

---

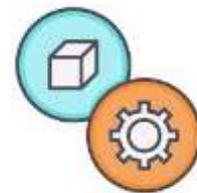
## White Paper

---

# Configuration and Customization in MES and LIMS: A Practical Guide

## Introduction

In today's digital manufacturing and laboratory environments, organizations adopt MES (Manufacturing Execution Systems) and LIMS (Laboratory Information Management Systems) to gain efficiency, ensure compliance, and foster traceability. However, no off-the-shelf solution fits all. As a result, configuration and customization are both necessary and normal.



This white paper explores the distinctions, challenges, and best practices in configuring and customizing MES and LIMS systems. It reflects industry insights and includes a comparative opinion on leading vendors, especially regarding their capabilities and limitations in configurability and customization.

Common challenges across MES and LIMS relate to the balance between efficiency, compliance, and the specific demands of manufacturing or laboratory workflows. Understanding these helps organizations better plan their system adaptations.

Commonality between MES and LIMS lies in the need for efficiency, regulatory compliance, and robust traceability. Yet each system faces unique challenges: MES solutions contend with production variability and automation integration, while LIMS handle diverse lab workflows and complex test plans.

The author of this paper, Gerard Ipskamp ([gerard.ipskamp@mom-institute.org](mailto:gerard.ipskamp@mom-institute.org)) works at MOMi as a Senior Business Consultant. MOMi is a MESA partner and an independent expert division of ATS Global, offering education and business consultancy in the domain of Manufacturing Operations Management, for companies on their path to digital manufacturing.

# What is configuration?

The process of adjusting the software's settings and options to suit specific requirements without altering the underlying code, is considered as configuration. Configuration uses tools and interfaces provided by the vendor. Examples include: defining workflows, setting user roles, adjusting permissions, or selecting specific modules to activate.

Configuration is done with predefined, pre-tested and documented structures. Making choices between predefined and valid, documented paths. A configured solution that gets a version upgrade that is backwards compatible does not require testing.

Configuration only requires product skills: configuration options can be taught by the product vendor. Upload of (configuration-) data (via Excel) to the product, or manual configuration in the product is both considered as configuration.

Not configuration is making changes to the software's code or adding new functionality to meet unique business requirements that the Out-Of-The-Box features and configuration options cannot address. Developing custom scripts, plugins, or modules that need definition, development, testing and documentation. This work is performed by developers or consultants with technical expertise and normally not by application specialists. Technical (coding) skills are needed.

Use of APIs: everything out of the software that triggers and processes an API is not configuration. Work in an IDE (Visual Studio) or on database or OS level is not considered configuration.

Examples of configuration:

- Configuration of authorization roles: choose between predefined roles (e.g., QA inspector vs maintenance technician) or configure your own roles with a tool in the product.

Not configuration: define roles and implement them outside of these paths.

- Configuration of a workflow for event handling: use predefined workflows or configure your own workflow with a tool in the product.

Not configuration: define and manage workflows outside of the product.

- Configuration of User Interface: using options in the software to change the user interface or user interface components or use a tool in the product to arrange and connect UI components.

Not configuration: use tools outside the product or components outside of the product.

- Configuration of data collection: use the structures and definitions in the product to pick up data, process it, and give it context or trigger events.

Not configuration: processing collected data outside the product for formatting, validation, structure, or storage is not considered configuration.

# Managing customizations

It is very normal to have some customizations because no Out-of-the-Box solution fits perfectly for your unique processes. Customizations allow you to extend functionality where configuration options end. However, managing customizations well is key to maintaining system integrity and compliance.

Ownership aspects must be clear upfront: who owns the customizations? Possible scenarios include vendor ownership if they develop it, or customer ownership if developed internally or by a partner under contract. This clarity is important for support and future upgrades. It would not be the first time to have a software vendor announcing an upgrade of their product, and telling the customer that there will be additional work attached to test all the customizations that have been made in the past.

Decision-making on custom development usually involves several stakeholders. Business owners define the needs, IT evaluates feasibility, and management balances cost and resources. Some organizations follow a money-first strategy focusing on budget constraints, while others prioritize people-first strategy, ensuring user adoption and sustainability.

