

MANUFACTURING EXECUTION SYSTEMS

A PRODUCTION DRIVEN PERSPECTIVE ON MES ARCHITECTURE IN STEELMAKING

Production Logic Before Software

PRODUCTION LOGIC FIRST

MES architecture must prioritize industrial production realities over standard software features.

THE PARADIGM CHALLENGE

Complex MES design stems from managing batch, continuous, and discrete production transitions.

CORE ELEMENT STRUCTURE

Sustainable MES systems rely on stable operational questions rather than vendor software modules.



FOREWORD

Manufacturing Execution Systems in the metals industry are routinely evaluated on product maturity, module completeness, and reference customer lists. This perspective places the inherent structure of industrial production at the center of MES architecture decisions, specifically the coexistence of batch, continuous, and discrete production paradigms within a single value chain, and the structural transformations that occur at the transitions between them.

MES complexity is a production logic problem, not a software problem. An architecture that begins with production logic will outlast one that begins with software modules.

A production-driven framework organized around three design principles and four Core Elements follows from this perspective. The design principles, a Proven Core, Process Modelling, and Architectural Discipline, translate the three competing pressures of security, process fit, and manageability into concrete architectural decisions. The four Core Elements, covering integration and platform readiness, released production intent, operational reality and production history, and quality compliance and traceability, provide a stable structural framework that applies across the full steel value chain while remaining specific to the production logic of each plant and process area.

A sustainable MES architecture must be able to carry the logic of industrial production faithfully and adaptably across the decades of operational life that a serious MES investment represents. The data model and architectural structure are what determine whether that is possible.

Jan Piskernik

Head of Software Delivery at qoncept technology GmbH

PIONEERING TECHNOLOGIES

MAKING INTELLIGENT PRODUCTION A REALITY

Software and engineering solutions that increase process efficiency, optimize raw material use, and increase operator safety in steelmaking.

GLOBAL FOOTPRINT

15+

Software
Projects

5

Countries
Software

60+

Equipment
Projects

19

Countries
Equipment

40+

Consulting
Projects

9

Countries
Consulting

PRODUCT PORTFOLIO

SOFTWARE & HARDWARE

qontrol MAPS

Raw material management & charge optimization

qontrol L2

Level 2 automation & digital twins

qontrol MES

Manufacturing execution system

qontrol LMS

Ladle management system

curve AI

Operator terminals & camera systems

EQUIPMENT & DEVICES

qouplerG

Automatic gas coupling system

qouplerM

Patented multi-media coupler

qouplerD

Automatic slide gate coupling, globally unique

Bins & Silos

Complete industrial silo systems

Engineering Services

Custom mechanical design & revamps

CONSULTING & STUDIES

Metallurgical Process Consulting

Measurable improvements in melt shop operations

Digitalization & Software Strategy

Feasibility studies and transformation roadmaps

COMPANY HISTORY

2018



FOUNDATION

Founded in Leoben, Austria, by metallurgists with strong industrial backgrounds.

Establishment of the internal software architecture creating the scalable foundation for modular qontrol integration.

2020



MARKET READINESS – qouplerM & FIRST PRODUCT DEPLOYMENT

qouplerM, an automatic multimedia coupler, reached market readiness and was patented.

Implementation of the first productive scrap tracking system, marking the transition to operational digital steel production.

2022



PATENT AND FULL STEELMAKING INTEGRATION

Patent granted for the automatic slide gate coupling system qouplerD. This expanded the portfolio into automated slide gate actuation and signal integration.

Implementation of the first productive steel plan management system (Level 2) on full scope.

2024



NORTH AMERICAN EXPANSION & PLATFORM SCALING

Establishment of qoncept Americas Inc. to support North American customers.

Functional specialization of the qontrol platform with dedicated modules for heat optimization, ladle tracking, and advanced scrap yard management.

TODAY



100+ PROJECTS WORLDWIDE

Active operations across Europe, the Americas, and Asia.

CONTENTS



03

THE MISMATCH BETWEEN
PROCUREMENT CRITERIA
AND OPERATIONAL
REQUIREMENTS
PAGE 5



04

ARCHITECTURAL
DEGRADATION
THROUGH
CUSTOMIZATION
PAGE 10



06

TRANSLATING DESIGN
PRINCIPLES
INTO SOFTWARE
ARCHITECTURE
PAGE 14

01 INTRODUCTION 5

How quick fixes and workarounds quietly destroy long-term system maintainability.

02 PRODUCTION LOGIC AS THE FOUNDATION OF MES ARCHITECTURE 6

Mastering the complex leaps between batch, continuous, and discrete manufacturing.

03 THE MISMATCH BETWEEN PROCUREMENT CRITERIA AND OPERATIONAL REQUIREMENTS..... 8

The dangerous gap between boardroom expectations and harsh plant-floor realities.

04 ARCHITECTURAL DEGRADATION THROUGH CUSTOMIZATION 10

From smart configuration to unmaintainable chaos: the hidden path of system decline.

05 BALANCING SECURITY, PROCESS FIT, AND MANAGEABILITY 12

The delicate tightrope walk between security, process fit, and future flexibility.

06 TRANSLATING DESIGN PRINCIPLES INTO SOFTWARE ARCHITECTURE 14

Building bulletproof systems with a proven core and explicit process modeling.

07 A PRODUCTION-LOGIC-DRIVEN MES STRUCTURE 16

Switching from rigid vendor modules to universal, adaptable operational questions.

08 CONCLUSION 18

The ultimate blueprint for durable, future-proof MES architecture in metals manufacturing.



01 INTRODUCTION

Discussions about MES in steelmaking and metals manufacturing tend to follow a familiar pattern. Vendors present module overviews, feature lists, and screens. Buyers evaluate reference customers, implementation timelines, and product maturity. The conversation is organized around software, and the implicit assumption is that a sufficiently complete and proven product will, with appropriate effort, map onto the production environment of the buyer.

