



WHITE PAPER

The Role of Modular and Low/No-Code Development in Software Efforts

How aligning requirements to the correct approach can improve return on investment.

June 2023

Contents



Summary

A brief description of modular versus low/no-code approaches.

2



The Requirements Spectrum

The role of requirements in solution approach selection.

3



Development Solution Approaches

A summary of the alignment of requirements to solution approaches.

4



Low/No-Code Approach Summary

The general characteristics and attributes of low/no-code approaches.

5



Modular Development Approach Summary

The general characteristics and attributes of modular development approaches.

6



Example Efforts

Examples of a low/no-code effort and a modular development effort.

7



Northramp's Modular Approach

A summary of the attributes of Northramp's approach to modular development.

8



Northramp's Low/No-Code Approach

A summary of the attributes of Northramp's approach to low/no-code development.

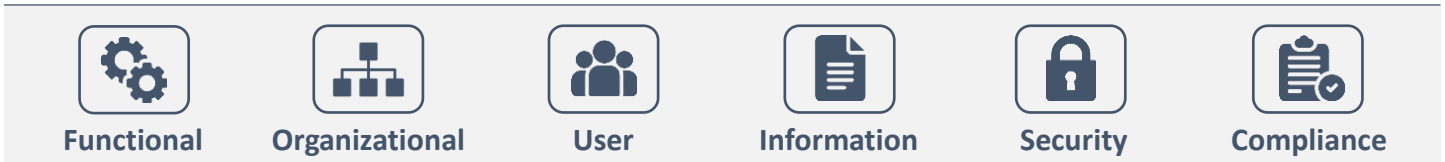
9

Summary

- The advent of modular and low/no-code development approaches has created a paradigm shift in software development - systems are no longer built from scratch using long development cycles and significant investment.
- When considering whether to use modular development or a low/no-code approach, an organization should consider various characteristics of the requirements, which help determine the appropriate software solution or approach an organization should follow to address their needs.
- Low/no-code solution approaches can be faster and more achievable for non-technical resources, but pose critical tradeoffs in flexibility, extensibility, and capability.
- A low/no-code approach used to address more complex requirements can result in a solution that may not adequately address the organization's needs - ultimately requiring additional time and cost to develop.
- Modular solution approaches are more flexible and capable but can require more advanced skill sets and longer development timeframes.
- Selecting a modular development approach to address simple requirements will deliver a functioning solution but may result in unnecessary investments in both time and money.
- Modular development is a frequently overlooked and misperceived approach, but when implemented, can serve as the optimal solution for addressing a range of software requirements.
- Effectively identifying the general characteristics of requirements is critical for selecting a solution approach that fully supports the Organization's needs without over investment.
- Northramp's modular software development approach delivers a proven and reliable option to address moderately complex and/or unique requirements.
- When leveraging our approach, organizations will realize faster time-to-value, lower investment levels, and higher user satisfaction over other approaches.

The Requirements Spectrum

Virtually every organization possesses a wide range of requirements to automate processes and capture and share information for insight and value. As illustrated below, these requirements typically possess a number of characteristics that help determine the appropriate software solution or approach an organization should follow to address their needs.



FUNCTIONAL SCOPE

Functional scope encompasses the tasks, processes, or operations that the solution will automate or assist. It also includes the complexity of these functions and how frequently they change, which directly influences the design and selection of the solution approach or platform.

ORGANIZATIONAL SCOPE

Organizational scope refers to the breadth of the solution's use within and across the organization. It affects design aspects such as data segmentation, protection, and the display of features and information.

USER SCOPE

User scope includes the number and types of users the solution will serve. It influences various aspects such as architecture and configuration flexibility. The scale of users can also affect decisions regarding hardcoded requirements and the value of investments in user-driven configurability.

INFORMATION SCOPE

Information scope encompasses the depth and volume of data the solution will handle. The interrelationships between information sets, within and external to the solution, also play a crucial role in solution design and selection. Data complexity, when combined with organizational and user scope, can substantially impact the suitability of different solution approaches.

SECURITY AND COMPLIANCE SCOPE

Security, legal, and other compliance scope characteristics can also drive a range of additional requirements. The sensitivity of the information the solution will handle as well other required controls often dictate important solution decisions. Additionally, the solution may be required to adequately address a variety of industry and governmental regulations such as data privacy, change control, audit trails, and accessibility.

