Low-code development in fusion teams

A best practice guide

A PEGA WHITEPAPER

Introduction

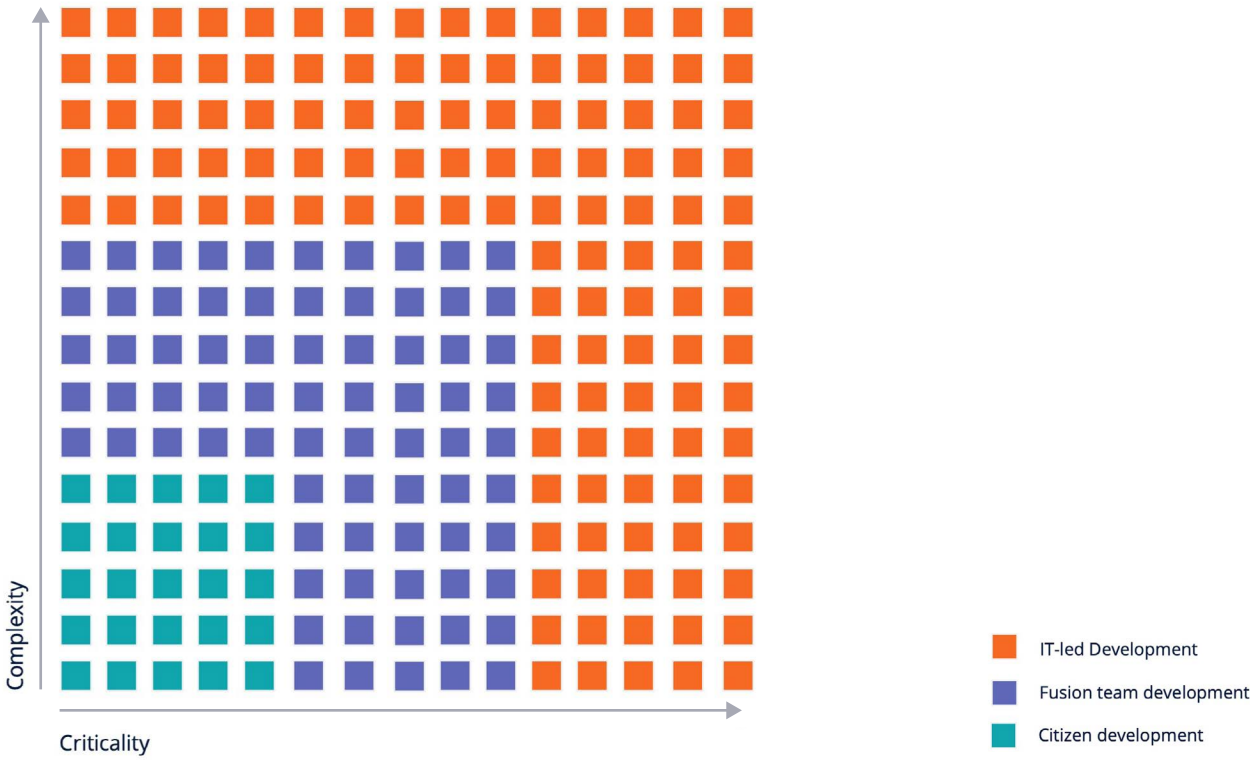
Low-code development falls along a continuum.

On one end of that continuum, individual tech-savvy business users (which Gartner terms citizen developers) can build workflows that are low in complexity and have limited risk potential. Pega's **Low-code factory approach** is an ideal operating model for these use cases because it empowers citizen developers working either as individuals (makers) or in squads to build apps quickly and easily – in ways that reduce IT backlogs without introducing additional risk.

On the other end of the continuum, professional developers can use low-code tools to accelerate the creation of complex and critical enterprise applications while reducing long-term maintenance costs. For these use cases, the **Pega Express™ Delivery approach** provides a toolkit of best practices for using Pega technology to bring business and IT together through a combination of agile and design thinking principles.

But what about everything else on the continuum? What approach is best when it comes to building applications and workflows that are too critical or complex for a maker to build and maintain, but not critical enough to warrant prioritization by enterprise IT?

Low-code development continuum



The Pega Platform™ is uniquely capable of supporting low-code development across the entire continuum of low-code use cases. This whitepaper will provide guidance for navigating department-led low-code development in citizen development squads and fusion teams.