This assumption deserves closer examination. The challenge in industrial MES implementation is rarely that the software lacks a particular feature. More commonly, the software's underlying data model and process logic do not reflect how production actually runs. A feature can be added or configured. A data model that does not carry the right production concepts creates constraints that propagate through every layer of the system over time.

The consequences of this misconception are well known to practitioners, even if they are rarely named directly. Projects close successfully, systems go live, and the real difficulty appears when the first serious process change meets a system that was adapted

under time pressure rather than designed with production logic at its center. By that point, the gap between product logic and production reality has been filled with configuration decisions, custom code, and operational workarounds, and the accumulated weight of those decisions has made the system both harder to change and harder to understand.

This article develops a production-driven perspective on MES architecture. It begins with the production paradigms themselves, because that is where MES complexity originates. It then examines the structural tension between what buyers seek in an MES procurement and what their production environment actually demands. From there, it traces the degradation path that follows when this tension is not resolved architecturally. The paper concludes with a framework of design principles and a production-logic-driven MES structure that treats the stability of operational questions, not the completeness of software modules, as the organizing principle of MES architecture.

Traditional MES implementations in steel and metals manufacturing often fail because they focus on software features rather than aligning the system's architecture and data models with actual production realities.



02 PRODUCTION LOGIC AS THE FOUNDATION OF MES ARCHITECTURE

Any serious MES discussion must begin with the way production actually runs. In most industrial environments, and in steelmaking in particular, production does not follow a single, uniform logic. Multiple inputs may be combined into a batch. That batch may be processed as one production object through one or several process steps. The process may then transition into a continuous material flow. At the end of the value chain, the output becomes discrete

units that need to be counted, inspected, sorted, packaged, and traced individually. These three paradigms, batch, continuous, and discrete, do not appear in isolation. They appear in sequence, and the transitions between them are precisely where MES complexity concentrates. **Figure 1** illustrates this coexistence of paradigms and the transition points between them.

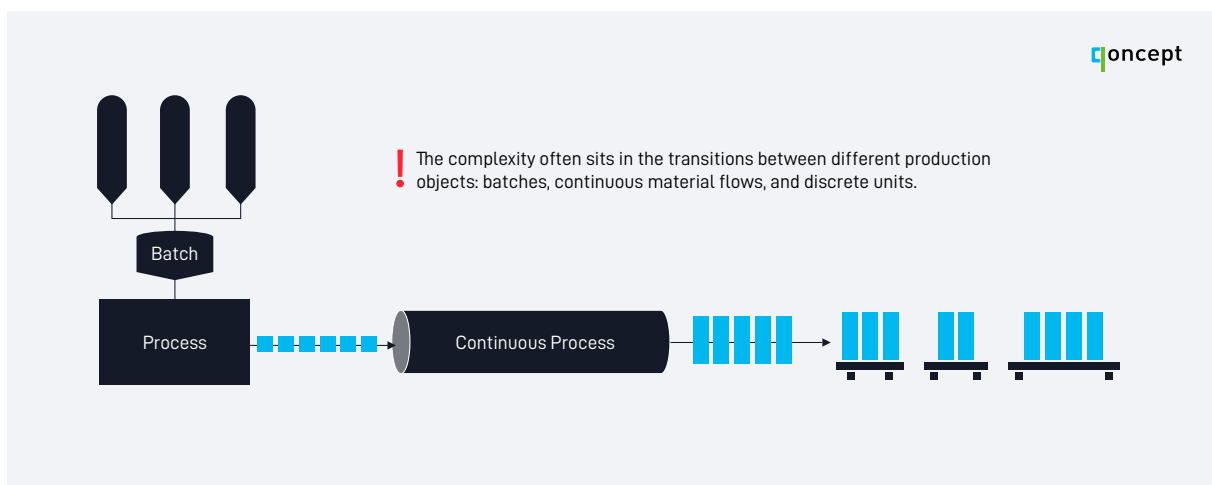


Figure 1: Production Paradigms and Transitions



Data structure is the decisive factor here. Each paradigm carries a fundamentally different set of attributes. A batch carries composition, origin, recipe, genealogy, and quality context. A continuous material flow carries time windows, process states, sensor measurements, and material balance. A discrete unit carries countability, sequence position, inspection results, handling events, and packaging relationships. When production moves from one paradigm to another, the MES must perform a structural transformation of the production object, not merely a status update.

For an MES, the decisive test is whether the underlying data model can represent the relationship between a batch, the continuous material that derived from it, and the discrete units that were cut from that material, while preserving quality context, genealogy, and traceability across all three. A feature list can show what an MES does. The data model reveals whether it can carry the logic of the production process, and these are not the same question.

This distinction matters because data model gaps do not announce themselves at go-live. A system that approximates batch-to-unit

relationships at implementation will produce workarounds in quality traceability, gaps in production history, and fragile interfaces whenever the production process changes. In steelmaking, where a customer order may trace back through multiple heats, casts, rolling campaigns, and inspection events, the integrity of that chain carries both operational and often regulatory weight.

The implication for MES architecture is direct. Production paradigms and their transitions must be first-class concepts in the data model, not afterthoughts accommodated through configuration. Any MES architecture that does not begin with this requirement is building on an unstable foundation, regardless of how complete its feature set appears at the time of procurement.

Understanding why production paradigms create structural complexity in the data model is the first step. The harder question is why so many MES procurement processes fail to treat this complexity as the primary evaluation criterion.

An effective MES architecture for metals manufacturing needs to treat the transitions between batch, continuous, and discrete production paradigms as first-class concepts in its data model to ensure true traceability and adaptability.



