Building your Low-Code Factory

A best practices kit for success

Why app dev has gone low-code

The traditional approach to application development cannot keep pace with enterprise demand for rapid transformation and productivity gains. Skilled developers are scarce, but in high demand. And those who are available are more likely to spend the **majority of their time** maintaining code. As a result, business users are being increasingly involved in enterprise app development. **Gartner predicts** the number of active citizen developers at large enterprises will be at least four times the number of professional developers by 2023.

Low-code application development platforms are here to help. These SaaS solutions enable staff with little to no coding experience to create apps, bridging the gap between business needs and the shortage of skilled developers. They facilitate rapid development, are less expensive to customize in house, and don't require an army of engineers. Low-code provides citizen developers, or "makers," with visual drag-and-drop tools and empowers them to collaborate with IT.
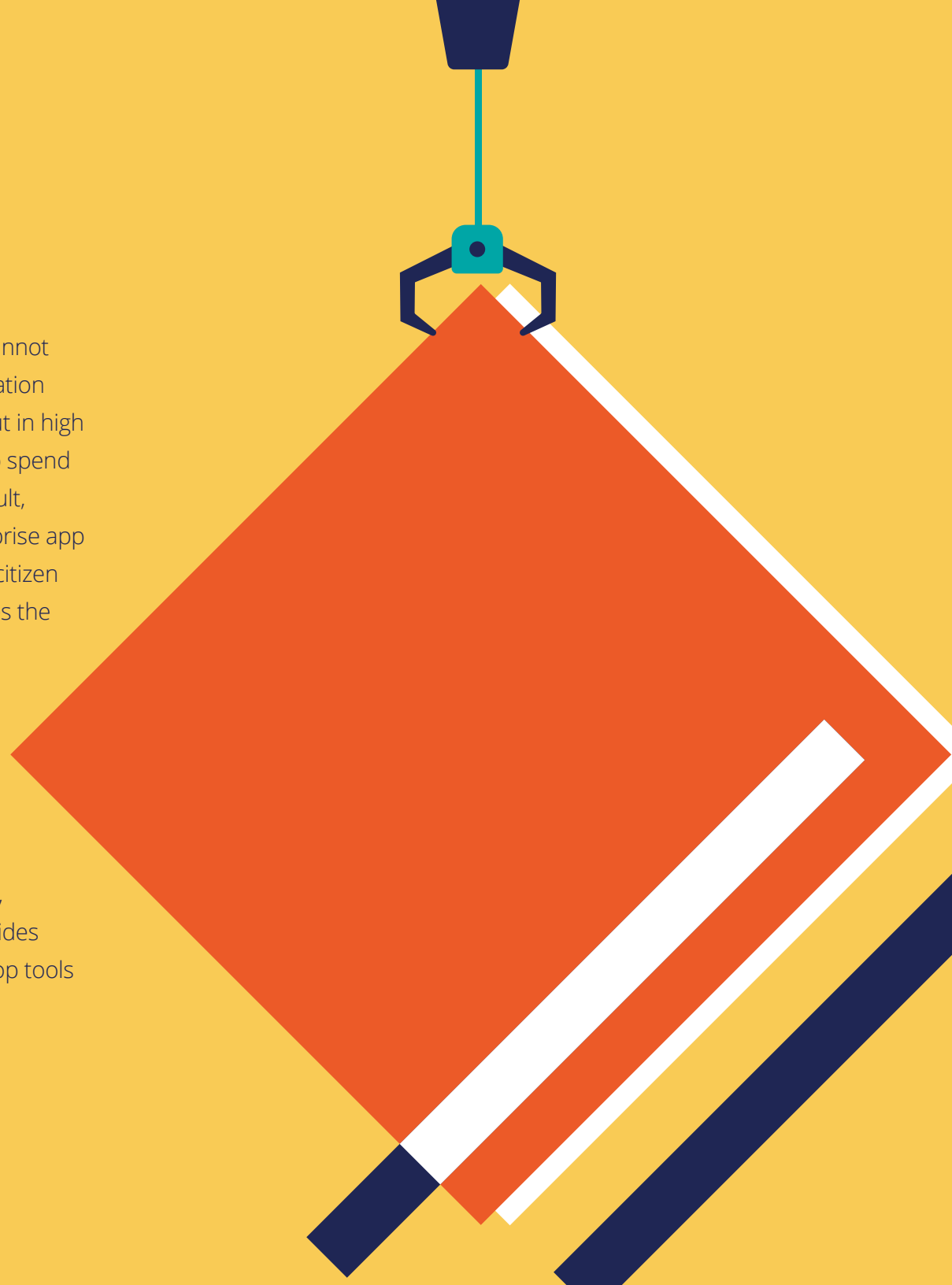
To win at low-code, you need strategy

