

NUXEO



EXCLUSIVE
CHECKLIST
REPORT

Low-Code Platforms and Applications

Getting the Lowdown on Low-Code

By Marydee Ojala, Conference Program Director, Information Today, Inc.

You've probably heard the old saying that you can have cheap, you can have fast, you can have good, but you can't have all three. I once had a colleague who actually said this to her clients, asking them to pick which two they wanted since she refused to promise delivery of all three. I'm beginning to think that low-code puts the lie to that. With low code, you just might be able to deliver all three.

Low-code development platforms exist to simplify the development of software applications by taking a visual approach. Low-code uses graphical user interfaces and easy to use configurations without relying on the manual writing out of coding commands and scripts that is part of traditional programming. Model-driven design, automatic code generation, drag-and-drop, and visual programming are the hallmarks of low-code development. This allows for a more rapid delivery and ongoing collaboration in the development process. It also gives less technical people the opportunity to build their own business applications.

Given the increasing familiarity of people with the computing environments presented to them, first by their desktop computers, then by laptops and handheld devices, the move to low-code is inevitable—and growing. The ability to understand how computers work and make them do what you want them to do is much more common now than it was in the past. People want to take charge of their devices rather than rely solely on their IT departments. This does not, by any means, obviate the important contributions of those IT departments, but it does pave the way towards more flexibility in automating business processes.

THE DOWN LOW ON CONTENT MANAGEMENT

When it comes to Enterprise Content Management (ECM) and its more recent designation of Content Services Platform (CSP), there's been at least the implicit acceptance of the two out of three mentality. If you want a robust ECM developed quickly, you'll pay a significant amount of money. If you cut the budget, you may be able to speed up the timetable and still have a decent product. How about high quality, low cost, but the delivery date is so far in the future you can't enter it into your online calendar? Ensuring that all the features you want are included has ramifications both on the money you'll spend and the time it takes to develop the platform.

Choosing two of the three, however, involves some value judgements. What qualifies as "cheap"? Depending on the enterprise, the definition of inexpensive versus expensive varies. What is considered an astronomical amount of money in one situation could easily be considered mid-range in another. How about "fast"? Again, this can be subjective. If I say that I'm really in a

hurry for something, do I mean in the next hour, the next week, the next month, or sometime next year? It's useful for any contractor, whether internal or external, to quantify "fast." Nail down when the client expects results.

The same goes for "good." As a reference point, spell out what is most important for a specific ECM project. Everyone will tell you they need high quality. Ask them for examples, which will give you an idea of what they prioritize as high quality. Is there a current pain point they need addressed and fixed? Employees might be complaining about finding irrelevant or out of date content. Not discovering certain formats of content could be an issue with their legacy ECM systems. Siloed information is a perennial problem.

It's frustrating, from the developers' perspective, when an existing CSP isn't meeting the needs of its users. They know what people want, yet they are unable to supply it properly. One problem could be that, although the original developers listened to what users said was wanted and what the current system lacks, they didn't probe for the intent behind the users' words. This is crucial to getting a solution to the problems adequately addressed the first time around when choosing a new platform. That's when a new direction is indicated.

"Low-code's drag-and-drop functionality bypasses extensive dialoging with the IT department. Instead, citizen developers who already understand user needs can build working prototypes of workflows, reports, or apps, providing rapid time to value."

LOOKING FOR A THREE-PEAT

Sean Baird, Director of Product Marketing, Nuxeo, suggested three slightly different criteria than cheap, fast, and good when I spoke with him about low-code development. He came up with faster, easier, and more effective. Oh, and yes, you can definitely have all three. With low-code, you can expand the pool of people involved in creating the ECM platform. Nuxeo provides tools to help both experienced developers and novices develop solutions.

What you should be thinking about when choosing a low-code CSP revolves around three key areas, according to Baird. Start with why you want to investigate low-code in the first place. Most likely, it's because the legacy ECM currently in place simply isn't doing the job. It's become difficult to refine and enhance given the increasing complexity of the content it's trying to manage.

