pipefy

The Citizen Developer's Guide to Workflow Optimization

Everything you need to know to improve efficiency, solve common problems, and build better workflows with low-code.

The days when businesses could afford to use IT and developer resources to solve every workflow problem are past.

The present macroenvironment — with its emphasis on agility and operational efficiency — requires a new approach to building and optimizing workflows. Today, low-code tools make it possible for citizen developers to build, automate, and optimize any type of workflow, without any coding experience.

The way things were

In the past, business teams would often wait weeks or months for the IT team to solve even minor issues with their workflows. This system was inefficient and unsustainable, and it had two important consequences.

First, the old model hampered business team agility. Teams weren't able to respond quickly to customer feedback, competitor activity, or market changes. Frustrated by long wait times for automations and optimizations, teams in HR, Finance, Customer Ops, and other departments resorted to shadow IT or other unsanctioned workarounds.

Second, the old model meant that IT teams faced an ever growing backlog that kept them from other critical priorities — such as innovation and security. Combined with the ongoing shortage of IT talent and developers, this model frequently resulted in stressed IT teams, long lead times, and burnout.

Welcome to the era of the citizen developer

Your role as a citizen developer is a game changer for workflow optimization. Using IT-sanctioned low code tools, citizen developers like you build and optimize workflows faster than ever before. This keeps business teams agile and allows IT teams to conserve their resources.
Citizen developers like you are uniquely qualified to optimize workflows. You know the workflows better than anyone, and your insights into how work gets done — and how it should get done — have the potential to drive real change. That means delivering greater operational efficiency, more consistent results, and better user experiences.

All you need are the right tools and a trusted framework.

What will I get from this guide?

No matter your team or department, all the work your team does is organized into workflows. Whether it's HR, Finance, IT, Customer Ops, Facilities — or work that doesn't fit neatly into one of these categories — a deeper understanding of how workflows work will bolster your skills as a citizen developer and help you become a more valuable asset to your team.

This guide is designed to help citizen developers like you identify both problems and opportunities with current workflows, and to create workflows that are structured, efficient, and consistent. Here's what you'll get:

- Review of the basics of workflows
- Checklist - solving the 10 most common workflow problems
- How to identify (and deal with) with long-tail workflows
- Framework for analyzing and optimizing any workflow
- Close-up of how low-code tools help citizen developers optimize any type of workflow

Table of Contents

Workflows 101

Every business depends on efficient and effective workflows. Without them, the business cannot meet its goals or execute its strategy. As a citizen developer, understanding how workflows are organized is essential for solving some of the most common and complex problems for your team.

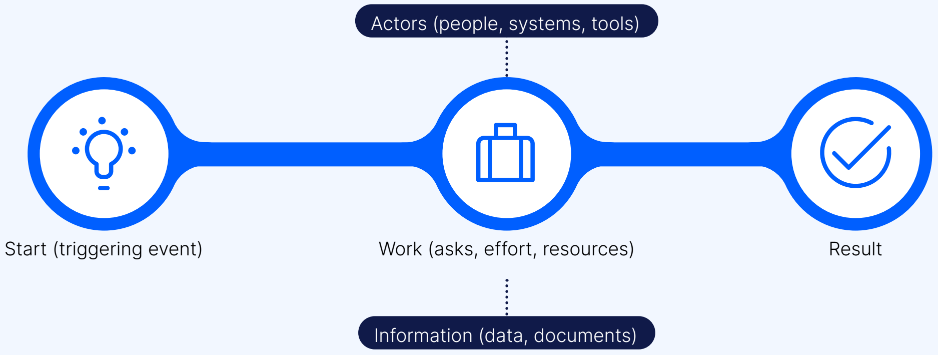
What is a workflow?

Short answer: A method of organizing and completing tasks in order to achieve a specific outcome. Technical answer: A series of tasks with a defined starting point (trigger) and stopping point (result) in which the tasks (or work) require input or effort from multiple actors including people, departments, machines, or systems.