Low code doesn't automatically guarantee success. Some platforms only support simple, lightweight applications that cannot scale and only offer limited reusability. Others have disconnected authoring environments that perpetuate silos. Both types cause headaches forv organizations as they waste time maintaining apps created in these disparate environments.

Even with the right end-to-end solution, proper strategy is required so IT can streamline and govern innovation in scalable, secure, repeatable environments. Building an app, whether it's simple or complex, shouldn't be a struggle.

What is the low-code factory approach to development? At its core, it's a methodology that offers:

1 **A framework for success**

Build an environment that encourages reuse and collaboration with IT to ensure quality, scalability, and sustainability.

2 **The right processes and governance**

Low-code tools – and their related processes – enable business experts to rapidly build apps without writing code. And the proper guardrails will protect your business users' creations and the outcomes they drive.

Embracing a low-code factory approach for the enterprise

A low-code factory approach allows business experts to rapidly build apps without writing code, while providing governance frameworks so IT can safeguard projects. Some projects may start and remain small, delivering tactical results. Others may be more complex and bridge multiple products, departments, and geographies.

Delivering your first low-code project might not require a factory approach. In fact, it might just take someone with an idea and a little bit of online training. To evolve from one-off success to repeatable transformation requires best practices, the right skills, business and IT collaboration, and a framework for reuse and scale. That's what the low-code factory approach is all about.
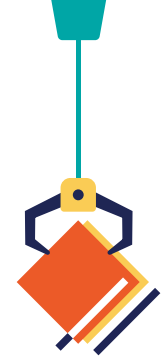
Constructing your team for now – and the future

Like anything, fostering low-code development requires the right team working together to achieve a shared outcome.
A successful rollout means understanding the roles that make up a strong factory team.

Makers
Business people or citizen developers. Makers understand business outcomes and the processes, people, data, and logic that drive those outcomes.

Coaches
IT developers or proficient makers. Coaches are skilled in low-code development and guide makers through the development process.

Practice managers
IT leaders or employees with knowledge of the organization's application landscape and best practices. Practice managers oversee all low-code development.

Professional developers
IT engineers or front-end experts. Professional developers create reusable low-code components and frameworks and tackle technical obstacles in the application development process.

Makers need business requirements knowledge and technical aptitude and curiosity.
To identify good candidates for citizen development, you'll need to qualify them:

Do they have a deep understanding of business objectives?
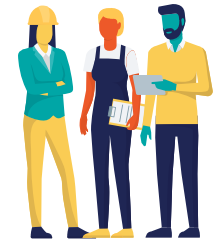
Are they savvy with spreadsheets?

Have they ever built their own webpage or site?

Are they proactive?

Do they enjoy mentoring others?

Are they nimble and eager to learn?

Creating a Community of Practice

Once your core team has delivered a few apps, establishing a low-code Community of Practice (CoP) is the next step to scale your factory. A CoP is a collective of people who create and benefit from shared best practices and tools. A mature CoP is self-directing and can support both makers and IT in successful project completion.

A CoP reduces business risk and organizes onboarding, support, enablement, and resources to position the team for success.