Legacy systems, while perfectly adequate in their day, often do not allow for the agility and scale needed for today's content platforms. Technology has moved on and tools exist to help meet the challenges. Much as we would all like to believe that software of any variety works "right out of the box," the reality is that it isn't that simple. It needs to be configured to individual needs, which requires not only initial development, but also continuing maintenance and tweaking.

“Low-code, as it enhances and replaces traditional programming, has the potential to create business value. By using the concept of citizen developers, it spreads the workload among many people, most of whom work outside the established IT department.”

What’s helpful is for end users to be able to develop solutions on top of a CSP. If the legacy system rollout didn’t quite meet the requirements and didn’t take end users into account, low-cost low-code can remedy that.

BENEFITS OF LOW-CODE

The benefits of low-code is the next area that Baird cites. Ongoing iterations, he emphasizes, can take many different forms. It could involve altering workflows, adjusting metadata, reconfiguring user experiences, and/or meshing with other business systems to accommodate changes in the business environment. He mentions benefits such as being able to quickly adjust to new regulations—and these are proliferating at a great rate—and to engage with users on many levels.

Another advantage to low code development is the involvement of what Baird calls “citizen developers.” Involving those without a highly technical background is key to low-code development. That doesn’t mean making every employee a software developer. Some probably don’t want to be a developer and, for others, their time might be better spent on other business activities. It is not, Baird explained to me, a totally democratized process, but expanding the pool of potential developers is beneficial to the overall success of content management. They don’t have to jump into the deep end of programming; the low-code tools will get them started.

Low-code’s drag-and-drop functionality bypasses extensive dialoging with the IT department. Instead, citizen developers who already understand user needs can build working prototypes of workflows, reports, or apps, providing rapid time to value.

Many of Nuxeo’s clients are in the financial services and insurance industries. These are highly regulated and it’s imperative that companies in those industries keep up with changing regulations and guarantee compliance. Agility in reacting to regulatory changes is difficult for legacy ECM systems—another benefit of switching to low-code development. Pivoting swiftly is a core component of a low-code environment.

THE HIGH ROAD TO LOW-CODE

Once you’ve decided low-code is for you, the third area to consider is what to look for in a low-code offering. Baird suggests several criteria. Most importantly, assess your core requirements and what your end goal is in terms of workflow and apps to be built on top of your content platform. You need a roadmap that accommodates both today’s situation and what is likely to happen in the future. That’s the high road to low-code. Consider what internal skills already exist within your organization. Baird recommends taking the typical software procurement approach and flipping it. Start with the needs then move on to the platform functions and features.

Don’t forget the user. Dazzled by the promise of shiny features and glowing functionality, it’s all too common in the procurement process to overlook the user. Remember that, even with its shortcomings, a legacy system is something that users have become accustomed to. They may not like parts of it, but they also dislike change. What low-code can provide is simple trialing and the ability for users to give feedback.

Beyond the user experience, tools that speak to more than one category of users smooth the adoption of low-code. Business processes include multiple use cases, and extensive re-coding is not desirable. The process configuration, automation, and deployment should embed low-code functionality.

Platform configuration benefits greatly from low-code development. This cuts across numerous areas on the CSP, including the roles and permissions of users, document types, metadata, system reporting, and external application integrations, as well as compliance with government rules and regulations.

Low-code, as it enhances and replaces traditional programming, has the potential to create business value. By using the concept of citizen developers, it spreads the workload among many people, most of whom work outside the established IT department. Citizen developers often have a more accurate idea of what users want since they are closer to the user base.

LOW-CODE ICE CREAM

Step back a minute from the technicalities a low-code platform to consider ice cream. Specifically, ice cream that combines three flavors. It could be Neapolitan, which contains vanilla, chocolate, and strawberry, or Spumoni, which has cherry, chocolate, and pistachio. It would probably take extra effort to swirl those three ice creams together—that’s what could happen in a non-low-code environment. Separating them allows consumers to mix flavors to please their individual taste buds. Less effort, better outcomes.