		Department-led citizen development (in squads)	**Fusion team development**	**IT-led development**
Complexity	**Functionality.** Create data models, functionality, and front-end	Low complexity	High complexity	High complexity
	Integrations. Integrations with larger IT landscape including databases and applications	OOTB and pre-built integrations	Involves custom integrations	
	User Interface. Create the user experience	OOTB	Some custom	Highly customized
Risk	**Security.** Potential breaches of confidentiality, data loss, and uncontrolled user access.	Low	Medium	High
	Operational IT. Acceptable rules to create, manage, maintain, and use IT			
	IT Architecture. Rules to build, modify, and interface IT resources			
	Regulatory. External laws, industry standards, or compliance requirements			
	Reputational. Potential loss of brand perception, social capital, or market share			
	Financial. Potential impact on revenue			

[2] Adapted from the Citizen Development the Handbook for Creators and Change Makers. Project Management Institute, Inc, 2021

Fusion teams

Gartner defines a fusion team as "a multidisciplinary team that blends technology or analytics and business domain expertise and shares accountability for business and technology outcomes."[2] A fusion team approach has a number of advantages for departments as a complement to IT-led and citizen development low-code initiatives:

- **Flexibility.** The model allows teams to come together in an *ad hoc* capacity with resources assigned based on the nature and complexity of a project.
- **Cost.** The relatively simple nature of most workflow automation apps that a department needs to produce means the number of full-time resources required for a departmental low-code program will be less than what would be necessary to build and support applications within enterprise IT. And because applications are built mostly in Pega's App Studio, they do not require highly trained professional developers (which are costly and hard to find). According to Forrester's TEI study, the cost of department-led low-code efforts is 34% less than IT, and 44% less compared to the cost of using outside resources.
- **Time to value.** The number one reason that departments seek to build out their own low-code development capacity is that their need for agility is not being met by enterprise IT. Given the scarcity of professional development resources, IT can often only prioritize critical applications that are expected to deliver the greatest amount of ROI from the organization's perspective as a whole. The result: IT backlogs often include a multitide of departmental requests for simple to moderately complex workflow automations that are deprioritized in the face of larger projects. By using their own resources to build upon a low-code platform that is sanctioned and supported by IT, departments are able to prioritize and accelerate projects that drive significant value.
- **Continuity.** Although fusion teams themselves are multidisciplinary, responsibility for ongoing maintenance and support of applications typically falls to the department. Because fusion teams build using a low-code platform that IT has approved of, they can focus on maintaining their applications without also being responsible for maintaining the platform itself.
- **Risk.** The most important factor determning whether to form a fusion team is the potential for a new application to introduce risk. Highly correlated with business value, risk may relate to security, operations, IT architecture, regulatory exposure, reputational harm, or financial impacts. Where a project taken on by a citizen development squad can be successully mitigated by adding specialized resources from IT and/or some other part of the organization, cost of delivery can remain low even as business value remains high.

A successful department-led low-code program needs to carefully consider the right people, processes, and technology. What follows is a set of best practices and recommendations in each of these key areas.

[2] "Gartner Glossary," https://www.gartner.com/en/information-technology/glossary/fusion-team.

The right people

A fusion team approach places responsibility for building and maintaining applications on the department rather than on IT or individual makers. Shifting responsibility in this way serves to reduce the burden on enterprise IT while eliminating the risk of orphaned applications.

Fusion teams are typically created in an ad hoc fashion, according to the complexity and criticality of the applications being created, but the most effective department-led initiatives are staffed with a small number of dedicated resources who work collaboratively with departmental stakeholders to build moderately complex applications.

Important roles to include as part of a successful departmental low-code delivery program include:

Program management and governance

- **Practice manager.** A practice manager is responsible for assessing the fitness of application requests, ensuring that solutions do not already exist, and collaborating with the center of excellence (COE) to prioritize and build shared components for commonly requested features and integrations. They also bear responsibility for prioritizing work based on potential business impact and assigning resources based on project complexity and resource availability.

Fusion team resources

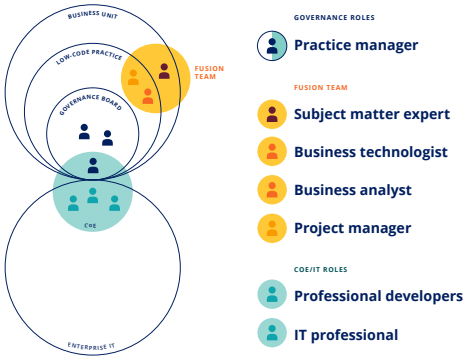
- **Project manager.** A project manager is essential for ensuring that delivery is completed on time, in scope, and within budget. It is best practice for project managers to follow an Agile approach, using something like Pega Agile Studio to coordinate activities, collect key project artifacts, and focus on the delivery of a minimum loveable product (MLP). A minimum lovable product is different from a minimum viable product, which focuses on achieving something feasible. An MLP delivers a solution that is not only viable but desired and embraced by end users. An MLP is packaged to quickly deliver business outcomes in a way that delights customers and makes their lives easier.
- **Business technologist.** A business technologist is the low-code solution developer who builds the application or workflow. Because the applications they create could potentially be more complex or critical than a maker-led use case, and because they work as a member of a project team, they should be certified as a Pega System Architect. A mature program will have at least one senior system architect in the department and have access to a lead system architect through the organization's COE for coaching on more complex projects.
- **Business analyst.** The business analyst is responsible for gathering requirements from key stakeholders and translating those requirements into app specifications for the business technologist. Best practice is to capture objectives directly in App Studio to produce a working prototype and mitigate the risk of miscommunication. We recommend Pega Business Architect certification so analysts can most effectively translate between the needs of the business and the capabilities of the Pega Platform. The most effective business analyst will have deep hands-on domain expertise relevant to the needs of departmental stakeholders.