SUMMARY

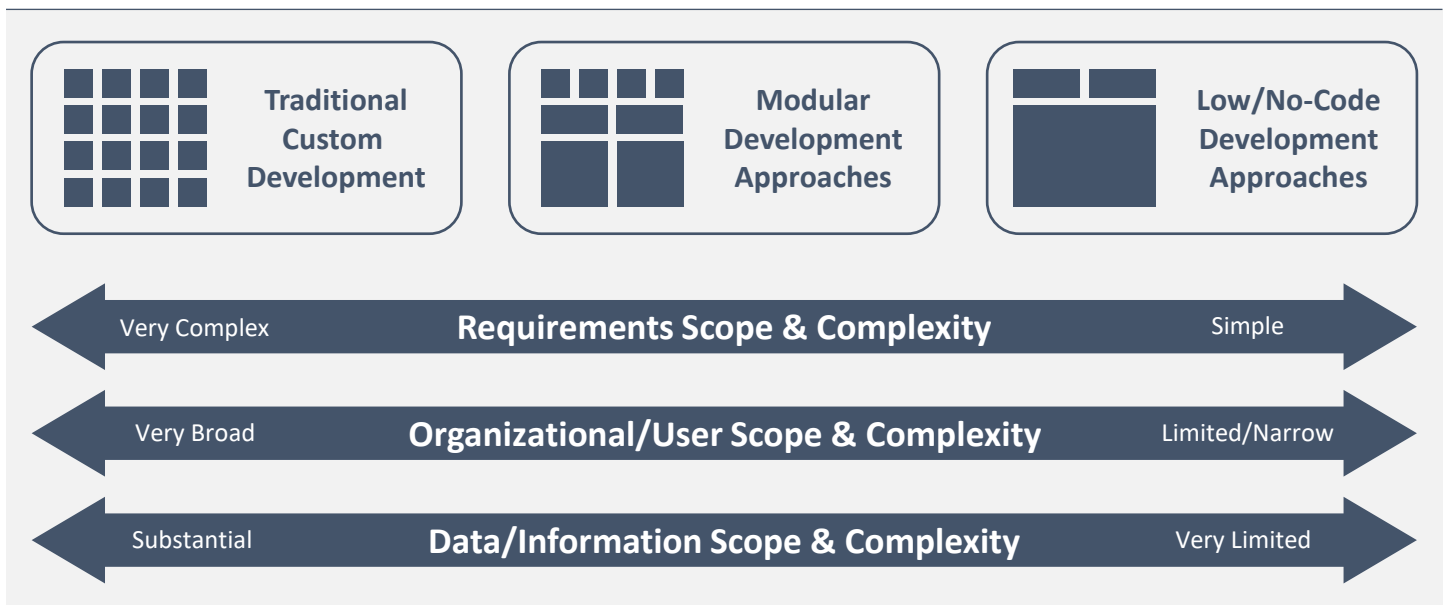
These 'requirements characteristics', as well as others such as performance and scalability needs, act as the primary drivers for the selection of an appropriate solution type and path. From 'simple to complex', 'few to many', or 'low to high', effectively identifying the general characteristics of requirements is critical for choosing a solution approach that fully supports the requirements without over investment.

Development Solution Approaches

Over the last decade, traditional custom software development, with its lengthy and intricate coding processes, underwent a significant transformation. The advent of modular and low/no-code development brought about a paradigm shift in software development.

In contrast to traditional custom development, in which developers build complex systems from scratch by meticulously writing lines of code (often resulting in long development cycles, high costs, and limited flexibility), modular and low/no-code development empowers individuals and organizations to create software solutions rapidly and efficiently. These approaches leverage pre-built modules, components, and interfaces that allow users to assemble applications like building blocks. This modular nature enables the reuse of existing functionalities, saving time and effort.

However, as illustrated below, when making decisions about whether to leverage a modular or formal low/no code approach, it is important to carefully consider the nature and scope of the requirements. Each approach is best suited to address a specific level of complexity and scope, with trade-offs in time and investment.



Deciding upon the right development approach is critical for the success of any custom solution. For example, selecting a modular development approach to address a simple set of requirements will deliver a functioning solution, but may result in unnecessary investments in time and money when a low/no-code approach could have more readily addressed the requirements. Alternatively, attempting to leverage a low/no-code approach for more complex requirements could result in a solution that does not adequately address the organization’s needs, which may ultimately require greater cost and time investments for development due to the number of design considerations and unique customizations based on the limitations of the chosen platform.

The following sections further summarize the benefits and drawbacks of each approach as well as their common use cases.