Citizen developers can decide which flavor to eat first or whether they’d like to put a tad of chocolate *and* strawberry on their spoon. They might even decide that they want to change the formula. Maybe they want vanilla instead of pistachio in their Spumoni or rum raisin in addition to chocolate, vanilla, and strawberry in their Neapolitan. They might decide to rival Ben & Jerry’s in the creative and often pun-laden naming of their ice cream creations.

There’s an old joke in coding circles, or perhaps it’s better to call it a truism, about computers that don’t do what programmers want them to do, only what they tell it to do. That is still a valid point, but low-code development makes it much easier to change what you’ve told the computer to do so that it *does* perform as you expect.

We’ve moved from two out of three aren’t bad to three’s a charm. Or, perhaps it’s more a refutation of the old saying “You can’t have your cake and eat it, too.” With low-code, you can not only have your cake and eat it, but you can also top that piece of cake with Neapolitan ice cream. ■



Marydee Ojala is conference program director for Information Today, Inc. She works on conferences such as Enterprise Search & Discovery, which is co-located with KMWorld, and Data Summit, among others. She is a frequent speaker at U.S. and international information professional events. In addition, she moderates the popular KMWorld webinar series. Ojala is based in Indianapolis, Indiana and can be reached at marydee@infotoday.com.

Three Key Areas to Consider When Selecting a Low-Code Content Services Platform

By Sean Baird, Director of Product Marketing, Nuxeo



INTRODUCTION

Imagine yourself buying a new car. The feeling as you drive it out of the car dealership and down the road is of pure excitement and bliss. But then imagine the disappointment when you find that the new car can only be used in the daytime because there is no switch for the headlights or that the temperature of the

A/C is permanently set to 49.5 degrees and cannot be changed.

This feeling is familiar for many business users of enterprise content management (ECM) systems—which offer tremendous potential for automation, process improvement, and productivity gains only to fall flat because the solution is not adequately tuned to meet the business’s requirements.

LOW CODE CAN HELP REMOVE THIS FRUSTRATION.

The concept behind low-code application development within a content services platform (the modern equivalent of an ECM system) is simple. Low-code development enables organizations to be more agile by reducing the amount of effort that is needed to build on the platform—providing business analysts and software developers with the tools to develop software applications *without* writing complex code or scripts.

The benefits of doing this are clear—empowered developers, reduced reliance on IT or software vendors to make system changes, and overall increased agility levels within the organization. So, how can organizations take advantage of low code to strengthen their enterprise content management capabilities?

WHY IS LOW CODE NEEDED?

Software applications continue to increase in capability and complexity, especially for domains such as document and knowledge management.

An ECM or CSP is rarely used “out of the box” without any setup and configuration to the customer’s specific needs. Legacy ECM software is difficult to configure and amend to address the dynamic needs of the enterprise, which ultimately stifles the business’s ability to adapt quickly or, in some cases, even get the software up and running.

Deployment projects are notoriously troublesome¹, but challenges in system rollout ironically are often due to the software’s inability to be easily configured further downstream.

Think about it—if a software application is challenging to adapt and refine once it has been installed and configured, then by default,

you have to get it right the first time during the initial deployment. The pressure-cooker combination of overwhelming complexity levels, time-demands, and the software’s inability to deal with changing requirements lead to both long deployment projects and, unfortunately, numerous failed installations.

Yet, the need for continual refinement and enhancement does not stop when the initial application deployment is completed. As business requirements change, new regulations are passed, and environmental factors such as remote working take hold, the CSP needs to adapt accordingly.

“Providing a CSP solution that’s easily configurable and extendable reduces the need to get everything right in an initial deployment, encouraging a mindset of proactive, ongoing refinement rather than a one-and-done viewpoint.”

Providing a CSP solution that’s easily configurable and extendable reduces the need to get everything right in an initial deployment, encouraging a mindset of proactive, ongoing refinement rather than a one-and-done viewpoint. It also removes the need for dedicated assistance from the vendor by allowing internal “citizen developers” to step up. These citizen developers can be business analysts or less experienced developers—all being enabled by the reduced development skills needed to work in the low-code environment.