03 THE MISMATCH BETWEEN PROCUREMENT CRITERIA AND OPERATIONAL REQUIREMENTS

If production logic is the primary design driver, the natural question is why so many MES evaluations fail to treat it as such. The answer lies in the structure of the procurement process itself, and in a hidden assumption that connects buyer expectations to vendor positioning.

When companies evaluate an MES, they are managing genuine organizational pressures. A proven product with visible modules and a strong reference list creates confidence in the boardroom and makes the investment defensible within the organization. A fast and predictable rollout timeline reduces project risk and internal resource commitment. Industry references provide the

social proof that closes internal arguments. These are not irrational criteria but legitimate responses to real organizational constraints.

At the same time, the production environment has its own requirements. It has specific equipment sequences, material flows, and production variants that do not exist in any generic template. It has quality release and hold logic that reflects years of operational learning. It has interfaces to ERP, automation, laboratory, and logistics systems that are connected in ways specific to how the organization actually works. It also has a fundamental requirement that the system remain adaptable when processes change, without triggering a vendor project every time.

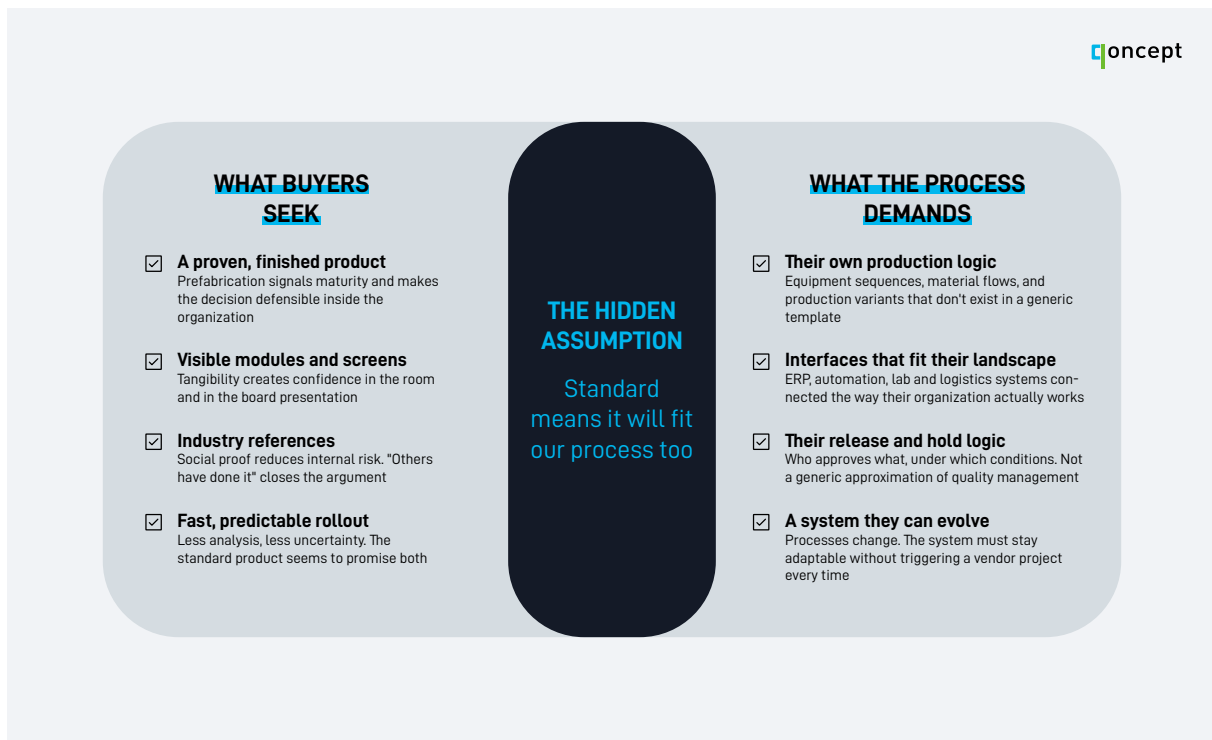
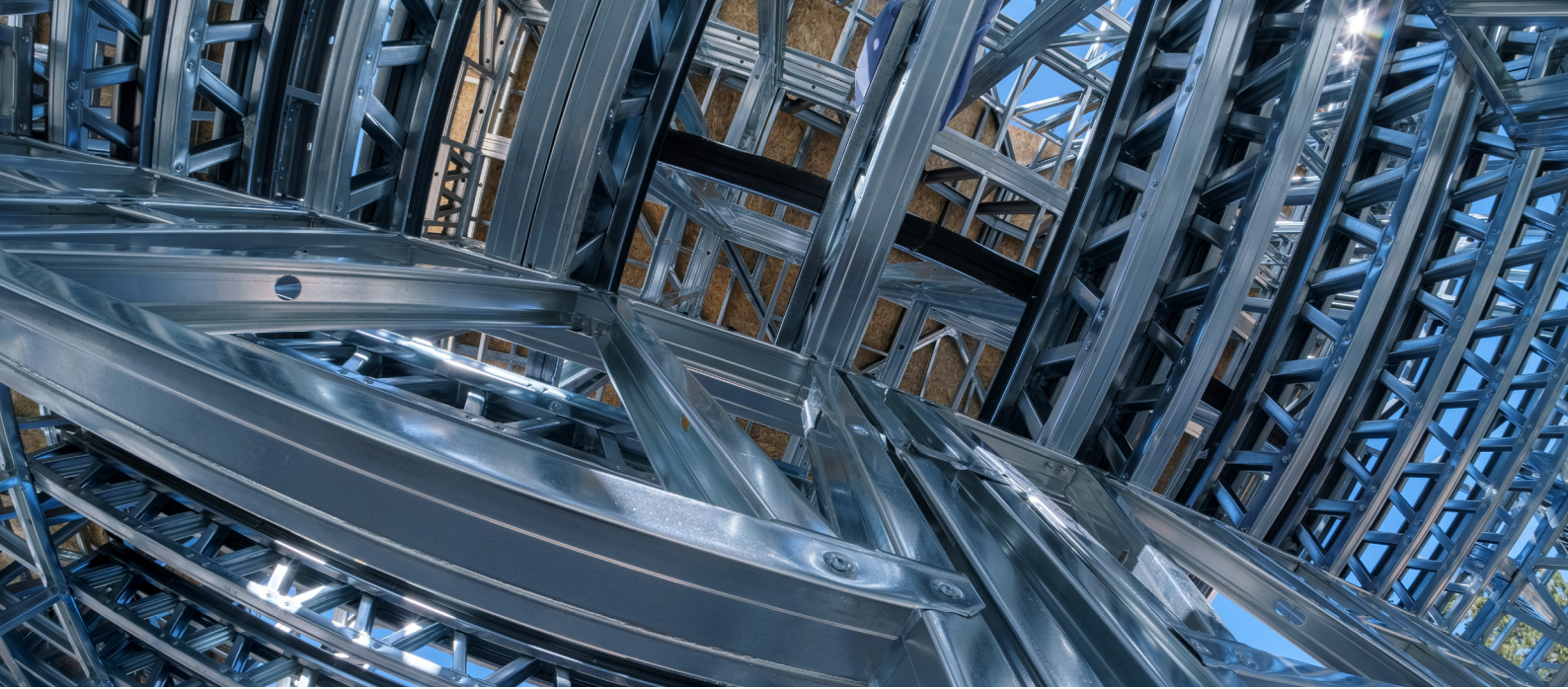


