

MAKING INFORMED DECISIONS

**Low-Code vs. Custom
Application Development
in Federal Agencies**



EXECUTIVE SUMMARY

Low-code software development platforms are increasingly popular, and Federal IT decisionmakers now face a key question: should their agency build software applications using low-code tools or take a custom software development approach?

The answer impacts the effectiveness and cost of solutions essential for mission delivery. To choose the right path, Federal IT leaders need to account for a number of variables, including time, budget, availability of development and support resources, requirements

for user interfaces, and workflow complexity. If these dynamics are not considered, agencies may find themselves trying to put a round peg in a square hole, wasting time and money.

This whitepaper provides a framework for Federal OCIOs and decisionmakers to help determine when it makes sense to use low-code solutions and when custom application development is best. It is based on TCG's extensive experience in both low-code and custom development to meet Federal agencies' needs.

DECISION FRAMEWORK — LOW-CODE VS. CUSTOM DEVELOPMENT RUBRIC

Organizational dynamics and priorities vary greatly when it comes to development projects, so determining which factors play the most important role in choosing low-code or custom can be daunting. The areas below are by no means exhaustive, but they provide an overview of key considerations for the decision-making process. This framework is intended to be expanded upon by your organization as your evaluation process

kicks off, and can turn into a list of recommendations for leadership to make informed, strategic decisions.

Federal agencies need to strike a balance between innovation, cost, and efficiency, often while dealing with limited resources. Federal CIO's should consider the following:



SCHEDULE TOLERANCES

Schedule Tolerances — 3 Essential Questions

- How quickly do you need to see a minimum viable product (MVP)?
- How much do you want to test the system prior to going live?
- Is there a critical date the system must be working by?

Regardless of industry, project management and solution acquisition involve time constraints. In the case of evaluating low-code versus custom development implementations, there is a drastic difference between their time to a working minimum viable product.

Low-code considerations

As a general rule, low-code solutions offer the opportunity for rapid prototyping, which can radically reduce the total time and cost of a solution being implemented. TCG delivers rapid prototypes using Microsoft Power Apps at the Millennium Challenge Corporation (MCC) and Atlassian Jira and Confluence for the Office of Management and Budget (OMB). Similarly, low-code solutions leverage prebuilt application components and attributes to speed development time.

Custom development considerations

Custom development of applications is inherently slower and typically more costly, because of the human and technical resources required to sustain it. That said, custom development offers complete control of solutions. If there is ample time to complete a custom development project it will likely produce a more tailored, refined, and precise product.



SECURITY REQUIREMENTS

Security Requirements — 3 Essential Questions

- Does your project require role-based access or field visibility based on user roles?
- What is your organization's current security culture and how would this product adhere to it?
- Does your environment require any security reviews and approvals (such as Authority to Operate) for tools needed to build the application, such as Microsoft's Dataverse, or external libraries and dependencies for custom development, etc.?

FedRAMP programs standardize security requirements under the existing legal and regulatory framework to support efficient adoption of modern technologies. When considering low-code solutions, a technology-agnostic approach allows the use of whichever provider best fits the needs of the agency and project. Using authorized low-code applications can minimize start-up efforts and personnel requirements needed to architect and implement security solutions in a custom project.

Low-code considerations

Most low-code application platforms offer out-of-box functionality to provide data security and access permissions, with modifications as needed. However, there are times when the out-of-box security functions are not sufficient. For instance, a recent model-driven application with numerous user roles and permissions required TCG to add a significant amount of custom Javascript to handle highly complex security requirements, like conditional logic to lock down individual application fields based on a user's role.

Custom development considerations

As a general rule, custom applications typically have to architect and implement a security model from the ground up. This allows a high level of access controls and security granularity, but can also lead to very complex structures to ensure those security needs are addressed. If pursuing the custom development route, TCG recommends evaluating the organization's current security posture and culture. Keep in mind factors such as current threat monitoring, testing, and secure coding practices for a complete picture of your resource advantages and requirements in the security realm.



