* defines training for technicians, catalog administrators, process owners and analysts, differentiating tray/ticket, catalog, tab+BPMN and reports/KPIs. This segmentation must be maintained because each role has distinct responsibilities within the executable architecture.

The process owner must understand objective, traceability, BPMN, indicators, changes and approval. The user manager must understand activities, data, rules, evidence, times and exceptions. The catalog administrator must understand services, offers, visibility, validity and relationship with processes. The analyst must understand KPIs, reports, source data, quality and reading of results.

## 18.3 Role governance and accountability

Role governance ensures that the solution does not depend on specific people but on institutionalized responsibilities. The operational guidance requires profiles and FUNCs per role from instance readiness, including permissions such as bpmModeler, serviceCatalogAdmin, slaEditor, reportDesigner, tableCreation, CMDB Edit, and configAdmin. This separation protects post-exit governance because it avoids "all-powerful" users and distributes responsibilities according to


All Rights Reserved – Dexon Software-2026

function.

Adoption is consolidated when each role understands its contribution to the objective of the process. Technology executes the design, but the organization sustains the behavior.

# Chapter 19. Post-exit government

*The chapter treats the process in production as a living architecture subject to data, exceptions, and real changes. The exit does not close the initiative, but rather initiates a regime of versioning, reapproval and impact analysis. Any modification in activities, rules, SLAs, layouts, data, integrations, offers or indicators may alter the ability of the process to meet its objective and, therefore, must be evaluated for its effects on strategy, capabilities, services, users, risks and measurement. The methodology requires that dependencies be queried in the living repository, avoiding manually reconstructed arrays and reserving expert judgment to interpret consequences. It also states that the evolution of the data model must be done through mechanisms governed by the platform. For CEOs, CIOs, and process owners, the chapter offers a discipline for controlling post-launch transformation debt: preserve consistency, repeat affected tests, update training and documentation, and maintain evidence of approval on each relevant release.*

## 19.1 The process in production as living architecture

The release to production does not close the methodological cycle. It marks the beginning of operational governance. A process in production is a living architecture: it receives real requests, generates real data, reveals real exceptions, evidences bottlenecks, and shows whether the objective is met under operating conditions.

The operational guidance states that, after the live output, the capability map and catalog should be revised when the business model changes; the processes have an owner and date of approval and must be reapproved when changing; and the data model must evolve by designer, never by direct SQL. These rules turn post-output steering into an architectural coherence control system.

## 19.2 Versioning and Reapproval

Any relevant change must go through versioning and reapproval. Changing an activity, rule, SLA, layout, required data, integration, offer, or KPI can alter the process's ability to meet its goal. Therefore, the change should not be evaluated only by technical effort, but by impact on the traceability chain.

TOGAF states that the implementation must ensure compliance with the defined architecture and that governance functions must be maintained while the system is implemented and deployed (The Open Group, 2022a). At Dexon, this compliance must be extended to the subsequent life cycle: any change must respond to what objective it modifies, what capacity it affects, what service/offer it touches, what data it alters, what risk it introduces and what indicator changes. This versioning and


All Rights Reserved – Dexon Software-2026

reapproval regime is the methodological expression of ITIL 4's platform-certified change enablement practice: maximizing successful changes through risk assessment and appropriate authorization (Axelos, 2019; Pink Elephant, 2026).

## 19.3 Changes in Services, Capabilities, and Data

A change in the service catalog may require a review of offerings, processes, capabilities, and ICs. A change in capabilities may require reviewing processes that orchestrate them. A change in data can affect forms, rules, integrations, reporting, and auditing. A change in indicators can alter the understanding of performance.

The post-exit government must manage these dependencies through an impact matrix. No changes should be approved without response: which artifacts are affected, what tests need to be repeated, which users need to be trained, what documentation needs to be updated, and what reapproval date is recorded.

In an executable architecture, moreover, the impact matrix is not built by hand: it is derived from the relationship repository itself. If traceability is materialized as real relationships between strategy, capabilities, services, offers, processes, data, ICs and measurement elements, impact analysis is a query of the model – which artifacts refer to the one that changes – and not a document that someone writes and maintains. This also applies to measurement: as formulas declare their dependencies (section 15.6), before modifying the source, periodicity or formula of a variable, it is possible to consult which indicators consume it directly or transitively, and which decisions would be affected. Without that graph declared, the question is only answered by reading formulas one by one, which is how indicator batteries degrade over time.