Basic elements of a workflow

Workflows vary in their complexity and scope, but all workflows include the same basic components: a trigger, the work (or tasks), actors, and the outcomes/results.

Anatomy of a Workflow


Actors (people, systems, tools)
Start (triggering event)
Work (asks, effort, resources)
Result
Information (data, documents)

Starting point or trigger
This is the event that begins the workflow. Triggers can be an action, a decision, a specific or scheduled time, an email, or any other signal that the workflow needs to be initiated.

Work
Tasks, steps, or activities that create the result. This could be a physical or mental activity done by humans, or the function of a system or machine.

Actors
Actors can be people, systems, or machines who handle the work item in any way including influencing the work, moving the work along, or delaying the work. In their book Workflow Modeling: Tools for Process Improvement and Application Development, workflow experts Alec Sharp and Patrick McDermott define "actors" as anyone or anything who "holds the work." In other words, any person or system that influences the work or moves it through the flow.

End result
The product, service, or information that the workflow produces for the customer. Keep in mind that the customer may be internal or external. Sharp and McDermott define the customer as "the recipient or beneficiary of the result produced by the business process."

10 Common workflow problems (and how to fix them)

Teams that are struggling with unstructured and inefficient workflows rarely have the bandwidth or expertise to solve these problems themselves. That's where a citizen developer's skills are most valuable. In these scenarios, it's the citizen developer who takes the lead in diagnosing and solving the broken workflow.

Regardless of the team or department, the most prevalent workflow problems can impact any type of workflow. Here are 10 of the most common, their symptoms, and how to solve them.

Problem	What it looks like	How to fix it
Workflow is inefficient and requires too much manual work.	• Recurring tasks and activities have to be managed manually. • Team members have little time for value-adding activities like customer service or problem-solving. • Bottlenecks are frequent.	• Use a low-code tool to automate any task or activity that is frequent or repetitive. Examples include emails, approvals, routing, status updates, and data entry.
There's no standardization or structure to the workflow.	• Each team member or department completes the workflow in a unique way. The workflow — and its results — are inconsistent. • Difficult to enforce security and compliance measures. • Visibility is limited.	• Apply forms, rules, and conditionals to structure workflows and eliminate variation. • Map all workflow phases to improve consistency and predictability.
Manager has limited visibility and control of the workflow. External stakeholders have limited visibility into status.	• Difficulty determining who has ownership of items in the workflow. • Reports require too much manual work and over rely on spreadsheets and email threads.	• Apply labels that assign and route items to specific users. • Use a workflow tool that tracks each item in the workflow and gives team members multiple views of their items. • Shared inboxes, permission management, and automated routing can ensure visibility and that incoming tickets and requests reach the right stakeholder. • Give external stakeholders the power to check on statuses themselves using a link, or automate notifications.

Problem	What it looks like	How to fix it
Items in the workflow are often delayed or finished after the due date.	• Requests and tickets often blow past SLAs or deadlines without anyone's awareness. • Processing times take too long.	• Create automatic notifications to alert managers and other users anytime an item approaches or passes a deadline. • Use a custom view to identify all at-risk items before they become late.
Team is overwhelmed by a high volume of requests, often from multiple channels.	• Teams accept new requests, tickets, or other information through a variety of formal and informal inputs. This system has become unmanageable and items sometimes fall through the cracks.	• Consolidate incoming requests into a single stream, whether they come from email, an app, or even an informal request. • Use a shared inbox, labels, notifications, and routing rules to distribute incoming items to team members.
Teams spend too much time switching between apps or systems.	• Teams need to update the same data in more than one place, or they spend too much time copying and pasting data from one app to another. • Data must be moved between multiple spreadsheets.	• Use integrations to consolidate data, orchestrate workflows, and centralize information onto a single screen.
Tools used to manage the workflow can't keep up with pace of change.	• Current system requires complex and expensive customization to stay relevant. The IT team doesn't have the resources for continual updates.	• Manage workflows with low-code software that makes it easy for business teams to modify existing workflows or create new ones without any coding experience. • Low-code software helps conserve IT resources.
High error rate due to manual data entry.	• Teams have to enter and re-enter the same data in multiple apps and systems.	• Use integrations to connect the workflow software with the components of the existing tech stack, so that data moves seamlessly between apps and systems.
Teams waste time tracking down missing or incomplete information.	• Requests or tickets arrive via email, apps, on paper, or even through verbal requests. Items frequently lack all the information needed to get them underway.	• Use a standardized form with rules and conditionals to ensure that all relevant information is entered before the ticket or request can be submitted.
Delays due to broken handoffs or failure to follow up.	• As items move through the workflow from one person to another, they fall off the radar. Team members are unaware items require their attention.	• Automate notifications via email or messaging apps (like Slack) to alert team members when an item in the workflow is waiting on them. Use labels to assign ownership.