Managing customizations requires thorough documentation capturing purpose, design, and change history. Version control is essential, ideally with source code repositories and automated testing to reduce risks during upgrades.

Development should be allocated appropriately: own IT team is suited for known business needs with stable requirements, while vendors or specialized partners are better for complex or regulated customizations requiring deeper product knowledge.

Where applicable, design customizations to be configurable later, reducing technical debt and easing future changes.



Figure 1 SDLC

# Common customization areas in MES and LIMS

Certain functionality in MES and LIMS frequently requires customization due to unique operational or regulatory needs. It helps to categorize these by those common to both systems and those specific to each.

For both MES and LIMS, workflow logic such as deviation handling with escalations often needs adaptation. Role definitions like multi-level QA or reviewer separation, and country-specific compliance reporting are typical customization points. Legacy system integration, such as ERP or SCADA, also frequently requires tailored solutions.

For MES specifically, machine connectivity protocols (e.g., OPC-UA), OEE calculations adapted to site/product KPIs, dynamic scheduling logic, and customized shop floor user interfaces are common areas of customization.

For LIMS, customizations often focus on sample tracking rules (e.g., stability or expiration logic), instrument data parsing for vendor-specific formats, audit trail enhancements to comply with 21 CFR Part 11, and adaptive batch test plans based on product specifications.

Concrete customization examples:

## MES Examples:

- *Site-Specific OEE Calculation Logic:* Different plants have their own methods of calculating Overall Equipment Effectiveness (OEE), reflecting local priorities and production practices. MES is customized to apply these site-specific rules to provide meaningful performance data.
- *Advanced Workflow for Deviation Handling:* Manufacturers often require complex, multi-tiered escalation and approval processes for deviations, quality incidents, or non-conformances beyond the standard MES capabilities. Custom workflows automate and enforce these regulated processes.

## LIMS Examples:

- *Custom Stability Study Tracking:* Pharmaceutical labs often require extended stability studies with unique sampling schedules and specific data analysis. Standard LIMS workflows may not support these special plans. Customization enables tracking these studies in compliance with regulations.
- *Instrument Data Parsing for Rare Vendor Formats:* Some labs use niche or legacy instruments whose data outputs are unsupported by the LIMS by default. Custom parsers or middleware are developed to correctly ingest and validate this data.

Customizations should always be carefully documented and validated to ensure ongoing compliance and support.

## Financial perspective on customization

Understanding the cost breakdown of customization is essential for proper budgeting and project planning. Customization costs typically involve several phases: design (including requirements gathering), build or coding, testing/validation, and documentation. These phases vary in effort and cost depending on whether the system is deployed in a regulated or non-regulated environment.

Table 1 Detailed cost of customization

Phase	MES (Regulated)	MES (Non-Regulated)	LIMS (Regulated)	LIMS (Non-Regulated)	Notes
<b>Design</b>	25–35%	15–25%	30–40%	20–30%	Regulated environments need thorough requirements gathering and risk assessment to meet compliance
<b>Build / Code</b>	30–40%	40–50%	25–35%	35–45%	Coding effort can be higher in non-regulated MES due to complex automation; LIMS builds need integration focus
<b>Test / Validate</b>	25–30%	10–15%	30–40%	15–25%	Regulated environments require extensive validation and documentation for audit readiness
<b>Documentation</b>	10–15%	5–10%	10–15%	5–10%	Compliance requires traceable documentation; lighter effort for non-regulated settings

### Overall Customization Cost as % of Total Services

Table 2 Overall cost of customization

System	Regulated Environments	Non-Regulated Environments
MES	30–40%	20–30%
LIMS	25–35%	15–25%

**Key Observations:**

- **Regulated industries** such as pharma, food & beverage (more and more), or medical devices demand more rigorous documentation, testing, and design efforts to comply with standards like FDA 21 CFR Part 11 or ISO 17025.
- **Non-regulated environments** often have more flexibility, reducing the testing and documentation burden but may increase build effort due to less standardized processes.
- **MES customizations** in regulated settings often require more upfront design and validation, while non-regulated MES customizations tend to shift more cost into the build phase due to complex automation or process needs.
- **LIMS customizations** tend to require relatively more effort in design and testing phases, especially when regulatory compliance is critical, reflecting the need to ensure data integrity and audit readiness.