Content services platforms that offer low-code tools provide citizen developers and software developers alike with the ability to configure, refine, and extend the software application. The low-code approach significantly lowers the level of technical knowledge to make such adaptations. With low code, you don’t have to be an expert—but if you are, then the productivity gains can be even more significant.

LOW CODE IN CONTENT SERVICES PLATFORMS (CSPs)

All the way back in 2010, Cheryl McKinnon of Forrester wrote,² “constant tailoring, tweaking, [and] improving helps tune how the

“The low-code approach significantly lowers the level of technical knowledge to make such adaptations. With low code, you don’t have to be an expert—but if you are, then the productivity gains can be even more significant.”

business runs, and ECM platforms should be able to easily keep up with ongoing innovative iterations demanded by business analysts and content architects.”

In 2021, these ongoing iterations take many forms. They include the editing of workflows, the amendment of metadata schemas, reconfiguring user experiences, and integration with other business systems. Each of these is very different, but all require deep system or development knowledge in legacy systems.

Low-code configuration provides the enhanced flexibility and agility to allow such iterations to be performed without deep technical knowledge—offering a modern business several benefits:

✓ **Increased agility and scale**

Agility in today’s fast-paced business environment is critical. The ability to adapt and quickly respond to shifting markets, user requirements, and potential opportunities can be a real competitive advantage. Rigid ECM systems generally fall short of these objectives—low code, however, can deliver agility across the whole organization.

✓ **Enhanced ability to comply with new and changing regulations**

A key use case for ECM and CSP systems is corporate compliance. The ability to quickly respond to new and changing compliance and regulatory requirements such as GDPR, CCPA, HIPAA, and others is critical—and being able to implement these changes without requiring dedicated IT resources is a key benefit of low code.

✓ **Rapid time to value**

Drag and drop visual development offers organizations the ability to develop entirely new solutions to business challenges with existing resources. In the time it takes to describe an idea to IT, citizen developers can build their own working prototypes of new workflows, reports, and even apps.

✓ **Increased user engagement**

Traditional ECM systems have a reputation for low user adoption. This is not surprising given the rigidity of legacy systems and users’ inability to make any changes. Low code, of course, rewrites the script here.

The benefits of a modern CSP that includes a low-code development environment are considerable for the enterprise, department, and developer. The creation of a wide group of enabled citizen developers is a powerful business agility driver.

Low code is not just for experienced developers. However, for those that do have deep coding skills, low code enhances the development experience significantly:

✓ **Faster deployment**

Organizations will always need developers to deliver the most complex aspects of an enterprise content management solution.

Customized complex workflows, interactive dashboards, and API-level integrations to other applications are examples of where a software developer will likely step in. In these areas, low code allows experienced developers to deliver faster, reducing the time to prototype and test solutions through consistent, platform-enabled, visual tools.

✓ **Lowered expertise requirements**

Even highly skilled developers cannot be experts in all areas. Back-end coders often struggle with interface changes, and in return, UI experts are typically not back-end coders. Low-code configuration can bridge the gap between these extremes, providing each group with the ability to work beyond their core skill sets without extensive retraining.

✓ **Iterative development**

Due to the ability to rapidly prototype and deliver new solutions, low code strongly encourages innovation. The low cost of initial development, and any subsequent redevelopment, allows developers significant freedom to be creative.

WHAT SHOULD AN ORGANIZATION LOOK FOR IN A LOW-CODE OFFERING?

Not all low-code environments are created equal. Some focus on the user interface, some on workflow editing, while some offer more comprehensive capabilities across multiple areas.

However, before exploring any specific low-code tooling, organizations need to understand their own core requirements and capabilities. At the top level, these should be analyzed across three dimensions:

1. Core requirements

What does the business want to configure and what apps will be built on top of the CSP? Will changes be limited to specific workflow design areas or include more comprehensive configuration options such as UI layout, notifications, and metadata schema management?

2. Low-code roadmap—today and tomorrow

What is the low-code roadmap for your solutions? How much of the CSP will need to be low-code enabled? Which users will be targeted to use these tools? Will this change over time?