The scope of the analysis extends in two more directions. The impact of a **measurement item** includes its **alignment boards** (section 15.9): who is left without a goal if the item is inactive. And the impact of a **risk** includes its uses—the processes whose design it conditions, the portfolio items analyzed, and the KRIs that anticipate it (section 8.4)—which is precisely what makes the rule of not deleting an entry from the registry that is in use applicable. The method therefore requires that the impact be assessed on the living repository, and reserves expert judgment to interpret the consequences of the change, not to reconstruct its dependencies.


All Rights Reserved – Dexon Software-2026

*Figure 22. The impact matrix drafted versus the impact queried to the repository.*

# Chapter 20. Evidence-based continuous improvement

*This chapter makes operational evidence the engine of learning and redesign. Improvement should not be based on isolated perceptions, but on timing, volumes, backlog, SLAs, reprocesses, exceptions, data quality, satisfaction, risks, paths executed, and goal achievement. The proposed cycle follows the logic of observing, interpreting causes, formulating hypotheses, redesigning, testing, deploying, and re-measuring, with explicit correspondence with PDCA. Improvements can be one-off, such as adjusting a rule, layout, integration, or SLA, but they must always retain traceability to the target. Process mining extends this discipline by analyzing event logs to discover the process actually executed, verify compliance, and locate waits, variants, and bottlenecks. For management, the consequence is to move from debates about how the company is supposed to operate to decisions based on how it demonstrates that it operates. Continuous improvement thus becomes an institutional capacity, not a succession of disconnected initiatives.*

## 20.1 Evidence as a Learning Engine

Continuous improvement should not be based on isolated opinions or perceptions of efficiency. It should be based on operational evidence: timing, volumes, backlog, SLA compliance, rework,


All Rights Reserved – Dexon Software-2026

exceptions, data quality, satisfaction, risks, route usage, and goal achievement. In terms of ITIL 4, this cycle integrates the practices of problem management – finding and eliminating causes of recurring failures – and knowledge management – turning what has been learned into reusable assets – both certified on the platform (Pink Elephant, 2026).

Dexon BI allows you to build reports, dashboards, indicators, maps and analytics; reports are used to build KPIs, complement notifications, send information, automate process starts and feed tableros.la *Process Implementation Guide Dexon* requires KPIs with goal, meaning, frequency and ownership, linked to the actual process and strategic objectives where they exist.

## 20.2 Improvement Cycle

Improvement must follow a disciplined cycle: looking at evidence, interpreting causes, formulating hypotheses, redesigning, testing, deploying, and measuring again. The PDCA cycle, formalized by Deming from the Shewhart cycle, defines Plan-Do-Check-Act as a four-step model for executing changes and repeating them continuously in search of improvement (Deming, 1986; Moen & Norman, 2010).

At Dexon, the "Plan" corresponds to interpreting indicators and defining a hypothesis for improvement; the "Do" corresponds to adjusting the design or configuration in a controlled environment; the "Check" corresponds to validating E2E and comparing results; the "Act" corresponds to approving, deploying, documenting, and updating the governance of the process.

## 20.3 Iterative Redesign

Continuous improvement does not always involve redesigning the entire process. It can consist of adjusting a rule, simplifying a layout, improving an integration, redefining an SLA, correcting master data, extending an offer or changing an allocation criterion. However, any improvement must retain traceability with the process objective.

The BIZBOK Guide (Business Architecture Guild, 2026) highlights capability and value mapping, dynamic rules-based routing and performance metrics as elements to clarify value creation, align capabilities with value delivery and enrich performance management. This orientation reinforces Dexon's philosophy: improving is not just optimizing tasks; is to increase the capacity of the process to produce measurable value.

## 20.4 Process mining: evidence of execution as an object of analysis

Evidence-based improvement now has its own scientific discipline: process mining, which uses the event records of information systems to discover the process actually executed, verify its conformity with the designed model, and improve the model based on the observed behavior (van der Aalst, 2016; IEEE Task Force on Process Mining, 2012).