**Note on compliance cost:**

Investing appropriately in documentation and validation during customization reduces risks and costly audit findings. Underestimating these can lead to expensive rework or regulatory non-compliance.

# User Acceptance and Compliance Benefits of Customization

Customization brings value beyond direct financial savings. From a compliance perspective, customizations can enable specific audit trails and documentation flows that reduce audit time and cost. This lowers compliance burden, especially in regulated industries.

User acceptance also improves dramatically when software workflows, screens, and interfaces are tailored to real-world tasks. This alignment reduces training time, errors, and resistance to change, ultimately improving operational speed and staff morale.

Audit efficiency benefits as well, with customizations that enhance tracking and reporting helping minimize audit findings and shorten inspection durations.

## Real MES Customization Examples:

### 1. **Automotive Production Line Traceability:**

A global automotive manufacturer customized their MES to add a vehicle-specific traceability module that tracks components across multiple plants and suppliers, integrating RFID scanning and legacy ERP data. This customization ensured compliance with strict industry standards and dramatically improved recall management capabilities.

### 2. **Pharma Batch Release Workflow:**

A pharmaceutical company added custom logic to their MES batch release process to enforce additional QA checkpoints and automated electronic signatures beyond standard features. This ensured 100% compliance with FDA 21 CFR Part 11 while reducing batch release cycle times by over 10%.

## Real LIMS Customization Examples:

### 1. **Environmental Lab Instrument Integration:**

An environmental testing laboratory customized their LIMS to parse and normalize data from a wide variety of field instruments, many of which lacked standard communication protocols. This customization streamlined data collection and reduced manual transcription errors, improving turnaround time and report accuracy.

### 2. **Clinical Trial Adaptive Test Plans:**

A clinical research organization extended their LIMS with adaptive test plan logic that automatically adjusts sample testing sequences based on interim results and protocol amendments. This helped them remain compliant with complex study protocols while maintaining lab efficiency.

## Cultural and Regional Differences in Customization Benefits and Change Management

The perceived benefits of customization and the way changes are managed can vary significantly by country or region, influenced by cultural norms, regulatory environments, and business practices:

- **Regulatory Focus:**  
Regions with strict regulatory frameworks (e.g., EU, US, Japan) prioritize compliance-driven customization, investing heavily in documentation and validation. In contrast, emerging markets may prioritize rapid deployment and user flexibility over exhaustive compliance features.
- **User Acceptance and Change Culture:**  
In some cultures (e.g., Northern Europe, Japan), users expect highly standardized processes with limited customization, emphasizing stability and auditability. Elsewhere (e.g., US, India), users may favor flexible, highly configurable systems allowing quick adaptation to business changes.
- **Change Control Rigor:**  
Companies in highly regulated regions often have formal change control boards and multi-level approvals for customization requests, while organizations in less regulated regions might adopt agile or less formal approaches, leading to faster but potentially less controlled customization.

Understanding and adapting to these regional and cultural differences is essential for the successful design, deployment, and management of MES and LIMS customizations, ensuring both business value and compliance.

Overall, even when direct ROI from customizations is sometimes hard to quantify, the indirect benefits of risk reduction, better compliance, and happier users contribute strongly to a successful system implementation and adoption.