Long-tail workflows

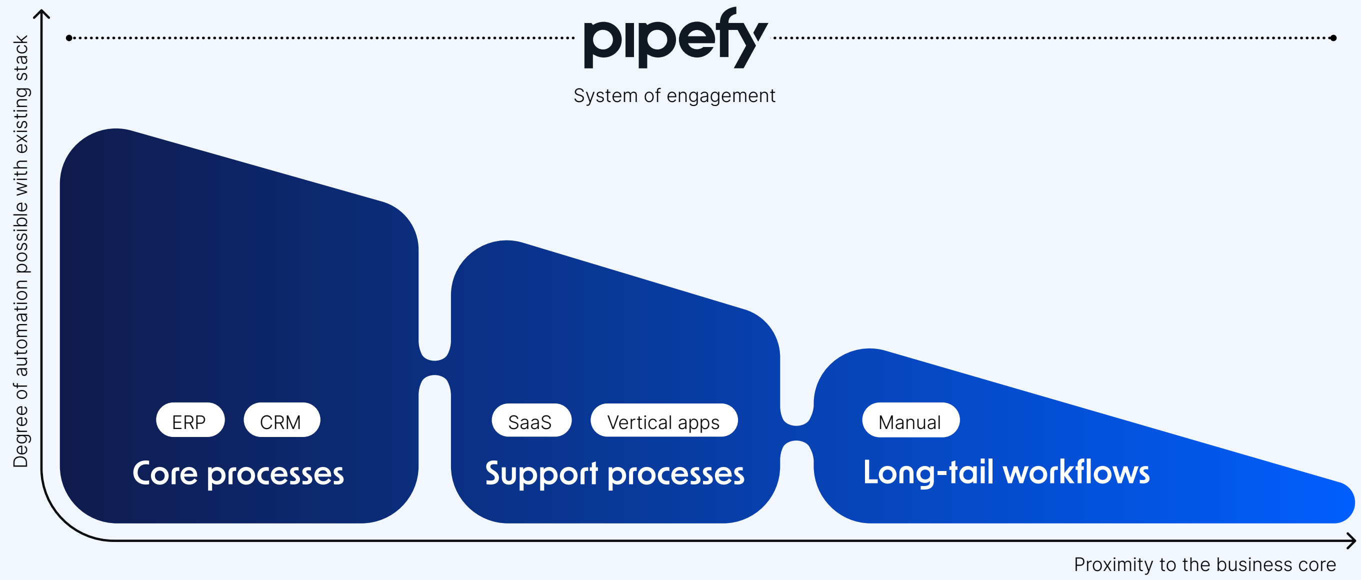
At some point, every citizen developer encounters a unique or cross-team workflow. These types of workflows aren't easily managed with existing apps or systems, and they typically require a lot of back-and-forth between people and apps. We call these long-tail workflows.

What is a long-tail workflow?