Dexon architecture natively produces the raw material of this discipline. Each instance of the process


All Rights Reserved – Dexon Software-2026

records case, activity, timestamp, user and action; That audit trail is, in technical terms, an event log suitable for discovery and verification of conformance. In addition, in processes with segments that run as a case family (section 12.4), there is a case-by-case recording of the **arcs actually traveled**: which outcome was chosen by who closed each activity and which activity was created accordingly. That trace is a primary source of mining, because you don't have to infer the route from timestamps: the decision was declared in the language of the business. This enables three post-exit governance practices. First, the discovery of the real process: to contrast the routes actually executed with the approved TO-BE trajectories, revealing variants, shortcuts and informal exceptions that no survey workshop detects. Second, conformity verification: quantifying the proportion in which real cases respect the design, its rules and its controls, which makes architectural conformity a measurable indicator and not a presumption. Third, the performance analysis on the model: locating waits, reprocesses and bottlenecks directly on the trajectories, prioritizing redesign where the evidence shows loss of value.

With process mining, the improvement cycle in section 20.2 is closed with execution data and not with perceptions: the organization does not discuss how it believes it operates, but how it proves that it operates.

# Chapter 21. Dexon Maturity Model

*The chapter proposes a five-level model that assesses the evolution from basic documentation to governed executable architecture. Progression comprises structured modeling, integrated automation, traceable architecture, and finally, the ability to measure, version, reprove, and improve with continuous evidence. The criterion for advancement is not the number of automated processes, but the organizational ability to connect objectives, processes, data, roles, indicators, and learning. As such, maturity must be demonstrated with platform-extractable evidence, such as traceability relationships, owners, approval dates, linked indicators, and audit trails, rather than relying on declarative self-assessments. The model is anchored in BPMM, PEMM, and the BPM maturity elements related to alignment, governance, methods, technology, people, and culture, but emphasizes the complete executable architecture. For senior management, it serves as a diagnostic and prioritization tool: it allows them to distinguish tactical automation from sustainable business capability and direct investments towards the gaps that prevent them from governing value from end to end.*

## 21.1 Purpose of the model

The Dexon maturity model allows you to evaluate the evolution of an organization from basic process documentation to an executable, traceable, governed, and goal-oriented business architecture. BPM CBOK is presented as a common body of knowledge that provides standardized terminology, accepted practices, and competencies for BPM professionals (ABPMP, 2019). This methodology takes that professional orientation and adapts it to Dexon architecture.


All Rights Reserved – Dexon Software-2026

| Level | Name | Methodological description |
|---|---|---|
| 1 | Basic documentation | The organization documents processes and data, but traceability with strategy, capabilities, services, and indicators is weak. |
| 2 | Structured Modeling | The processes are modeled in BPMN, characterized with a file, data is identified and responsible are defined. |
| 3 | Integrated automation | Processes run in Dexon with forms, rules, Related Data, SLAs, roles, offers, and reports. |
| 4 | Traceable architecture | Each process demonstrates a link to strategy, capabilities, services, offerings, data/CIs, and indicators. |
| 5 | Governed Executable Architecture | The organization measures, governs, versions, reapproves, and improves processes with continuous evidence and organizational learning. |

*Figure 23. The Dexon Maturity Model: Five Levels from Basic Documentation to Governed Executable Architecture.*


All Rights Reserved – Dexon Software-2026

## 21.2 Advancement criteria

Maturity is not measured by the number of automated processes. It is measured by the organization's ability to govern its operation from objectives, evidence, and traceability. An organization can have many automated flows and remain immature if it does not control data, indicators, roles, changes, and learning.

Superior maturity is reached when the organization can formulate a goal, design a TO-BE oriented to that goal, implement it in Dexon, measure its compliance, learn from the operation and modify it without breaking its architecture.

For the same reason, the maturity assessment should be based on evidence extracted from the platform—demonstrable traceability in artifacts and their relationships, approval dates and registered owners, indicators linked to real processes, audit trails—and not on declarative self-assessments. An organization is not at the level it claims to be; He is at the level he can demonstrate. This principle turns the maturity model into a verifiable instrument and prepares its evolution towards a formal evaluation scheme.

## 21.3 Anchoring in Reference Maturity Models

The Dexon maturity model is deliberately compatible with established maturity models, so that an organization can position itself without abandoning its existing assessment frameworks. OMG's Business Process Maturity Model defines five levels—initial, managed, standardized, predictable, and innovative—that describe the transformation from inconsistent practices to quantitative and innovative process management (OMG, 2008); Dexon levels 1 to 5 follow an analogous trajectory, with a difference in emphasis: maturity is assessed on the complete executable architecture – traceability, data, indicators and platform governance – and not only on the management of the process as a unit.