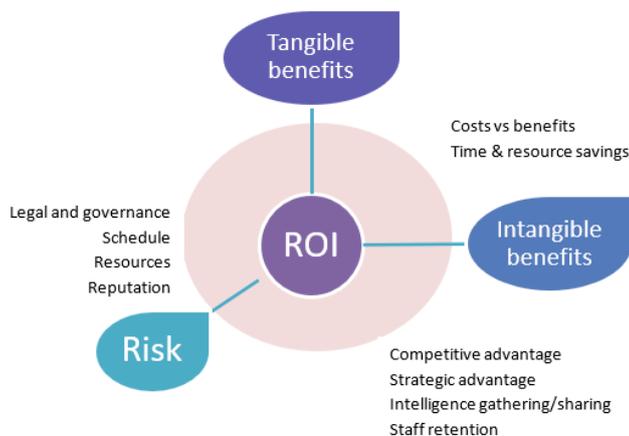


Figure 2 ROI is more than money

# Software Vendor Opinion — MES

Below is a general evaluation of MES vendors focusing on their strengths in configuration and customization capabilities.

*Note: The following evaluation is compiled by ChatGPT based on industry knowledge as of mid-2025. It reflects general market observations and is intended to support informed discussion, not as a substitute for vendor-specific due diligence.*

Table 3 Some MES vendors evaluation by ChatGPT

Vendor	Strength in Configuration	Customization Flexibility	Notes
Siemens Opcenter	High	Moderate	Strong in pharma and discrete industries
Rockwell FactoryTalk	Moderate	High	Best suited for Rockwell environments
Honeywell MES	High	Moderate	Common in process industries
GE Proficiency	Moderate	High	Requires experienced integrators
Schneider Wonderware	High	High	Strong in hybrid industries
Parsec TrakSYS	High	High	Partner-driven deployment model
MPDV Hydra	High	Moderate	Strong out-of-the-box MES
Tulip	Very High	Low	No-code platform for agile deployment
Dassault DELMIA MES	Moderate	Moderate	Strong in aerospace and automotive
Infor (MES)	Moderate	Moderate	Best with Infor ERP integration
Eyelit	High	Moderate	Focused on rapid deployment and ROI

## Software Vendor Opinion — LIMS

LIMS vendors show different balances between configuration and customization.

*Note: The LIMS evaluation table is generated by ChatGPT based on publicly available information, user feedback, and known solution architecture. As with all vendor selection efforts, validation through live demos and direct engagement with vendors is advised.*

Table 4 Some LIMS vendor evaluation by ChatGPT

Vendor	Strength in Configuration	Customization Flexibility	Notes
LabWare LIMS	High	High	Market leader, needs expert setup
Thermo Fisher SampleManager	High	High	Ideal for enterprise labs
STARLIMS	High	Moderate	Well-suited for regulated labs
LabVantage	High	Moderate	Requires governance for scaling
Autoscribe Matrix	High	Moderate	Excellent GUI for mid-size labs
Agilent SLIMS	Moderate	Moderate	Strong integration focus
PerkinElmer LabWorks	Moderate	Moderate	Often used in environmental labs
CloudLIMS	High	Low	SaaS-focused, fast to deploy
Core LIMS (Dotmatics)	Moderate	High	Gaining traction in biotech
Opcenter RD&L	High	High	Strong PLM-LIMS integration
iVention iLES	High	Moderate	Unified LIMS/ELN/LES in the cloud

Understanding vendor support policies for customizations and upgrade paths is critical to avoid technical debt and excessive maintenance costs.

## Best practice. Roles and behaviors

The following roles can be found in implementation projects, beside others.

Table 5 Some important roles in a solution implementation

Role	Responsibility	Additional Explanation / Example
<b>Process Owner</b>	Approves business need, owns process alignment	The Process Owner ensures that any customization aligns with overall business objectives, for example, by approving a workflow change that speeds up order processing without compromising quality.
<b>IT Architect</b>	Designs sustainable, scalable solutions	The IT Architect selects technologies and design patterns that support future upgrades, such as using modular plugins instead of hardcoded features to allow easier system evolution.
<b>Validation Lead</b>	Ensures compliance and validation planning	Validation Leads develop test protocols and documentation to prove that customizations meet regulatory standards, like preparing traceability matrices for FDA audits.
<b>Vendor/Partner</b>	Executes configuration and custom build	Vendors often bring specialized expertise; for instance, they might implement a complex data interface that internal teams lack experience with.
<b>QA/Compliance Officer</b>	Reviews system use in regulated environments	QA officers monitor if custom features are used as intended, preventing unauthorized changes—such as verifying audit trails on a new electronic signature module.
<b>Project Manager</b>	Manage timeline, scope, and change control	Project Managers coordinate all stakeholders, balancing resource availability with changing business needs, like negotiating scope adjustments when unexpected compliance requirements arise.
<b>Change Manager</b>	Oversees communication, training, and impact of changes on users and processes	For example, when introducing a new customized workflow, the Change Manager ensures end users are informed, trained, and supported during the transition. Minimizing resistance and maximizing adoption.