INTEGRATION WITH EXISTING SYSTEMS

Integration with Existing Systems — 3 Essential Questions

- How does each approach fit within the existing IT portfolio of your organization?
- Are there opportunities or requirements at the start of this project for connections between existing tools?
- Is this new application going to replace, enhance, or provide distinctly new functionality in your organization?

To promote adoption of new applications, agencies should take steps to ensure the applications make use of information from existing systems and are a good IT portfolio "fit" to deliver new capabilities for users. When done properly, these can improve users' experiences, save time, and support enterprise reporting.

Low-code considerations

Low-code solutions are often flexible and can be integrated with existing IT solutions at an organization using prebuilt vendor connectors. Some environments may have low-code solutions associated with their vendor platform, such as Power Platform for a Microsoft-supported IT shop or Amazon App Studio when leveraging AWS.

Custom development considerations

Custom development supports unique scenarios or may be a better fit to integrate with an agency's existing custom applications. Custom development allows extensive control over system integrations, but often requires increased time and cost to do so.



COST

Cost — 3 Essential Questions

- Is your organization's IT budget dependable enough to forecast several years out?
- Are there any opportunities to find cost savings in other areas of the organization by implementing one solution approach instead of the other?
- Aside from regular maintenance, will this solution require enhancements and continuous improvement over the course of its lifecycle?

Regardless of your agency's budget, it's critical that a detailed understanding of costs, both up-front and into the future, is analyzed by leadership prior to making a decision on a development path.

Low-code considerations

Licensing is the obvious up-front cost. Agencies should keep in mind the licensing that would be needed to support multiple application environments, i.e. development, testing, and production. This includes potential costs for data storage depending on which options are desired for the agency (e.g. SharePoint vs. Dataverse). However, historical trends all but guarantee that license models and pricing from low-code vendors will change to remain competitive, so there is inherent uncertainty in predicting costs for low-code solutions in the future. Low-code solutions will incur costs for ongoing maintenance (e.g. new features) and support, like custom-developed applications.

Custom development considerations

Custom applications tend to create high up-front costs for the development effort and then experience a cost reduction over time. That said, custom applications can have higher costs for ongoing maintenance (e.g. when new features need to be developed), and supporting the application amid changes in technologies and security is a constantly evolving spectrum. TCG's experience with custom development projects has shown that there is typically a higher risk for bugs, maintenance complications, and general upkeep than agencies often realize at the outset of projects. Also, the more complex and nuanced the code, the longer tail a product has to keep it up to date. If code and environmental factors aren't carefully monitored, the technical debt of custom development can snowball and become a huge cost risk.



WORKFLOW COMPLEXITY

Workflow Complexity — 3 Essential Questions

- What are the number of approval levels?
- What are the automation requirements?
- What are the nuances of the required conditional logic?

Workflows are a foundational element of all data collection and collaboration applications, and as such workflow requirements can play a major role in your evaluation process. Answering these questions will help you assess whether low-code solutions can meet your needs or if the complexity exceeds the capabilities of existing applications on the market.

Low-code considerations

Low-code solutions can address many workflow automation needs including review levels, notifications, and setting permissions. For MCC, TCG developed and supports a workflow involving sophisticated business logic, many department-level task reviews, and email notifications. However, very complex workflows involving high numbers of reviews and/or discrete send-backs depending on which review level rejects a submission may exceed low-code capabilities or result in overly complicated flows prone to failure.

Custom development considerations

Custom-developed workflows can be shaped to handle very complex needs of an agency, with tailored logic and action triggers, as well as simpler automations. For one agency, we rebuilt and expanded a custom management information system. Custom development was required to handle the complexity of Federal compliance for audits and record retention, internal and external user groups, and multiple reporting packages and workflows. This supported role-based reviews, comments, and notifications for varied data types.



PRE-BUILT COMPONENTS

Pre-Built Components — 3 Essential Questions

- Is the add-on or plugin from a trusted vendor or a lesser known source?
- What are the external dependencies used by the add-on?
- What is the reputation of the vendor that created the add-on?