Hammer's Process and Enterprise Maturity Model provides a distinction that this model incorporates: the maturity of a process depends on the enablers of the process—design, executors, owner, infrastructure, and metrics—and on the company's capabilities—leadership, culture, expertise, and governance—and no process can sustain a maturity greater than what your company supports (Hammer, 2007). That is why levels 4 and 5 of the Dexon model are not defined by individual process attributes, but by organizational behaviors: demonstrable traceability, change governance, reapproval, and learning.

In the same vein, academic research on BPM maturity identifies six core elements of the discipline: strategic alignment, governance, methods, information technology, people, and culture (Rosemann & vom Brocke, 2015). The Dexon model operationalizes them by requiring, in order to advance to the next level, simultaneous evidence in alignment (traceability chain), governance (owners, versioning, reapproval), method (Objective-Driven Process Design), technology (platform configuration), people (adoption by role) and culture (evidence-based decisions).


All Rights Reserved – Dexon Software-2026

# Chapter 22. Interoperability with ITIL, certified ITSM practices, and artificial intelligence

*The chapter explains how the methodology articulates with ITIL 4, Dexon BPM-certified ITSM practices, and the governed incorporation of artificial intelligence. The relationship with ITIL is established through value focus, current state analysis, iterative improvement, holistic vision, and pre-automation optimization. ITIL defines what practices should achieve, while Objective-Driven Process Design provides a method for designing, mapping, measuring, and governing the processes that implement them. Certified practices are presented as proven patterns that are instantiated within the traceability chain, especially in catalog, requests, service levels, incidents, problems, knowledge, changes, assets, configuration and improvement. AI is subordinated to the objective of the process and requires justification, adequate data, criteria for human intervention, reversibility, auditability and evidence on inputs, outputs and actions. For management, the chapter offers a prudent and strategic position: adopt AI and ITSM without fragmenting the governance model or replacing design, responsibility, and control with isolated technological capabilities.*

## 22.1 The ITIL 4 Service Value System and Methodology

This methodology does not compete with ITIL; it interoperates with it. ITIL 4 organizes service management around a service value system whose core is a value chain—plan, improve, engage, design and transition, source and build, deliver and support—fueled by practices, and governed by guiding principles such as focus on value, start where you are, progress iteratively with feedback, think and work holistically,  and optimize and automate (Axelos, 2019). The correspondence with this methodology is direct: the Dexon traceability chain is an instantiation of "focusing on value" – every artifact exists because of its contribution to a result for a stakeholder; purposeful AS-IS analysis is to "start where you are"; the evidence-based improvement cycle is "iteratively progressing with feedback"; the requirement to design processes, data, rules and indicators as a single system is to "think and work holistically"; And the rule of only automating structured decisions with reliable data is "optimize and automate" in its original formulation, where optimization precedes automation.

In ITIL 4 language, practices are sets of organizational resources designed to perform a job or achieve a goal. This methodology brings to those practices what ITIL deliberately does not prescribe: the design method. ITIL defines what a practice should accomplish; Objective-Driven Process Design defines how to design, trace, measure and govern the processes that implement it on the platform.

## 22.2 Certified practices as a verified functional reference

Dexon BPM's compatibility with this framework is not a manufacturer's claim – it was externally verified. Pink Elephant evaluated Dexon BPM under the PinkVERIFY scheme and certified it as compatible with integrated IT service management in **twelve ITSM processes**, listed in the certification letter of August 5, 2026 with the following official names: AI Capability, IT Asset Management, Change Management, Configuration Management, Incident Management, Knowledge Management,  Performance and Improvement Management, Problem Management,


All Rights Reserved – Dexon Software-2026

Request Management, Service Catalog Management, Service Desk y Service Level Management. The authorization to use the PinkVERIFY logo is valid until August 2, 2027 (Pink Elephant, 2026).
It is worth specifying two points that are often confused. The first is that *Integrated IT Service Management* is the global designation of certification and not one of twelve processes; the general criteria that Pink Elephant publishes as a cross-cutting document (Pink Elephant, 2025) apply to the assessment as a whole and do not constitute a separate certified process. The second is the scope of what the certification demonstrates: PinkVERIFY verifies that a tool meets the minimum functional and terminological criteria of each process; **it does not by itself prove the effectiveness of any particular implementation**, which depends on the design, data, and governance that the organization provides.