3. Internal skills

What skill levels exist internally? How many technical, semi-technical, and non-technical users exist? How many of each group would be required to use low-code capabilities?

With a clear understanding of what the organization is looking to achieve with low code and the level of internal resources able to use such capabilities, the search for an appropriate platform can begin.

However, when evaluating content services platforms with low-code capabilities, organizations should look to flip the typical software procurement approach on its head. Instead of focusing on platform functions and features first, user experience needs to drive the conversation. In turn, this leads to business process requirements, platform capabilities and configuration needs—but with a clear understanding of and focus on what users and the business require from the platform.

USER EXPERIENCE

By focusing first on user experience, priority is given to the benefits that end users will see from low-code development. Users typically do not like change that is forced on them so a key aspect of low code is the enablement of simple trialing and feedback capabilities. Any low-code tool needs to offer the following:

User Activation

1. simplified capabilities for
 - ✓ rapid development of user interface prototypes
 - ✓ real-time drag and drop customization of UI
 - ✓ deployment of test environments
2. ability to add and configure UI widgets
3. apply configurations across all devices and platforms

“The rigid ECM systems of the past never realized their full potential, in many cases were never fully implemented, and absolutely never delivered real value to the business...”

Business Process

At a business process level, the low-code development environment needs to offer tools that cater to various categories of users—from business analysts to experienced developers. The environment ultimately needs to enable the creation and management of multiple use-cases without extensive re-coding.

Process Configuration

The low-code environment needs to enable enterprise-grade workflow management capabilities. This requires visual drag and drop functionality to design the overall process flow, the individual steps, and the logic decision points. However, each part of the configuration should include comprehensive capabilities to configure parameters, integrate external data sources for validation, and add low-level code—if and when required.

Automation & Deployment

The definition and configuration of system components and workflows is a key benefit of low code, but of equal importance is the deployment of new capabilities across the business. Low-code tooling is not just for the visual creation of workflows but can be applied to broader automation exercises.

The tooling needs to embed access to low-code functionality in multiple areas of the system, including:

- ✓ automated tasks & reporting
- ✓ UI screens & elements
- ✓ user permissions & system security
- ✓ compliance & governance

PLATFORM CONFIGURATION

Wherever low-code tools are used, they provide an easy to use, often visual environment to configure and extend the content services platform. From one perspective, this delivers UI design, workflow editing, and app building capabilities, but the underlying configuration can benefit from low-code principles also.

Content services platform deployments typically involve metadata schemas that encapsulate complex multi-dimensional relationships between content, users, and the need for business governance, compliance, and security. Any CSP will benefit significantly by offering low code configuration of:

- ✓ users, roles, and permissions
- ✓ document types
- ✓ metadata—across multiple schemas
- ✓ system reporting
- ✓ external application integrations
- ✓ compliance and governance requirements

CONCLUSIONS

Low code was a secret sauce, well hidden by the development community for many years. But today, low code is not just for experienced software developers—the rise of citizen-developers who can configure, extend, and super-charge their use of content management software is increasingly frequent and beneficial.

The rigid ECM systems of the past never realized their full potential, in many cases were never fully implemented, and absolutely never delivered real value to the business. Modern content services platforms offer the foundation on which low code tools can build flexible, agile, user-focused solutions. This pivot is revolutionary, allowing a much wider audience to see value from this technology for the first time via:

- ✓ faster time to value
- ✓ optimized user experiences
- ✓ increased user adoption
- ✓ lower total cost of ownership
- ✓ faster solution development cycles
- ✓ reduced need for specialized development resource

The COVID pandemic and specifically the increase in remote working, has brought cloud-based content management to the world. But it's low-code configuration that has the potential to make the world forget the shortcomings of legacy ECM systems by falling in love with the flexibility and convenience of content services platforms enriched by low-code tools. ■

1. <https://blog.prosci.com/three-reasons-enterprise-change-management-efforts-fail>
2. <https://info.aiim.org/aiim-blog/newaiimo/2010/09/29/8-factors-to-consider-when-evaluating-a-next-generation-ecm-platform>