For both custom and low-code development, the available markets for plugins, add-ons, or other modular components can save development time and money by being added to a solution. Security considerations are of primary importance in assessing any pre-built component.

Low-code considerations

Amid all of the available options for pre-built components, it is critical to focus on trusted vendors and offerings like [Microsoft AppSource](#) and [Atlassian Marketplace](#). TCG recommends avoiding add-ons from lesser known sources to avoid potentially compromised code, but even with trusted “big names”, careful review of the actual creator (which could be Microsoft or a relatively unknown submitter) as well as any external dependencies used by the add-on is very important to ensure security.

Custom development considerations

As with low-code solutions, agencies need to consider the security of any add-on components being assessed, including dependencies on external libraries which could themselves become compromised. While each custom application has different requirements, TCG recommends leveraging as much as possible from external sources. Assuming the external resources are secure, have solid reputations for performance, and come with direct lines of support, this approach can help reduce the burden for your development team and allows them to focus their efforts more exclusively on the capabilities that matter most to your organization.



REQUIRED EXPERTISE AND RESOURCES

Required Expertise and Resources — 3 Essential Questions

- Does the agency have the properly skilled development and support staff in place, as well as the technical tools (e.g. development tools, development environments, repositories) needed for the job?
- Will new staff need to be hired, or existing staff trained?
- Is there a willingness and aptitude in the organization to foster a citizen development approach where agency employees take on some development work?

Having skilled staff and appropriate technical tools is critical to the success of any application development or enhancement effort. Agencies can foster having these foundational elements through a variety of methods depending on the business needs as well as the agency’s existing resources.

Low-code considerations

Availability of low-code platform expertise within and outside your organization, as well as existing or potential citizen developer communities, should be assessed. It is important to note that low-code citizen development is not as simple as vendors like Microsoft would lead people to believe! Check out [TCG’s blog post on citizen development](#) for perspective on the low-code route. Many low-code applications have robust documentation and support, which can help staff quickly onboard and begin using the tools.

Custom development considerations

Consider the availability of a prospective development team carefully when evaluating the custom development path to avoid future programmatic and technical roadblocks. In TCG’s experience, employing and empowering teams with diverse skillsets and backgrounds reduces risk by encouraging creativity and challenging convention. In other cases, though, where there is a scarcity of talent for a particular coding language for instance, it can be quite challenging to create stability in the application.



USER EXPERIENCE

User Experience — 3 Essential Questions

- How comfortable is the projected user base with new technologies?
- Is the user experience more or less important than the functionality of the application?
- Are there styling and design standards for this application that must be met to ensure consistency with other tools in the portfolio?

When deciding to build a new application, agencies must consider the importance of its look and usability in fostering success in user adoption, as well as consistency of user experience with existing agency applications. In the Federal space, particular attention should be paid to determine how much effort will be needed for Section 508 compliance as well as responsive application behaviors on mobile devices, e.g. dynamic resizing based on device screen size. TCG's experience has shown that user experience considerations can have drastic impacts on technology adoption and should be assessed with great care prior to deciding to follow a low-code or custom development path.

Low-code considerations

Power Apps and other low-code tools provide highly customized user experiences like through Canvas apps with their reusable screen elements, but these require responsiveness and accessibility to be specifically developed. Model-driven apps allow little customization but deliver friendly interfaces and compliance features out of the box. That said, they may require custom Javascript to complement application rules and send custom notifications.

Custom development considerations

Custom apps allow the highest degree of customization, but require the most work to deliver mobile-friendly interfaces and 508 compliance. They also present a greater risk for failure due to the added layer of UX responsibility for the development team. In TCG's experience building custom developed applications for Federal customers, user experience is the first thing to be sacrificed when resources are constrained. In most custom development cases, it's smart to include a UI/UX specialist on the development team and/or ensure that the product owner maintains a user-oriented product roadmap to make the most of the user experience.