For example, many CoP's establish a lightweight "request for approval" process for new apps. This captures the business objectives, technical considerations, and profile of the maker submitting the request. The CoP can then identify opportunities to provide reusable components, tailor maker support, reduce duplication of work, and review the risks.

This process may include gathering information such as:

Application purpose

ROI or metric-based KPIs to measure success

Number of users and time per day spent in the application

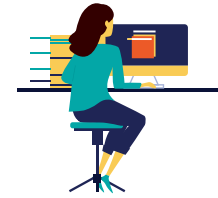Level of functional capability required in the application

Features on lockdown for certain use cases

Application data model

Existing system and legacy data interface and integration

Personally identifiable information (PII) that may be used as part of workflows

Final form factor (mobile, web, both) of the application

Developing maker support

Apps built solely by IT tend not to meet user expectations. Why? Without outcome-critical business knowledge, apps fail to achieve core objectives and face adoption challenges. When business units are meaningfully involved, all stakeholders are empowered to deliver better results. Factories enable this model at scale, using a balance of democratization with guardrails to mitigate risk and ensure quality.

Successful makers are empowered through initial training and continuous coaching. Consider the following best practices to recruit, engage, and retain your best makers:

1. Collaboration and engagement
Forums enable fast ad hoc support from a community of knowledgeable experts in an open, collaborative format. Makers and developers alike can find solutions, ask questions, provide feedback, and share tips.

2. Certification
Pega Certifications build credibility and confidence for makers by validating expertise. Data shows that certified teams deliver more business impact as their proficiency improves QA and minimizes errors – all while accelerating time to market.

3. Continuous learning
By establishing a culture of voluntary, self-motivated learning, you can drive competitive edge, productivity, and profitability[1].

4. Badges and awards
In addition to certifications, badges are digital representations of demonstrated skills and progress that can be shared socially. They're a great way to incentivize learning and elevate your team's social reputation.

5. Self-paced learning
Business users may need additional guidance with design, data architecture, naming conventions, testing, governance/access controls, and security and policy compliance. Offer the flexibility for makers to learn at their own pace in the format that works best for them.

6. Support and feedback
Allow all stakeholders to submit tickets for general support, new features, and application performance review and optimization. By scheduling weekly office hours with cross-functional experts – both in IT and business – you can capture feedback and avoid communication pitfalls.
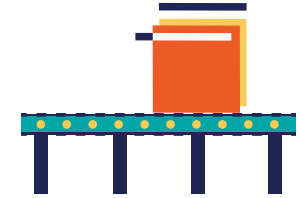
Defining a process framework

You've assembled the right team to carry out responsibilities and support objectives. They're trained and ready to build. Next, you'll want to identify which projects are appropriate for low-code development.

Identify the use case – and its future path

Great low-code candidates are those that have stakeholder consensus that a pain point requires urgent attention and rapid results. These projects will start as small solutions for a specific challenge, but can be scaled to address cross-functional and organizational problems. By planning for scale, you'll solve challenges more effectively, build for reuse, limit throwaway apps, and reduce duplicative development time.

Define the end-to-end process

At the core of every successful app are clearly defined processes, microjourneys™, and outcomes. This foundation provides the structure for makers to build in an iterative manner: focus on one step of the journey at a time, and still keep an eye on the bigger picture.

Focus on the three pillars

Every business process comprises three components – data, people, and microjourneys. Once you define microjourneys, consider which users – both internal and external – will interact with the app and at which stages. Afterwards, stipulate which data points to collect, manipulate, analyze, and present at each stage so you can create and prioritize a backlog of needed objects and connections.

Build and test in real time

With growing pressure to deliver better quality software at a faster pace, DevOps is fundamental to success. With DevOps, you can release more frequently, with higher quality and stability – from first hours to final days. The whole team can accelerate time to resolution and better manage workflow.