To manage customizations well over the lifetime of the solution, consider the following tips:

- 1. Prioritize configuration over customization whenever possible.**  
This reduces long-term maintenance effort and upgrade risk; for example, choosing to adjust user roles via built-in tools rather than coding new permission modules.
- 2. Document and version-control all customizations.**  
Proper documentation and versioning enable quick troubleshooting and rollback if issues arise, such as tracking changes in a shared repository like Git or Azure DevOps.
- 3. Engage stakeholders from business, IT, and compliance early.**  
Early involvement ensures that solutions meet business needs, technical feasibility, and regulatory demands, preventing costly rework later in the project.



- 4. Validate customizations thoroughly, including regression testing.**  
Regression testing confirms that new changes don't break existing functions; for example, automated test suites can help quickly verify critical workflows after each update.
- 5. Negotiate clear ownership and support terms for custom code.**  
Defining who maintains customizations avoids future disputes, such as agreeing that the vendor supports all custom plugins developed by their team while internal IT owns small scripts.
- 6. Choose vendors with proven configuration and customization frameworks.**  
Vendors offering robust tools reduce risks, like platforms that provide drag-and-drop workflow editors and comprehensive API documentation.
- 7. Plan for future upgrades and compatibility.**  
Design customizations so they don't block software upgrades; for example, implementing extensions via documented APIs instead of modifying core product files.
- 8. Train end-users on configured workflows to maximize adoption.**  
Effective training helps users understand new processes and reduces resistance, such as interactive sessions and quick reference guides tailored to different roles.
- 9. Continuously review customizations for potential simplification.**  
Regular audits identify obsolete or overly complex custom code that can be replaced with standard features as the product evolves.

## Conclusion

MES and LIMS customization is often unavoidable — and it should be approached strategically. A well-configured base with clearly governed customizations yields the best of both worlds: efficiency and flexibility, without sacrificing compliance or maintainability.

The key lies in understanding where the line is drawn between configuration and customization, and managing each with the right people, processes, and tools. When you start with configuration as your default approach, you minimize long-term cost and risk. But when customization becomes necessary, which it often does, you must treat it as a software development activity: with formal design, stakeholder involvement, documentation, and lifecycle planning.

Moreover, the benefits of customization go beyond operational gains. They touch on human and regulatory dimensions: enabling cleaner audits, reducing compliance risks, improving user satisfaction, and supporting adoption. In highly regulated industries like pharmaceuticals, automotive, and food manufacturing, well-managed customizations can actually lower the cost of audits and reduce the frequency of compliance findings.

It's also important to recognize that culture and geography play a role. In some regions, flexibility and tailoring are seen as essential to user satisfaction. In others, strict adherence to standard functionality is preferred for predictability and governance. A mature approach must balance both, respecting local needs while maintaining global standards.

Finally, customization should never be static. What was necessary five years ago may now be available out of the box. Periodically reviewing your system landscape allows you to simplify, standardize, and reduce technical debt. That's how you sustain the value of MES and LIMS over time, not just as software, but as platforms for digital excellence.

Contact MOMi:

Enquiries: <https://www.mom-institute.org/contact-us/>

Website: <https://www.mom-institute.org/>

## List of tables

Table 1 Detailed cost of customization .....	5
Table 2 Overall cost of customization .....	6
Table 3 Some MES vendors evaluation by ChatGPT .....	9
Table 4 Some LIMS vendor evaluation by ChatGPT .....	10
Table 5 Some important roles in a solution implementation .....	11