Long-tail workflows are the unique, ad hoc, or custom workflows that emerge in response to evolving business needs and increasing stack complexity. Long-tail workflows can be heavily dependent on manual work. If your team has cobbled together a makeshift solution to bridge a gap in the existing tech stack, you are probably looking at a long-tail workflow.

Long-tail workflows often lack structure or standardization, which means that their inputs, outputs, and steps may not be clearly defined at first. In some cases, different departments, teams, or locations may each have their own version of the workflow. This inconsistency makes it difficult to monitor and manage these workflows. It also makes it harder for IT to enforce security and compliance requirements. You can usually identify a long-tail process by the presence of some (or all) of the following characteristics:

- Addresses a gap between processes, systems, or apps
- Features a human-in-the-loop model
- Requires a high degree of manual work
- Crosses team or department boundaries
- Heavily dependent on spreadsheets, email, and collaboration tools
- Falls low on the list of IT priorities
- Currently managed with workaround or shadow IT



Create value for customers, generate revenue

- Structured and complex
- Expensive to own and mantain

Enable fullfillment of core processes

- Contained within departments or teams
- Higher degree of manual input

Manual or unstructured, fill process gaps between systems, workflows, and apps

- Lack of standardization and consistency
- Not prioritized by IT
- Hamper business agility

Long-tail workflows occur most often further away from the business' core processes.

Optimizing long-tail workflows

Long-tail workflows occur in the capability gaps of the existing apps and systems. For example, when the finance team's software doesn't accommodate a procure-to-pay workflow, or when a team handles a high volume of service requests that arrive through multiple channels.

To solve long-tail workflows, most citizen developers will use a low-code tool. Low-code features such as automations, rules and conditionals, and integrations make it possible to structure and standardize the flow, no matter how complex or unique.

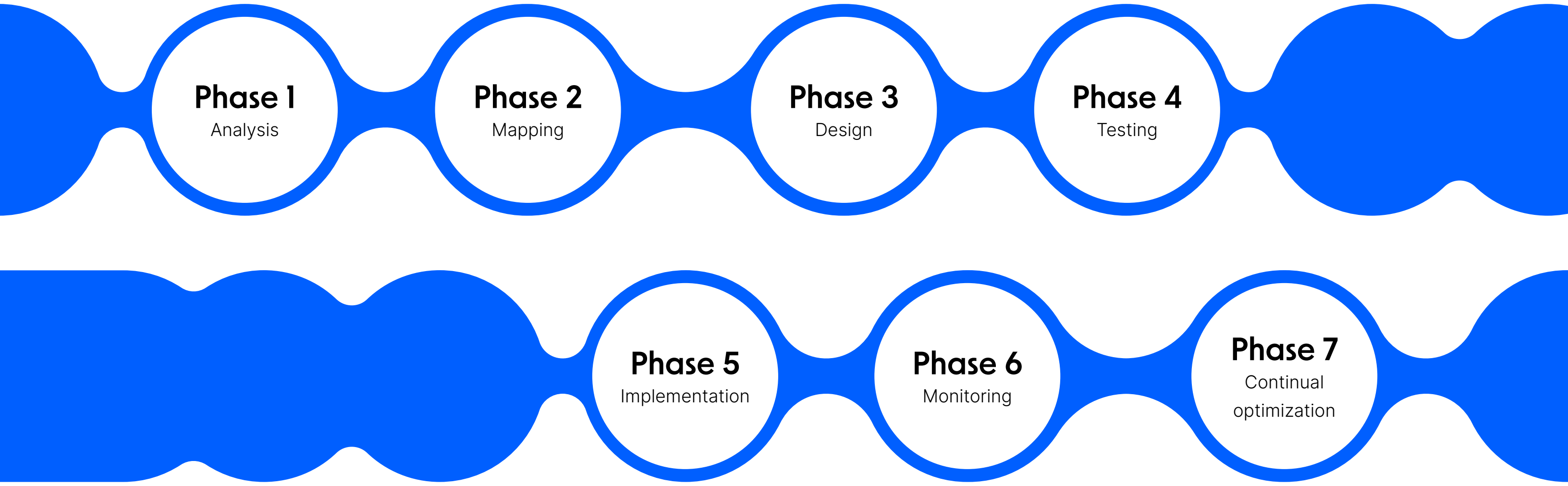
The good news is that even long-tail workflows can be optimized with the right tools and framework. The better news is that the framework and tools are the same for any type of workflow.