Building out governance

As makers leverage low-code tools, leadership may have concerns about performance, reliability, and security vulnerabilities. Left unchecked, rogue applications can proliferate shadow IT, duplication, and inconsistency. To mitigate risk, create governance with a system of checks and balances. This is where IT becomes your most important partner.

1. Reviews:
While makers are becoming increasingly technical, future rearchitecting or total rebuilds may still be required. By creating a succinct process for guidance and application architecture review, you can future-proof your team against costly rework.

2. Security management:
Whether on premise or in the cloud, IT is eager to protect sensitive information, such as customer and prospect lists, intellectual property, salary information, and core financials. By working with the CoP and using the security tools and protocols embedded within your development platform, you can build mission-critical applications that eclipse spreadsheet-based stop-gaps and protect your data.

3. Audits:
To check quality, adherence to protocols, and maintain best practices, establish a process to capture ongoing technical and user reviews to easily flag and remediate concerns.

4. User permissions:
Administrators can designate what data can be accessed and by whom, through a granular and extensive roles and permissions system. By defining roles, disabling capabilities, limiting personalization and denying permissions, you can reduce the potential for error and better protect your business.

5. Integration and APIs:
While many enterprise software vendors provide APIs, this isn't the case across the continuum of vertical products.
To drive more value and reduce friction, integrating legacy systems can increase efficiency through the renewal and reuse of existing structure and data. Skilled makers can help implement integrations alongside IT admins for business-critical applications, whether on-prem or in the cloud, to reach higher-order business objectives.

Experience the power of Pega

With the right team, tools, processes, and best practices, your Pega Low-Code App Factory will deliver the power of end-to-end automation at enterprise scale. For quick wins, start small and focus on these key elements:

Discipline
With the factory approach, your team can divide applications into outcome-focused cases that can go into production on their own – one journey at a time. Test early, quickly, and with subdivisions for maximum impact.

Quality
Forget minimally viable and adopt an "MLP" mindset: "minimum loveable products." Products should do more than just work. As makers will come to understand when an application has reached a critical threshold, they'll be able to scale their app from "MLP" to enterprise solution.
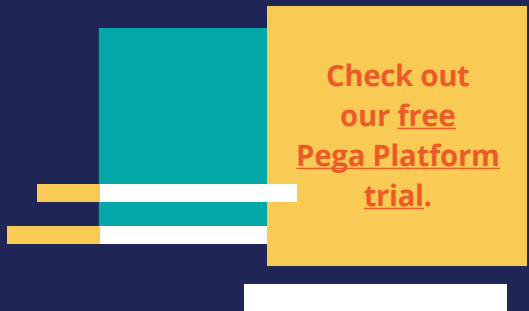
Agility
Continuous integration and continuous delivery (CI/CD) empowers makers to build, iterate, and deploy apps faster than ever – all while collaborating with IT. Coupled with low code, DevOps helps makers go beyond faster coding to rapid app delivery and true business dexterity.

Build for reuse
It's essential to reuse application components from related use cases to allow makers to work quickly with existing assets. Recording previous case types will also help identify additional use cases where new applications can be rapidly deployed. And with an app marketplace, you will save time and resources by providing a catalog of reusable building blocks that will help increase productivity, application quality, scalability, and time to market.

Simplify collaborative app development and start a Low-Code Factory strategy today.

Check out our free Pega Platform trial.

[1] CEOWORLD Magazine. Increase productivity and profitability through continued learning. https://ceoworld.biz/2014/09/03/increase-productivity-profitability-continued-learning/.

PEGA

We are Pegasystems, the leader in software for customer engagement and operational excellence. Our adaptive, cloud-architected software – built on the unified Pega Platform™ – empowers people to rapidly deploy and easily change applications to meet strategic business needs. Over our 35-year history, we've delivered award-winning capabilities in CRM and digital process automation (DPA), powered by advanced artificial intelligence and robotic automation, to help the world's leading brands achieve breakthrough business results.

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