Low/No-Code Approach Summary

LOW/NO-CODE DEFINITION

Low/no-code development is an approach that enables the creation of software solutions with minimal or no traditional coding. Using a visual interface provided by a platform, low/no-code development aims to simplify the application development process, allowing individuals with limited or no programming knowledge or technical skills to build functional applications. These platforms typically provide drag-and-drop interfaces, visual modeling tools, and pre-configured templates that enable individuals to design user interfaces, define business logic, and integrate with data sources.

BENEFITS

- ✓ **Faster development** cycle times between idea and usable software
- ✓ Solution capabilities are **more accessible** to non-developers
- ✓ **Increased productivity** of developers who can focus more substantially business needs versus coding
- ✓ **Reduced complexity** and maintenance activities driven by more operations handled by the underlying platform
- ✓ **Lower lifecycle costs** from shorter development times and fewer maintenance activities

DRAWBACKS

- ➖ **Limited customization and flexibility** compared to traditional coding makes it challenging to address more complex functional requirements
- ➖ **Difficulty handling more complex data models** and transactional integration with other systems
- ➖ **Difficulty establishing and maintaining UI,UX, and other technical standards** across diverse development ecosystem and devices
- ➖ **Increased fixed costs and vendor lock-in** from continuous investment in underlying licenses
- ➖ **More difficult to apply fine-grained user rights** to functionality and data

APPROPRIATE/FREQUENT USE CASES

- **Data Visualization and Dashboards** - By simplifying data integration and visualization tasks, low/no-code platforms enable users to rapidly create insightful dashboards without extensive coding expertise.
- **Web Content-Oriented Apps** - With built-in modular web functions for page design, content editing, list creation, and simple forms, low/no-code platforms are well-suited to address these use cases.
- **Individualized Tools and Process Automations** - Using a low/no-code platform, individual users and small groups can automate repetitive processes and share data in a substantially more efficient manner.
- **Rapid Prototyping / Minimally Viable Product (MVP) Solutions** - A low/no-code environment is often an effective approach for iterating on a potential solution or testing solution concepts with business users.

Modular Development Approach Summary

MODULAR DEVELOPMENT DEFINITION

Modular development is a software development approach that involves addressing the requirements of an application using smaller, independent modules or components. Each module is designed to perform a specific function or task and can be implemented and tested independently of other modules. This approach enables code reusability, maintainability, and scalability. By segmenting larger software efforts into smaller, self-contained modules, developers can focus on designing and implementing individual components enabling parallel development and easier collaboration among team members.

BENEFITS

- ✓ Allows for **greater flexibility and scalability** which is suited for more complex or large-scale projects that require unique functionalities
- ✓ Enables **highly tailored applications** that require specific processes, or deeper integrations with external systems
- ✓ **Increased development efficiency** over traditional methods by enabling significant code reusability
- ✓ **Allows for greater technical control**, enabling use of a variety of frameworks and libraries to address specific requirements

DRAWBACKS

- ✖ **Demands increased development skills** and experience to successfully address functional requirements
- ✖ **May require longer development timeframes** for simplistic internal web apps (e.g., single data source, serial workflow, small user base).
- ✖ **Increased maintenance complexity** based on code, framework, and library updates
- ✖ **Increased upfront costs** for the initial development of the base code
- ✖ **Requires additional infrastructure** to support traditional dev/test/prod environments

APPROPRIATE/FREQUENT USE CASES

- **Business Process Automation** - Modular development enables an organization to build extensible multi-user, secure, process automations that precisely meet their business process/workflow needs.
- **Data Management** - Due to its flexible nature and relational data capabilities, a modular approach enables an organization to manage complex datasets, relationships, and integrations with other systems.
- **Customer-Facing Systems** - The flexibility of modular development enables an organization to carefully architect a user's experience (UX) and address a variety of use cases within the system and user interface.
- **Specialized Requirements** - Addressing highly unique requirements, such as the creation of specialized outputs, fine-grained security, or internal controls, are best addressed using modular development.

Example Efforts



LOW/NO-CODE APPROACH EXAMPLE



The US International Development Finance Corporation lacked easy to use visualizations and data entry systems. Northramp used the Microsoft Power Platform to automate processes, normalize data capture and create visualizations to help with creating data driven decisions.