Figure 2: Buyer Criteria and Process Demands

Figure 2 maps both sides of this tension as they appear in practice, with the hidden assumption that connects them placed at the center. The left side reflects the organizational logic of procurement. The right side reflects the operational logic of production. The hidden assumption in the center is what makes them appear compatible when they may not be. Buyers implicitly assume that a proven, standard product will fit their specific process, because the vendor's references suggest it has fit similar processes before. This assumption is structurally unreliable, since references demonstrate that a product has been deployed in a given industry but say nothing about whether its data model can

carry the specific production logic of the next customer without significant adaptation.

The gap between these two sides does not remain empty. It gets filled, first with configuration, then with customization, then with workarounds. The manner in which it gets filled determines the long-term architectural fate of the system. When that assumption goes untested, the gap between procurement logic and production reality does not disappear. It accumulates, stage by stage, inside the architecture of the system itself.



04 ARCHITECTURAL DEGRADATION THROUGH CUSTOMIZATION

The path from procurement to operational complexity is rarely dramatic. It proceeds in stages, each of which appears locally reasonable, and each of which deposits a layer of technical debt that becomes visible only later.

The first stage is configuration. The standard product is adjusted within the parameters its architecture supports. This is legitimate and expected. Every MES deployment involves configuration, and a well-designed product can absorb a significant amount of it without losing structural integrity.

The second stage begins when production requirements exceed what configuration can accommodate. Custom logic is added: additional fields, specialized screens, extended workflows, and custom reports. The product is formally still standard software, but it is beginning to diverge from the vendor's baseline. Updates require testing against the custom layer, and every modification carries a small additional coordination cost.

The third stage is workarounds. When a production concept cannot be cleanly expressed within the product's data model and process logic, it has to be handled outside that model. Operators

develop informal procedures. Data is entered in fields designed for other purposes. Quality decisions are made in spreadsheets and transcribed into the system. Processes are circumvented rather than modeled. At this stage, the system still works operationally, but only because people remember why it works, and the institutional knowledge required to operate the system has grown faster than its documentation.

The fourth stage is architectural instability. The system has grown historically and project-by-project rather than according to a maintained structural principle. Updates become unpredictable in their consequences, because the dependencies between the standard layer, the custom layer, and the workaround layer are not formally documented, and no single person fully understands the system. **Figure 3** illustrates this degradation path across its five stages, from a clean standard product to an unmaintainable architecture.