SCENARIOS FAVORING LOW-CODE DEVELOPMENT

- Limited development budget or short timeline
- Clear requirements that allow straightforward design and implementation
- Need or desire to prototype or iterate quickly
- Desire to allow non-technical users the ability to create application functionality or reports
- Need for integration with other tools from same platform vendor

VS

SCENARIOS FAVORING CUSTOM DEVELOPMENT

- Complex, unique workflows
- High or unknown demand for scalability and/or performance
- Unpredictable product roadmap
- Significant customization required now or in the future
- Limited need for reuse of application components

TCG'S APPROACH TO LOW-CODE VS. CUSTOM DEVELOPMENT DECISIONS

TECHNOLOGY-AGNOSTIC APPROACH

At TCG, our approach to helping Federal agencies make low-code versus custom development decisions is based on our Software Methodology Rules for TCG (SMRT) framework, which uses the Decision Analysis and Resolution (DAR) process to evaluate technology options using an objective and methodical approach. A DAR provides a structured mechanism for comparing alternatives based on predefined, weighted criteria such as requirements complexity, cost, timeline, and alignment with agency objectives. This approach aligns with the Decision Framework described above, where time, budget, and strategic goals are central themes.

Our philosophy is to remain brand- and technology-agnostic. This means our recommendations focus solely on the solution that best meets the agency's requirements, existing technology investments, and strategic objectives - not on promoting specific

vendors or platforms. TCG evaluates each scenario independently, tailoring our recommendations to the agency's context and the desired outcomes.

We've used the criteria described in the Decision Framework section to successfully implement both custom-developed and low-code solutions across various Federal agencies.

TCG developed a DAR matrix to score case management system options for the Nuclear Regulatory Commission using weighted criteria:

91% of requirements met by low-code

59% met by COTS products

52% met by custom development



FOCUS ON COLLABORATION

A hallmark of TCG's approach is our emphasis on collaboration with stakeholders. We work closely with Federal agency staff to define and refine requirements, leveraging techniques such as rapid prototyping, mockups, and wireframes, as well as validation techniques such as user acceptance testing (UAT), functional testing against defined requirements, and iterative demonstrations to stakeholders to ensure that solutions meet user expectations. By maintaining open lines of communication, we

minimize misunderstandings and allow for iterative improvements based on feedback.

TCG's iterative approach allowed MCC stakeholders to evaluate features of a high-priority Power App during two-week sprints. This collaborative process fostered greater stakeholder buy-in provided valuable insights for future sprints.



COMMITMENT TO FLEXIBILITY

TCG takes a flexible approach to involvement, akin to the "Home Depot" model: "You can do it, and we can help." This philosophy allows agencies to determine their preferred level of participation in the development process. Whether they wish to take a hands-on role or delegate tasks entirely, TCG provides the necessary expertise and support to guide them through every stage of the project.

For agencies interested in citizen development, we offer guidance and structure to empower non-technical users to contribute effectively. At MCC, we have supported the Power Platform citizen development community through recommendations, reviews of developed solutions, and proposing reusable templates.

UNDERSTANDING TOTAL COST OF OWNERSHIP

Agencies need to consider not only the up-front costs of development but also hidden expenses such as licensing fees, maintenance, and vendor support. By conducting thorough analyses, we help agencies avoid unexpected costs and optimize resource allocation.

We are committed to helping agencies evaluate total cost of ownership (TCO) and identify opportunities to save money throughout a solution's lifecycle.

CONCLUSION AND RECOMMENDATIONS

Federal CIOs and other leaders whose agencies are using or considering low-code platforms for application development have much to consider when deciding whether a particular business need would be best addressed using a low-code or custom-developed application. Objective assessments of strategic alignment, requirements complexity, time and cost constraints, and other key factors are necessary to make the best decision for the agency.