Therefore, for this methodology, certified practices are not proven process standards but **externally verified functional and terminological requirements, usable as reference standards** that are instantiated using the method of chapters 8 and 9. The certification guarantees that the platform supports the mechanisms; The method is still responsible for ensuring that the process designed around them meets its objective. Its correspondence with the traceability chain is as follows:

| Group | Certified processes | Link in the chain and chapters |
|---|---|---|
| Value consumption | Service Catalog Management, Request Management, Service Level Management | Services and offers (Chapter 7); indicators and SLAs (Chapter 15) |
| Care and restoration | Service Desk, Incident Management | Requestable processes and exceptional trajectories (chapters 4 and 9) |
| Learning and Root Cause | Problem Management, Knowledge Management | Evidence and continuous improvement (ch. 20) |
| Change and resource control | Change Management, IT Asset Management, Configuration Management | Data/CIs and CMDBs (Chapter 13); Post-Departure Government (Ch. 19) |
| Measurement and improvement | Performance and Improvement Management | Indicators (Chapter 15); continuous improvement (ch. 20); Maturity (ch. 21) |
| AI Capability | AI Capability | AI subordinate to the objective (section 22.3) |

Two criteria deserve explicit mention because they validate methodological decisions in this document. First, the configuration criteria require distinguishing service components from IT assets, allowing them to be linked without confusing them; this is the same distinction that Chapter 13 establishes between the IC as a governable element that supports a service and the asset as a managed resource. Second, the criteria for managing requests require that the catalog


All Rights Reserved – Dexon Software-2026

systematically launch the corresponding workflows; That is the architectural condition of Chapter 7: the offer only materializes value when it is integrated with the process that executes it.

## 22.3 Artificial Intelligence in Goal-Oriented Process Design

The AI capability certification assesses the use of natural language processing, trainable AI and predictive analytics, conversational assistants, and generative AI within the operation of services (Pink Elephant, 2025, 2026). For this methodology, the relevant question is not what AI techniques exist on the platform, but under what design discipline they are incorporated into a process. The answer is a natural extension of already established principles: AI is subordinated to the goal of the process.

Before the rules, it is convenient to establish the principle that orders them, because without it each one seems an isolated precaution: **as algorithmic —deterministic— as possible, as probabilistic as necessary** (Villalobos, 2026). This principle is the natural extension of "architecture before automation" (section 2.1) into the realm of AI: the probabilistic technique is not introduced because it is available, but because the deterministic technique is not sufficient for the stated objective. Section 12.3 operationalizes it by separating the three execution zones of a process—augmented persons, algorithmic zone, and agents—and the R11 constraint makes it verifiable by requiring each activity to declare in which it operates.

This is specified in five rules.

- First, any AI intervention in a process—classifying a request, correlating an incident with problems or changes, predicting an SLA breach, writing a response, summarizing a case—must be justified by its contribution to the objective, constraint, or indicator, with the same standard that Chapter 5 requires justifying each BPMN element.
- Second, the distinction in Chapter 12 between automatable decisions and human decisions is applied more demandingly: AI can decide on its own when the decision is structured, reversible, and auditable; assists, but does not decide, when there is expert judgment, ambiguity or impact that is difficult to reverse, and in that case the design must define what information the person responsible receives and what evidence leaves their decision.
- Third, data quality as a condition of automation (chapter 12) becomes critical: a model trained or fed with incomplete or biased data does not accelerate the process, it industrializes its error.
- Fourth, AI interventions generate evidence: which model intervened, with what input, what output it produced, and what action was taken, so that the process mining and the Chapter 20 audit cover AI behavior as well.
- Fifth, when the intervention is not a model invoked within an activity but an **agent** that executes work with limited autonomy, the design declares three limits: its scope, its scaling criterion and the irreversible actions that it cannot take without human confirmation. An agent without these three declared limits is not a capacity of the process; it is an ungoverned dependency.

Governed in this way, AI capabilities expand every link in the chain: in offerings, conversational


All Rights Reserved – Dexon Software-2026

assistants that capture well-formed requests; in processes, classification and intelligent routing of cases; in the knowledge, recovery and generation of responses from governed articles; in indicators, prediction of defaults before they occur; and in improvement, detection of patterns that no static board reveals. This governance discipline is convergent with emerging AI management systems, which require managing artificial intelligence systems with objectives, roles, impact evaluation, and continuous improvement (ISO/IEC, 2023): the same logic that this methodology applies to processes, extended to the models that participate in them.