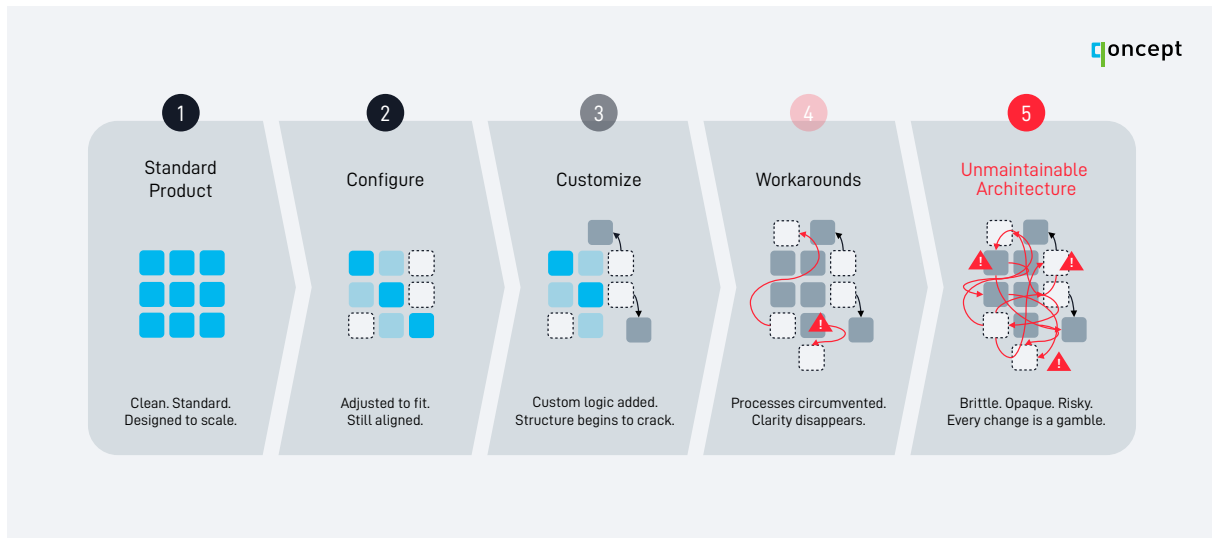


Figure 3: The Gradual Degradation Path

The figure makes visible what is easy to miss in practice: each stage looks manageable in isolation, but the cumulative effect is an architecture that has lost both transparency and adaptability. Four structural consequences accumulate along the way:

- **Loss of standardization:** The product remains formally standard software, but the accumulated customization has made every update a risk. What began as a proven baseline has quietly become something no vendor fully supports.
- **Growing complexity:** Every workaround creates dependencies. Later changes cost exponentially more effort, and the system begins to resist the very operational improvements it was supposed to enable.
- **Workaround culture:** Processes are circumvented rather than modeled. Institutional memory substitutes for documented logic, and eventually no single person fully understands why the system behaves the way it does.
- **Architectural instability:** The system has grown historically and project-by-project rather than according to a maintained structural principle. Long-term maintainability erodes silently until a routine change becomes a crisis.

The degradation described here follows structurally from the mismatch identified in Section 3. Every production requirement that cannot be expressed cleanly in the product's data model has to be handled somewhere, and the accumulation of those handling decisions produces exactly the architecture described above. The question is therefore not whether customization should be avoided, but whether the architecture was designed from the outset to absorb process-specific requirements in a controlled and maintainable way. That design question points directly to the three competing pressures that every MES architecture must hold in balance.



05 BALANCING SECURITY, PROCESS FIT, AND MANAGEABILITY

Understanding why MES projects degrade is useful. Understanding the structural forces that make degradation the default outcome, when those forces are not explicitly managed, is more useful still. In the following, three competing pressures are described that shape every MES decision, whether in procurement, implementation, or evolution. **Figure 4** presents these three pressures as a structural triangle, with MES architecture at the center required to hold all three simultaneously.

Security is the first pressure. The system must feel proven, credible, and defensible. References, vendor maturity, product stability, and a recognizable implementation methodology all contribute to security. For an industrial operation investing in a system that will run production for ten or fifteen years, the question of whether the vendor and the product will still be available and supported in year eight is entirely legitimate.

Process fit is the second pressure. The system must reflect how production actually runs. In steelmaking, this means heat tracking, material genealogy, quality release logic, equipment sequences, casting and rolling relationships, interface dependencies, and the specific operational patterns built up over years. Approximate fit is rarely sufficient. A heat tracking workaround can become an audit

issue, and a quality release exception that bypasses the system may work operationally only as long as someone remembers why it exists and manually ensures the outcome.

Manageability is the third pressure. The system must remain understandable and changeable as the production environment evolves. New grades, new equipment, new interfaces, new reporting requirements, process modifications, and plant expansions are not exceptional events in a steelmaking operation. They are the normal cadence of industrial life. An MES that cannot absorb these changes without a major project every time is a liability rather than a platform for operational improvement.