- **Subject matter expert.** The subject matter expert is a representative of the business function for which the application was requested. They are responsible for representing the needs of their function to the business analyst and ensuring, alongside the business analyst, that the built application is solving the right business challenge and will drive the expected business value.

When considering roles in a fusion team, remember that separate roles do not necessarily mean separate people. For simple workflow automations, the roles of project manager and business analyst may be performed by the same business technologist responsible for building the application. Or it may be that the role of the project manager is condensed into the business analyst function. In other words, the makeup of the fusion team is less important than the functions it represents.

Center of excellence

- **Professional developer.** While citizen development squads should have the skill to build most moderately complex applications without the need to engage a professional Pega developer, a professional resource may occasionally be required for coaching on more sophisticated techniques. For the most part, however, the role of professional developers through the COE is to build reusable components and establish guardrails in such a way as to avoid the need for active, hands-on support by professional IT/developer resources. Focusing professional development efforts in this way instead of actively co-developing applications means that low-code development velocity within departments can increase while also freeing up professional resources to focus on more complex and critical applications. Where active involvement is required from a more seasoned developer, a fusion team is formed, but delivery costs are minimized because most of the core functionality of the application is still designed and built by the squad.



- **IT professional.** IT professionals are required to manage the CI/CD pipeline, responsible for integration testing, performance testing and tuning, and deployment as required.

CASE STUDY

Citizen development in fusion teams

A large U.S.-based credit union launched a formal citizen development program to counter growing shadow IT development and risks inherent within it.

The credit union chose Pega because it is a platform that would support a common standard and governance model – ensuring that anything implemented has gone through the appropriate rigor and would not impact the core systems. The initial focus was on internal solutions and back-office automations; external-facing applications were out of scope.

To get started they created a COE in partnership with IT to support the program, provide guidance/coaching, establish a foundation of common reusable technical assets, and maintain the infrastructure. Fusion teams are typically comprised of three people – a business analyst, a business technologist, and a project manager. All three are permanently allocated to the fusion team; are sourced from the business, not IT; and have deep domain experience – a game changer for them as these resources can understand the problems and required solutions extremely well.

With Pega RPA and the Pega Platform, the credit union's citizen development program is responsible for saving the organization over 300,000 hours each year.

The right processes

Application intake

The purpose of application intake is to:

1. Ensure the requested application is a good fit for departmental delivery.
2. Verify that there is not an existing solution within the IT landscape capable of solving the business problem.
3. Prioritize requests against the current backlog.
4. Assign team members based on project complexity, risk, member availability, and required skill.

The intake process starts when a stakeholder submits a request through Pega App Factory. Practice manager(s) evaluate the request to ensure that it is an appropriate departmental use case (i.e., the business need is well understood, there's not already a solution for the business problem, its level of criticality doesn't require IT ownership, etc.). Requestors should be asked to submit a simple business case, including an estimated ROI for the project. This will help practice managers decide whether to approve the build and how to prioritize effort relative to other requests.

Once a project is approved, the practice manager will assign resources based on the complexity of the project. At the very least, this will involve assigning a business technologist to partner with the subject matter expert to complete the build. But for more complex projects, a dedicated project manager and business analyst may also be required, in addition to a cross-functional resource(s).

Delivery

The Pega Express™ Delivery approach combines scrum and design thinking to provide an incredible toolkit of best practices for enterprise low-code delivery – through every phase of a Pega project, from discovery and preparation to building and adoption. A thorough understanding of these principles (like direct capture of objectives, Microjourneys®, MLPs, etc.) on the part of regular project team resources will ensure that projects quickly deliver meaningful outcomes.