# Chapter 23. Methodological conclusion

The Dexon methodology does not simply propose modeling processes or automating tasks. It proposes to build an **executable business architecture**. Its central thesis is that an organization is transformed in a governable way when it can trace its strategy down to capabilities, services, offers, processes, data/CIs and indicators, and when that traceability is executed on a platform capable of operating, measuring, auditing and improving.

Objective-Driven Process Design redefines TO-BE: the future process is not the optimized AS-IS, but the mechanism designed to maximize the fulfillment of a verifiable goal. BPMN is still the formal language of modeling, but the goal governs design, data, rules, automation, measurement, and governance.

Dexon BPM embodies this philosophy by integrating design, workflow, data, ETL, ECM, BI, security, evidence, SLAs, reporting, and governance into a unified architecture. The methodology, therefore, does not end at the release to production. It ends—and begins again—when the organization uses real evidence to continuously learn, redesign, and increase its ability to produce value.
It is convenient to close where it was opened. Chapter 1 named a gap that the architecture discipline has identified and has not resolved: between the map of capabilities and the operating model that must implement them, there is rarely a declared mechanism, so that mapping is left to the discretion of each project and is lost with each team rotation. The answer of this methodology is cumulative and can be stated in one line: the traceability chain declares what should be connected, the metamodel declares how, integrity constraints declare which configurations are invalid, the goal of the process declares what each process exists for, the workspace declares where and who executes it, and the platform converts all those declarations into queries. Mapping ceases to be a criterion and becomes a verifiable structure. That's ultimately the difference between a documented business architecture and an executable business architecture.

Read as a whole, the methodology rests on an integrative thesis: strategic theory explains why capabilities are the unit of analysis; the business architecture explains how traceability is structured; the BPM discipline and goal orientation explain how the process is designed and validated; and the government, the data and maturity explain how the system is sustained over time. None of these bodies of knowledge is new. The contribution of the method is to integrate them into an executable, verifiable and governed practice on Dexon BPM.


All Rights Reserved – Dexon Software-2026

# References


- ABPMP. (2019). *BPM CBOK: Guide to the Business Process Management Common Body of Knowledge* (version 4.0). Association of Business Process Management Professionals.
- Axelos. (2019). *ITIL Foundation: ITIL 4 Edition*. The Stationery Office.
- Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management, 17*(1), 99–120.
- Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The Goal Question Metric approach. In J. J. Marciniak (Ed.), *Encyclopedia of Software Engineering* (pp. 528–532). Wiley.
- Business Architecture Guild. (2026). A Guide to the Business Architecture Body of Knowledge (BIZBOK® Guide), v15. Business Architecture Guild.
- DAMA International. (2017). *DAMA-DMBOK: Data Management Body of Knowledge* (2nd ed.). Technics Publications.
- Davenport, T. H. (1993). Process innovation: Reengineering work through information technology. Harvard Business School Press.
- Deming, W. E. (1986). *Out of the crisis*. MIT Press.
- Doerr, J. (2018). *Measure what matters*. Portfolio/Penguin.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of business process management* (2nd ed.). Springer.
- Hammer, M. (1990). Reengineering work: Don't automate, obliterate. *Harvard Business Review, 68*(4), 104–112.
- Hammer, M. (2007). The process audit. *Harvard Business Review, 85*(4), 111–123.
- Hammer, M., & Champy, J. (1993). Reengineering the corporation: A manifesto for business revolution. HarperBusiness.
- Hiatt, J. M. (2006). ADKAR: A model for change in business, government and our community. Prosci Learning Center Publications.
- IEEE Task Force on Process Mining. (2012). Process mining manifesto. In F. Daniel, K. Barkaoui & S. Dustdar (eds.), *Business Process Management Workshops (BPM 2011), LNBIP 99* (pp. 169–194). Springer.
- ISACA. (2018). COBIT 2019 framework: Introduction and methodology. ISACA.
- ISO/IEC. (2023). ISO/IEC 42001:2023 — Information technology — Artificial intelligence — Management system. International Organization for Standardization.
- ISO/IEC/IEEE. (2017). ISO/IEC/IEEE 15939:2017 — Systems and software engineering — Measurement process. International Organization for Standardization.
- Kaplan, R. S., & Norton, D. P. (1996). *The balanced scorecard: Translating strategy into action*. Harvard Business School Press.
- Kaplan, R. S., & Norton, D. P. (2004). *Strategy maps: Converting intangible assets into tangible outcomes*. Harvard Business School Press.
- Khadem, R., & Khadem, L. (2017). *Total alignment: Tools and tactics for streamlining your organization*. Entrepreneur Press. (Spanish edition: *Total Alignment*, Grupo Editorial Norma).
- Koliadis, G., & Ghose, A. (2006). Relating business process models to goal-oriented requirements models in KAOS. In A. Hoffmann, B. Kang, D. Richards & S. Tsumoto (eds.), *Advances in Knowledge Acquisition and Management (PKAW 2006), LNCS 4303* (pp. 25–39).