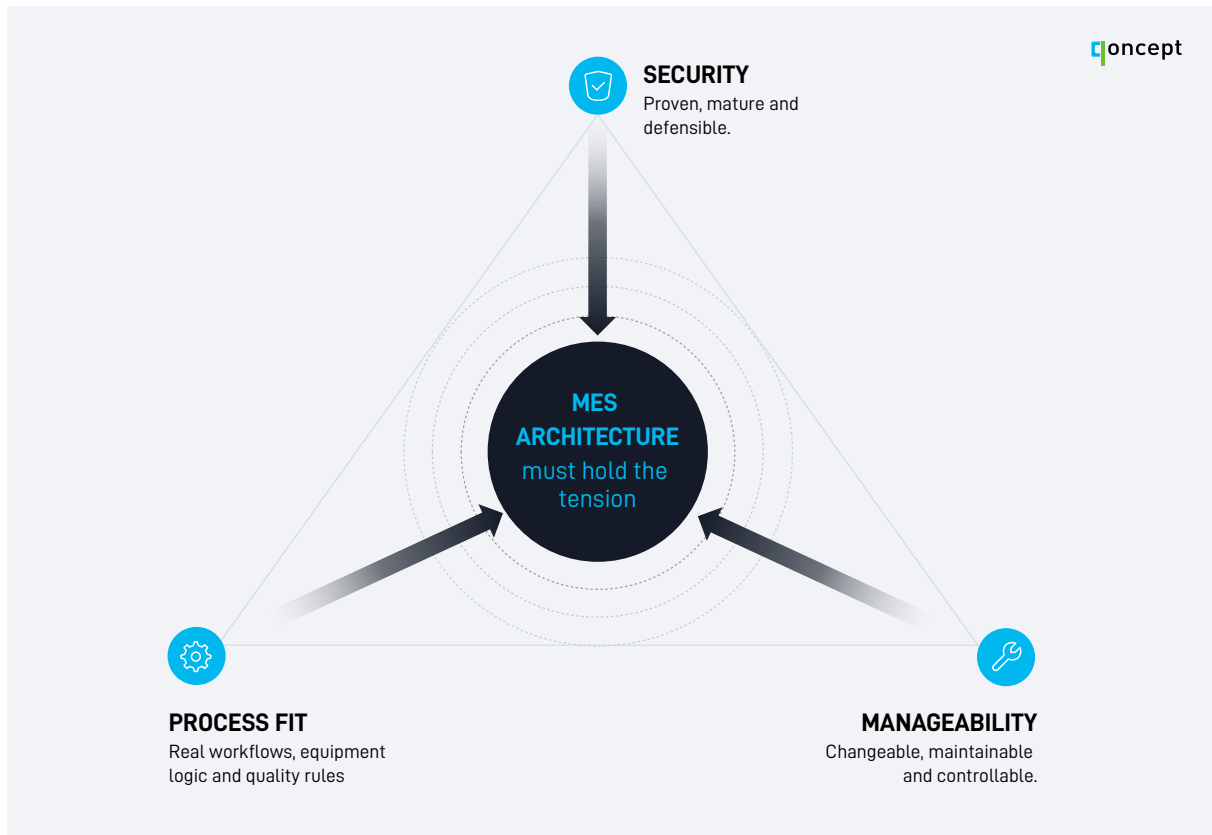


Figure 4: The Structural Tension Triangle

The figure illustrates why product selection alone cannot resolve this tension. Each corner of the triangle pulls in a direction that creates friction with the other two. A very finished, mature product tends to create security but may be difficult to adapt cleanly to specific process requirements. A highly process-specific solution creates fit but may become harder to manage as it diverges from its original structure. A highly flexible, open architecture may maximize long-term manageability but may look insufficiently proven to an evaluation committee at procurement time.

The tension is a structural property of the problem, and it will be present in every serious MES decision regardless of which product is chosen. What matters is whether the architecture was designed to hold that tension productively, or whether it was left to resolve itself through the degradation path described in Section 4. The answer to that question lies not in procurement criteria but in the design principles that guide how the architecture is built. Those principles are the subject of the next section.



06 TRANSLATING DESIGN PRINCIPLES INTO SOFTWARE ARCHITECTURE

The three pressures described in Section 5 do not resolve themselves. They require an explicit architectural response, and that response must be built into the system from the outset rather than managed through project governance after the fact. Three design principles follow directly from the analysis above, each addressing one dimension of the tension while remaining coherent with the other two. **Figure 5** shows how these three principles map from the problem space defined by the tension triangle to the solution space of MES architecture.

The first principle is a Proven Core. The stable foundation of any MES deployment should consist of validated data models, established process logic, and tested integration patterns proven across multiple industrial environments. Security is created here through the reliability of the underlying models and patterns that carry production data, rather than through the completeness of a software product catalog. A Proven Core reduces implementation risk by ensuring that what the system does is built on a foundation that has already demonstrated its stability under real industrial conditions.

The second principle is Process Modelling. The architecture must provide a structured layer in which specific workflows, equipment logic, quality rules, interface dependencies, and operational

patterns can be expressed explicitly as first-class architectural elements. Process fit is created here. The production reality of the plant is modelled within the architecture rather than approximated through configuration or bypassed through workarounds. A process concept that is modelled explicitly can be understood, documented, tested, and modified, while a process concept handled through a workaround exists only in the memory of the people who built it.

The third principle is Architectural Discipline. A clear and maintained separation between the stable Proven Core and the process-specific layer is what makes the system manageable over time. This means that changes to process-specific logic do not require modifications to the core, and that core updates do not break process-specific configurations. This separation does not happen automatically. It requires deliberate design decisions and ongoing discipline in how the system is extended, and it is the only structural mechanism that keeps the architecture controllable as both the software and the production environment evolve.