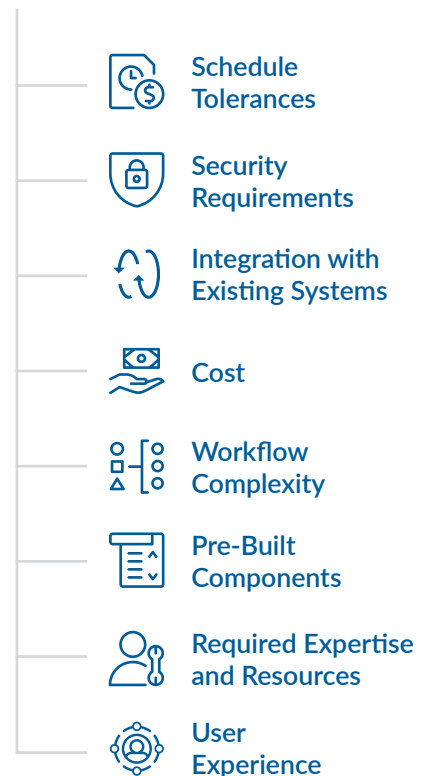
TCG recommends that Federal CIOs and other agency decision makers refer to the [Decision Framework](#) and subsequent sections above when faced with decisions on low-code versus custom application development, to ensure a comprehensive assessment of key factors. These factors should be incorporated into assessment tools like a DAR matrix to objectively compare development approaches.

TCG works each day to deliver innovative, effective low-code and custom software solutions in support of Federal agencies' missions, and helps IT leaders make informed decisions to save time and money.

Interested to learn more about how we can help your agency? Contact our low-code project manager or reach out for any other TCG questions at talktous@tcg.com or 202-986-5533.



What factors should be considered when evaluating whether a low-code or custom-developed solution is best?



For technical questions and more on how to implement low-code, please contact:

Tres Lewis
Project Manager
tres.lewis@tcg.com



To explore how this solution may fit your agency's needs, please contact:

Michael Drescher
Chief Growth Officer
michael.drescher@tcg.com



What are the key questions to help determine whether to use a low-code or custom software development approach?

SCHEDULE TOLERANCES



- How quickly do you need to see a minimum viable product (MVP)?
- How much do you want to test the system prior to going live?
- Is there a critical date the system must be working by?

WORKFLOW COMPLEXITY



- What are the number of approval levels?
- What are the automation requirements?
- What are the nuances of the required conditional logic?

SECURITY REQUIREMENTS



- Does your project require role-based access or field visibility based on user roles?
- Does your environment require any security reviews and approvals (such as Authority to Operate) for tools needed to build the application, such as Microsoft's Dataverse, or external libraries and dependencies for custom development, etc.?
- What is your organization's current security culture and how would this product adhere to it?

REQUIRED EXPERTISE AND RESOURCES



- Does the agency have the properly skilled development and support staff in place, as well as the technical tools (e.g. development tools, development environments, repositories) needed for the job?
- Will new staff need to be hired, or existing staff trained?
- Is there a willingness and aptitude in the organization to foster a citizen development approach where agency employees take on some development work?

PRE-BUILT COMPONENTS



- Is the add on or plugin from a trusted vendor or a lesser known source?
- What are the external dependencies used by the add-on?
- What is the reputation of the vendor that created the add on?

USER EXPERIENCE



- How comfortable is the projected user base with new technologies?
- Is the user experience more or less important than the functionality of the application?
- Are there styling and design standards for this application that must be met to ensure consistency with other tools in the portfolio?

COST



- Is your organization's IT budget dependable enough to forecast several years out?
- Are there any opportunities to find cost savings in other areas of the organization by implementing one solution approach instead of the other?
- Aside from regular maintenance, will this solution require enhancements and continuous improvement over the course of its lifecycle?

INTEGRATION WITH EXISTING SYSTEMS



- How does each approach fit within the existing IT portfolio of your organization?
- Are there opportunities or requirements at the start of this project for collaboration between existing tools?
- Is this new application going to replace, enhance, or provide distinctly new functionality in your organization?