Testing

Once an application is built and approved by the project sponsor, a deployment pipeline is provisioned so that testing by a senior development resource (either in the department or out of the IT-based center of excellence) can commence. The testing process for each application may include the following, depending on app complexity:

- **Guardrail score validation.** A guardrail score of more than 85% should be required before the release of an application, and exceptions should be reviewed to ensure that the app performs as expected while using resources as efficiently as possible. Guardrail scores may be viewed via the Application Quality Dashboard in Dev Studio.
- **Unit testing.** The purpose of unit testing is to verify that each element of the application (e.g., a decision table or a report definition) works as expected. Unit testing reduces the risk of a configuration error in one rule being propagated to other rules in the application, which causes significant delays to case processing. Unit testing should be continuously performed by the business technologist to ensure that what they are building meets business requirements.
- **Functional testing.** To ensure it functions as expected, application features should be tested individually by someone who was not involved in the application build itself. In most cases, the subject matter expert will be the best person to conduct functional testing.
- **App Studio compliance.** To ensure that the application can be automatically updated with Pega Platform updates, any application created in ways that involve the use of Dev Studio should be reviewed to ensure alignment with best practices. The App Studio Compliance tool in Dev Studio makes it easy to test from a UI perspective.
- **Other testing.** Depending on project complexity and standard operating procedures recommended by IT, additional performance, integration, and security testing may be required. Practice managers should coordinate with the COE to ensure that testing complies with organizational requirements.

The right technology

The key pieces of Pega technology required for a successful departmental low-code program are Pega App Factory, App Studio, Dev Studio, and Agile Studio.

- **Pega App Factory.** A lightweight and highly customizable tool for managing application requests, approvals, and onboarding. Pega App Factory includes valuable features to facilitate communication between project team members during the build process and users to gather feedback once the application is live. An internal app store feature makes it easy for department members to request access to applications. Tight integration with Pega's Deployment Manager means that practice managers can easily manage the entire application lifecycle from request to deployment.
- **App Studio.** Pega's low-code application development studio is where every build should begin. Best practice is to develop an application that will be built from start to finish without leaving the App Studio environment and that can augment App Studio functionality through shared components.
- **Dev Studio.** Pega's low-code authoring environment for professional Pega developers, Dev Studio, provides access to a wide range of advanced automation and case types. Dev Studio's flexibility allows you to do almost anything that could be done in standard code; it is vital that users are trained and certified to follow best practices. As part of a robust reuse strategy, Dev Studio is where professional developers create reusable components that can be easily consumed by citizen development squads and other teams in App Studio.
- **Agile Studio.** Many organizations have standard project management tools like Jira – and the Pega Platform integrates easily with many of them. However, many may choose Agile Studio as the project management tool of choice for departmental app creation.

While departmental low-code programs may run on the same Pega Platform deployment as an organization's enterprise applications, many organizations prefer to manage departmental programs on a separate deployment. Managing low-code programs on separate instances of the Pega Platform serves to mitigate performance risk to highly critical enterprise applications, with visibility across programs made possible through Pega App Factory and Pega Process Fabric.

It is best practice in most cases to host departmental low-code programs on Pega Cloud® to ensure that citizen developers, business technologists, and users all receive the best possible experience – while also reducing IT burden associated with manually maintaining multiple Pega deployments on-premises.

CASE STUDY

Citizen development at Deutsche Bahn

DB Systel GmbH, the digital partner of all Deutsche Bahn (DB) AG companies, worked with Pega to develop a software delivery model to build simple apps quickly and on a tight budget. In order to achieve Deutsche Bahn's goal of digitizing hundreds of business processes over the next few years, DB Systel built a program that uses citizen developers to build simple applications, while fusion teams focus on more complex use cases.

The most important component of Deutsche Bahn's low-code program is training. All citizen developers and members of the fusion team(s) are screened for basic skills. To support an efficient way to understand and capture processes, DB Systel set up an eight-step intuitive approach that focuses on process understanding, case lifecycle design and basic data modeling. The results enable quick realization of applications using Pega's low-code platform. The approach is consumable and requires only a limited time – learning the basics takes only a few hours.

In less than a year, Deutsche Bahn has already implemented three applications and another 18 are either ready to go live or in development. The first Microjourney in Contract Management reduced the manual workload in this department by 50%.

Although it is a relatively new program, the news about the first success in a single department spread quickly. The program has now been extended to several other business units and will help to drive the digitalization of Deutsche Bahn successfully.

Conclusion

When considering the right delivery model for your organization, it's helpful to remember that there is no one-size-fits-all solution. What this means is that the right operating model for departmental app production at a particular organization is likely to be hybrid, consisting of elements of both the Pega Low-code Factory and Pega Express approaches.

Pega clients who have seen the greatest success with this kind of program have something in common – a partnership with a trusted third party that can help craft a nuanced operating model that is the right fit for the organization and assist with the adoption to lessen time to value.

About Pegasystems

Pega provides a powerful low-code platform that empowers the world's leading enterprises to Build for Change®. Clients use our AI-powered decisioning and workflow automation to solve their most pressing business challenges – from personalizing engagement to automating service to streamlining operations. Since 1983, we've built our scalable and flexible architecture to help enterprises meet today's customer demands while continuously transforming for tomorrow.

For more information, please visit us at **pega.com**