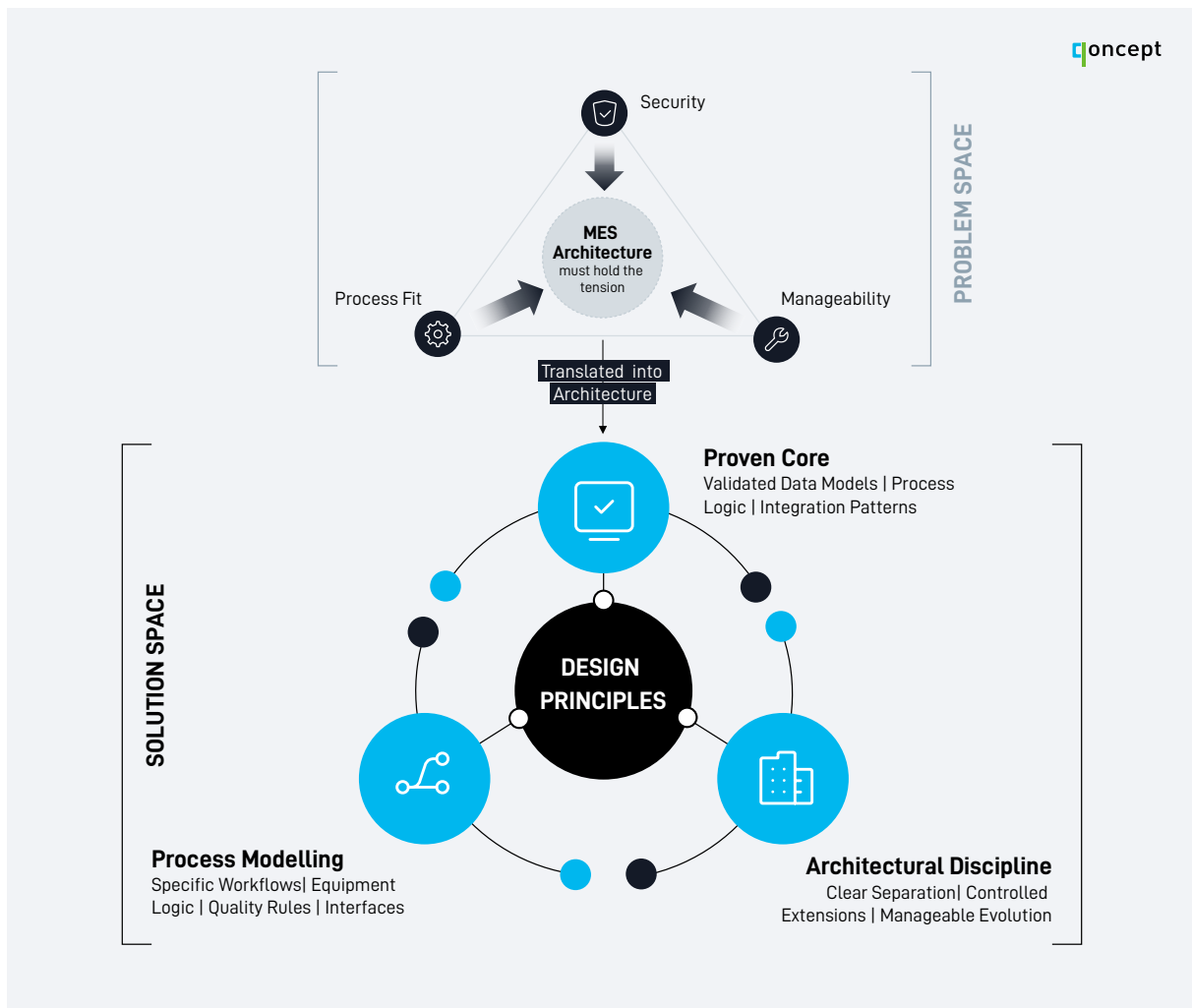


Figure 5: Design Principles as Architectural Translation

The figure makes the mapping explicit. Security, process fit, and manageability are properties of the problem space. Proven Core, Process Modelling, and Architectural Discipline are their counterparts in the solution space. Each principle addresses one pressure directly while supporting the other two. A Proven Core provides the security that comes from validated foundations. Process Modelling provides the fit that comes from explicit architectural representation of production logic. Architectural Discipline provides the manageability that comes from maintaining a clear separation between what is stable and what is process-specific.

Together, these principles reframe what an MES evaluation should look for, what an implementation should prioritize, and what an evolution strategy should protect. They do not, however, answer the question of how this architecture should be structured in practice. That requires a concrete organizational framework, one that derives its structure from production logic rather than from software packaging. That framework is the subject of the next section.



07 A PRODUCTION-LOGIC-DRIVEN MES STRUCTURE

The design principles described in Section 6 require operationalization. Principles without structure remain intentions. The three principles translate into a concrete MES architecture when they are organized around the questions that every production operation must answer, rather than around the modules that software vendors have found convenient to package.

Most MES structures start with software modules. The module logic is functional: it describes what the software does. Modules are useful for packaging and licensing software functionality, but they are an unreliable guide to MES structure in complex industrial environments. Module boundaries reflect vendor organization and product history rather than production logic. Two plants with identical process requirements may find that the same module covers fifty percent of their needs and leaves the other fifty percent in the gap between modules, to be addressed through configuration, customization, or workarounds.

A more stable starting point is the set of operational questions that remain constant across the steel value chain, even as the production objects, process steps, and equipment configurations change from plant to plant and from area to area within a plant. Four such questions, organized as Core Elements, provide a production-logic-driven MES structure applicable from the melt shop through rolling and finishing to shipment preparation. **Figure 6** illustrates these four Core Elements as a pyramid structure applied across the three primary zones of the steel value chain: upstream, downstream, and finishing.

Core Element Zero addresses integration, data, and platform readiness. This element comes first because everything else depends on it. Master data management, data models, security, event handling, interface architecture, and the foundational capabilities for AI integration and digital twin readiness all belong here. This layer determines whether operational data can be connected, governed, and evolved over time. A system that treats this layer as infrastructure rather than architecture will accumulate structural debt in its data foundations that is at least as damaging as the process logic debt described in Section 4.

Core Element A addresses released production intent and execution logic. Every production step requires a clear released intent: what should be produced, in what sequence, with which materials, consumables, equipment, and resources, and according to which execution rules. This element covers work order management, production scheduling at the execution level, resource assignment, and the logic that translates released production plans into actionable instructions for operators and automation systems. In steelmaking, this includes heat schedules, casting sequences, rolling programs, and the quality-conditional branching that determines how a production object should be treated based on its current state.

Core Element B addresses operational reality and production history. This element covers what is happening now and what actually happened. It provides live visibility across the changing production objects, from customer order to heat, from cast strand to slab, from slab to coil, and from coil to finished unit. The same data must remain retrievable, auditable, and structurally complete whenever production history, quality decisions, or traceability are reviewed. In practice, this means the system must maintain the identity and genealogy of production objects across paradigm transitions, preserving the relationships that connect a finished product back to its raw material inputs, process history, and quality decisions at every stage.

Core Element C addresses quality, compliance, and traceability. This element answers the question of whether the production result was good enough. At each stage of the value chain, this involves evaluating the production object against the relevant specification for its current state: chemistry in the melt shop, dimensions and surface quality in rolling and finishing, mechanical properties and customer-specific requirements before shipment. The outcome is a documented quality decision covering release, hold, downgrade, rework, or rejection. This decision must be traceable, auditable, and connected to the production history that justified it.