All Rights Reserved – Dexon Software-2026

Springer.
- Kotter, J. P. (2012). *Leading change*. Harvard Business Review Press.
- Mintzberg, H. (1987). The strategy concept I: Five Ps for strategy. *California Management Review, 30*(1), 11–24.
- Mintzberg, H., & Waters, J. A. (1985). Of strategies, deliberate and emergent. *Strategic Management Journal, 6*(3), 257–272.
- Moen, R., & Norman, C. (2010). Circling back: Clearing up myths about the Deming cycle and seeing how it keeps evolving. *Quality Progress, 43*(11), 22–28.
- Object Management Group. (2008). Business Process Maturity Model (BPMM), version 1.0. GMOs.
- Object Management Group. (2013). *Business Process Model and Notation (BPMN), version 2.0.2*. GMOs. (Also published as ISO/IEC 19510:2013).
- Object Management Group. (2024). Decision Model and Notation (DMN), version 1.5. GMOs.
- Osterwalder, A., & Pigneur, Y. (2010). *Business model generation*. Wiley.
- Pink Elephant. (2025). PinkVERIFY assessment criteria, v3.0–3.1: General ITSM, AI capability, and management practices for requests, issues, changes, IT assets, configuration, service levels, knowledge, service catalog, performance and improvement, and incidents. Pink Elephant.
- Pink Elephant. (2026). *DEXON BPM — PinkVERIFY® certified* . Pink Elephant.
- Osterwalder, A., Pigneur, Y., Bernarda, G., & Smith, A. (2014). *Value proposition design*. Wiley.
- Porter, M. E. (1996). What is strategy? *Harvard Business Review, 74*(6), 61–78.
- Rosemann, M., & vom Brocke, J. (2015). The six core elements of business process management. In J. vom Brocke & M. Rosemann (Eds.), *Handbook on business process management 1* (2nd ed., pp. 105–122). Springer.
- Ross, J. W., Weill, P., & Robertson, D. C. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business School Press.
- Rummler, G. A., & Brache, A. P. (2012). Improving performance: How to manage the white space on the organization chart (3rd ed.). Jossey-Bass.
- Salazar Yusti, A. (2021). The Emerging Strategy: And the Death of the Strategic Plan.
- Teece, D. J. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal, 28*(13), 1319–1350.
- Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal, 18*(7), 509–533.
- The Open Group. (2022a). *The TOGAF® Standard, 10th Edition*. The Open Group.
- The Open Group. (2022b). TOGAF® Series Guide: Business capabilities, version 2. The Open Group.
- The Open Group. (2022c). *TOGAF® Series Guide: Value streams*. The Open Group.
- The Open Group. (2023). *ArchiMate® 3.2 Specification*. The Open Group.
- van der Aalst, W. M. P. (2016). *Process mining: Data science in action* (2nd ed.). Springer.
- van Lamsweerde, A. (2009). Requirements engineering: From system goals to UML models to software specifications. Wiley.


All Rights Reserved – Dexon Software-2026

- Villalobos, J. (2019). *The Business Model* (Readings for Business Architects, No. 2). Self-published. (Available on ResearchGate and in Kindle edition).
- Villalobos, J. (2026). *Architecture as a Tool: Connecting Business and Technology* [Presentation]. Latam Architecture Day 2026. DOQMA Consulting / Universidad de los Andes.
- Villalobos, J. (n.d.-a). *Business Strategic Analysis* (Readings for Business Architects). Self-published.
- Villalobos, J. (n.d.-b). *The Business Capabilities Map* (Readings for Business Architects). Self-published.
- Villalobos, J. (n.d.-c). *The Strategic Model* (Readings for Business Architects). Self-published.
- vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., & Viaene, S. (2014). Ten principles of good business process management. *Business Process Management Journal, 20*(4), 530–548.



All Rights Reserved – Dexon Software-2026