Requirements Characteristics

- Small user community
- Automating manual tasks
- Saved data to a database
- Did not require a material level of security or other controls



Approach and Outcome

- Used canvas to capture data into a database
- Used Power Automate to automate manual and repeatable data entry scenarios and workflows.
- Used PowerBI to provide improved information visibility
- ✓ Substantially improved their ability to make data driven decisions



MODULAR DEVELOPMENT APPROACH EXAMPLE



The U.S. Nuclear Regulatory Commission (NRC) possessed a unique requirement to publish information in a dynamic and visualized manner to its publicly-facing website. Additionally, the information only existed previously in unstructured formats (documents). To address this requirement, Northramp developed a simple data management application that stored information in a structured format, enabled the publishing of data to the externally-facing website, and, as a secondary benefit, automated related processes.

Requirements Characteristics

- Office-level user communities
- Act as a data source for business information
- Maintain information and security controls
- Develop unique outputs for publishing



Approach and Outcome

- Used MS .NET and SQL Server to develop solution
- Used modular approach to accelerate delivery and minimize investments
- ✓ Enabled highly controlled publishing of dynamic data and materially improved external stakeholder satisfaction

Northramp's Modular Approach

SPARCHENGINE™

Most software applications are essentially made up of the same basic components applied to different business domains - forms to add, edit, or view information, reporting, notifications, security, administrative functions, and so on. Because of this, Northramp developed a collection of stable, custom code components which we call our Solution Pattern Architecture (SPARCH) to enable organizations to rapidly and efficiently build and deploy software when a modular development approach is appropriate.

RELIES ON OPEN AND STANDARD TECHNOLOGIES



All of Northramp's development efforts leverage open and standard technologies from Microsoft and its ecosystem which simplifies resourcing, operations, and maintenance requirements and avoids vendor lock in.



TESTED AND MATURE BASE CODE

Our base code has been tested, refined, and optimized over hundreds of efforts to work out virtually every kink - all before an organization spends a dime on development.



PRE-DEVELOPED APPLICATION AND INTERFACE COMPONENTS

SPARCH components include user interfaces, security and administrative controls, user help capabilities, data visualizations, reporting, business rules, notifications, and more.



NO LICENSES OR ADDITIONAL INVESTMENTS

Once we build the solution, the code belongs to the organization. Nothing developed is proprietary or pre-compiled, the result is simply code using industry standard technologies and a truly well-functioning solution.



EASILY DEPLOYED ON EXISTING INFRASTRUCTURE

Because of the nature of the technologies, virtually every organization already possesses the compute and storage capacity to readily deploy the resulting solutions into their environments.

SUMMARY

Northramp's modular software development approach is a proven and reliable option for the development of software to address moderately complex and/or unique requirements. When leveraging our approach, organizations will realize a faster time-to-value, lower investment levels, and higher user satisfaction over other approaches.

Northramp's Low/No-Code Approach

Northramp is committed to empowering organizations by leveraging our expertise in low/no-code technologies. We recognize that in the rapidly evolving solutions development landscape, agility and adaptability are critical. Consistent with this, we provide a comprehensive suite of low/no-code solutions across a multitude of platforms that are customizable to our client's unique business requirements.

POWER APPS



Microsoft Power Apps is a suite enabling rapid deployment of custom applications which connect data stored in underlying data platforms (e.g., SharePoint, Microsoft 365, SQL Server). Northramp has leveraged Power Apps to standardize and synthesize reports, integrate data from disparate sources, and provide executive level views to facilitate data driven decisions.

CANVAS



A feature of Power Apps, Canvas allows users to design and build business apps that integrate business data from a variety of sources. Northramp is currently building a variety of single and multi page low complexity applications in Canvas. Using Canvas enables straightforward development while maintaining data normalization and integrity that is otherwise lost in MS Excel or other data capture tools.

POWER BI



Power BI is a business intelligence tool provided by Microsoft which combines business analytics, data visualization, and best practices that creates insight and helps organizations make data-driven decisions. Northramp delivers both cost savings and the expedited delivery of dashboard views and reports by leveraging Power BI, rather than in the underlying systems themselves. These visualizations and reports can also replace more complex options while remaining directly embedded in applications.

SERVICENOW



ServiceNow serves as a robust low/no-code development platform, empowering organizations to accelerate their digital transformation. Leveraging its interface and pre-built functionalities, users can design and deploy tailored applications without the need for traditional coding expertise. Northramp currently support and oversees the development and deployment of workflow applications within ServiceNow at a number of organizations.

SUMMARY

Northramp possess the capacity and capability to leverage low/no-code development platforms and approaches to deliver highly tailored applications that meet the unique needs of our clients when requirements align with their use. Our approach combines traditional agile practices which further enhance and accelerate delivery while avoiding over investment in development activities.