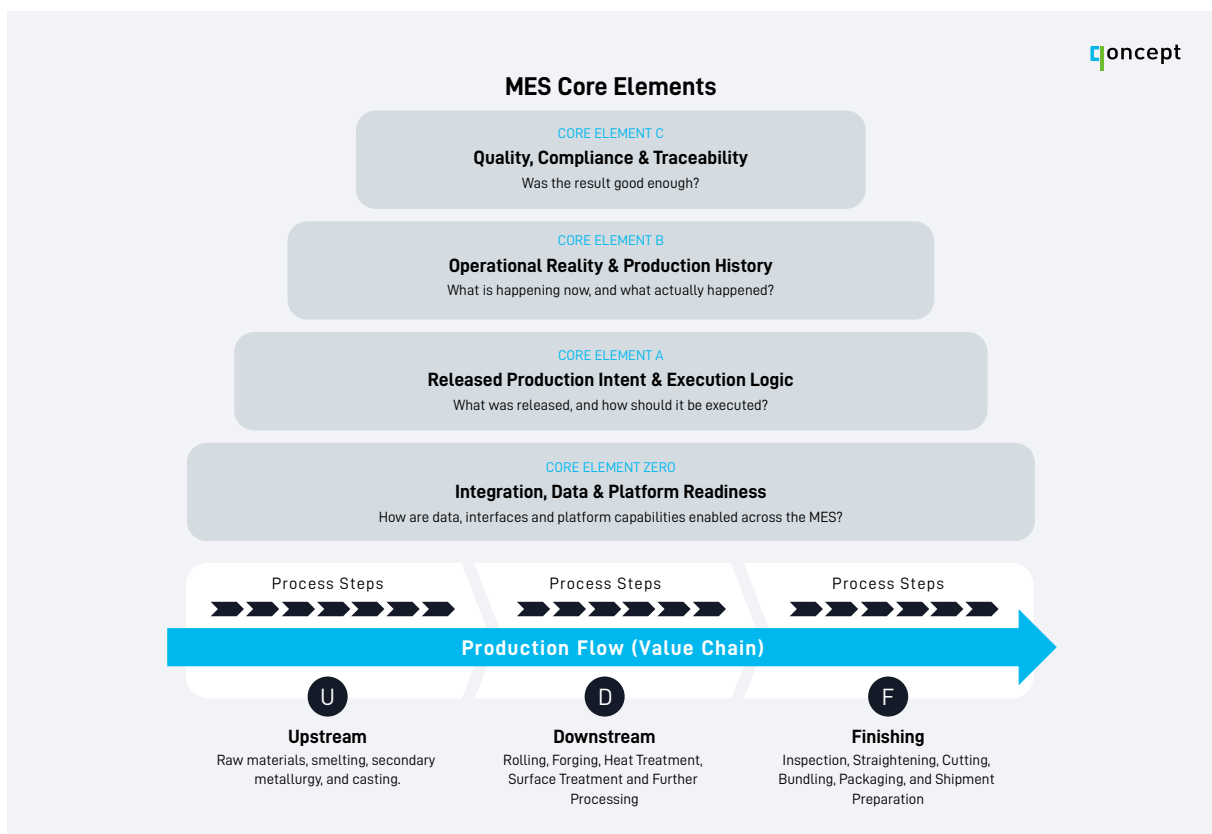


Figure 6: Core Elements Across the Steel Value Chain

The figure illustrates a property of this structure that is easy to state but important to internalize. The Core Elements remain stable across the entire value chain while their content changes with the production context. In the melt shop, Core Element B may be primarily heat-based. In a rolling mill, it may be unit-, order-, or campaign-based. In a finishing and inspection line, it may be driven by routing decisions, inspection results, rework loops, and release workflows. The operational questions are the same. The production logic that answers them is specific to the area and the plant.

This stability is what distinguishes a Core Element structure from a module structure. Modules describe what software does. Core

Elements describe what production needs. When production needs change, a Core Element structure provides a stable framework for extending and adapting the system without destabilizing the architecture. The production logic changes within the element, while the element itself remains a coherent, documented, and maintainable part of the architecture. That distinction is the practical expression of the design principles developed in Section 6, and it is the foundation on which the conclusions of this paper rest.



08 CONCLUSION

The argument developed in this paper can be stated directly.

MES complexity in steelmaking and metals manufacturing originates in the structure of industrial production, and an MES architecture that begins with production logic will create more durable value than one that begins with software modules and adapts them toward process requirements under project pressure.

This conclusion has practical implications for how MES decisions are made. Procurement evaluations that focus primarily on product maturity, module completeness, and reference customer lists are measuring how well the vendor has packaged software functionality, not how well the system's data model and process logic can carry the specific production reality of the buyer. The gap between these two measures is where most MES projects encounter their deepest difficulties, typically not at go-live but in the years that follow, when the accumulated weight of configuration decisions, custom code, and operational workarounds begins to constrain the system's ability to evolve.

The framework presented in this paper, three design principles translated into four Core Elements, addresses this gap directly. A Proven Core provides the security that comes from validated foundations rather than product completeness. Process Modelling provides the fit that comes from explicit architectural representation of production logic rather than from customization depth. Architectural Discipline provides the manageability that

comes from a maintained structural separation between what is stable and what is process-specific. Together, these principles give the architecture a way to hold the tension between security, process fit, and manageability without degrading into the configuration and workaround layers described in Section 4.

The four Core Elements operationalize these principles across the full steel value chain. Core Element Zero establishes the data and integration foundation on which everything else depends. Core Element A carries the released production intent and execution logic that every process step requires. Core Element B maintains the operational reality and production history that connect the present state of production to its complete record. Core Element C documents the quality decisions that determine whether each production result meets the requirements of its next destination in the value chain. Each element addresses a question that production must answer regardless of plant configuration, process area, or product mix. The questions remain stable. The production logic that answers them is specific to the plant and the process.



challenge US

qoncept group

Waasenstraße 16
8700 Leoben, Austria

(+43) 3842 30 300
office@qoncept.at
www.qoncept.at



qoncept Americas Inc.

100 Corporate Center Drive
Coraopolis, PA 15108
United States of America

(+1) 412 286 9890
office@qoncept.us