So what does this framework look like?

7 phases of workflow optimization

Workflows are the connective tissue of any organization. Their quality and consistency have a direct impact on operational efficiency, productivity, and the bottom line. At some point, every workflow will need to be optimized.

That's because businesses evolve, strategies change, and new technologies are introduced into the stack. Workflows also have to scale in order to keep pace with growing businesses. As a citizen developer, you'll need to be prepared to assess and improve all of the workflows that impact your team. To do that, you will want to follow an established framework. Here's what that looks like:



Phase 1 Analysis

Phase 2 Mapping

Phase 3 Design

Phase 4 Testing

Phase 5 Implementation

Phase 6 Monitoring

Phase 7 Continual optimization

Phase 1: Analysis

The first step in optimizing any workflow is assessing the quality of the current or "as-is" workflow. This step is crucial, and the more details you gather, the deeper your understanding of the workflow will be.

Workflow analysis helps you identify the problems with the current workflow like bottlenecks, collaboration or data silos, flawed handoffs, and common sources of error. The analysis will also bring inefficiencies to light. To get you started, here is a checklist for conducting your own workflow analysis:

Workflow analysis checklist

☑ What is the status of the current workflow?

Your assessment should include the following aspects of the workflow:

- ✓ Efficiency
- ✓ Speed
- ✓ Visibility
- ✓ Ownership
- ✓ Consistency
- ✓ Handoffs
- ✓ Frequency of errors
- ✓ Amount of manual work
- ✓ Quality of user or customer experiences
- ✓ Amount of repetitive data entry
- ✓ Frequency of app switching

☑ What is the workflow trigger?

Every workflow begins with a starting point or trigger. For example, a customer service workflow begins when a request is received. In an employee onboarding workflow, the trigger may be the receipt of a signed employment agreement.

Other workflows may kick off when someone sends an email or a message. Some workflows happen on a regular schedule, such as the first of the month or each week on a certain day.

☑ What is the workflow endpoint?

The efficiency and success of workflows are measured by the results they produce. The result is the endpoint — the final outcome of all the combined activity in the workflow.

Results should be discrete and countable. In other words, you should be able to distinguish one outcome from another.

☑ List everything that happens between the starting point (trigger) and the endpoint (result)

This is where the real work of workflow analysis happens. In this step, you will need to trace — in detail — every task, activity, and handoff that occurs in the workflow.

Also be sure to note every system, app, and information source that is involved in the workflow.

Finally and most importantly, you want to make sure you know every person or system that handles the work, even if they don't take any action. For example, if incoming items in the flow are sent to a shared inbox, make sure you note that in your list.

> **Pro tip:** workflows often contain hidden work. These tasks and activities might not be visible to an outside observer, so it's important to talk to everyone involved in the workflow to make sure you have the most detailed information possible.

Phase 2: Workflow mapping

At this point, you will want to organize all of the information you've gathered into an illustration or visual. This is known as a workflow map or workflow diagram. It helps all stakeholders understand how the work is getting done. You can create your workflow diagram in many ways. A few common tools for mapping workflows include:

- Pen and paper
- Drawing app on a tablet
- Visual collaboration tool like Miro
- Whiteboards

Whichever method you choose, remember that the workflow diagram you create will be a reference point for the rest of your optimization efforts. These diagrams are useful for documenting workflows, training new employees, and helping external stakeholders (such as your IT department) understand what happens between points A and B.

Common elements of a workflow diagram

The point of creating a visual representation of a workflow or process map is to help us identify the who, what, when, and where of our work. Here are the most common components of a typical workflow:

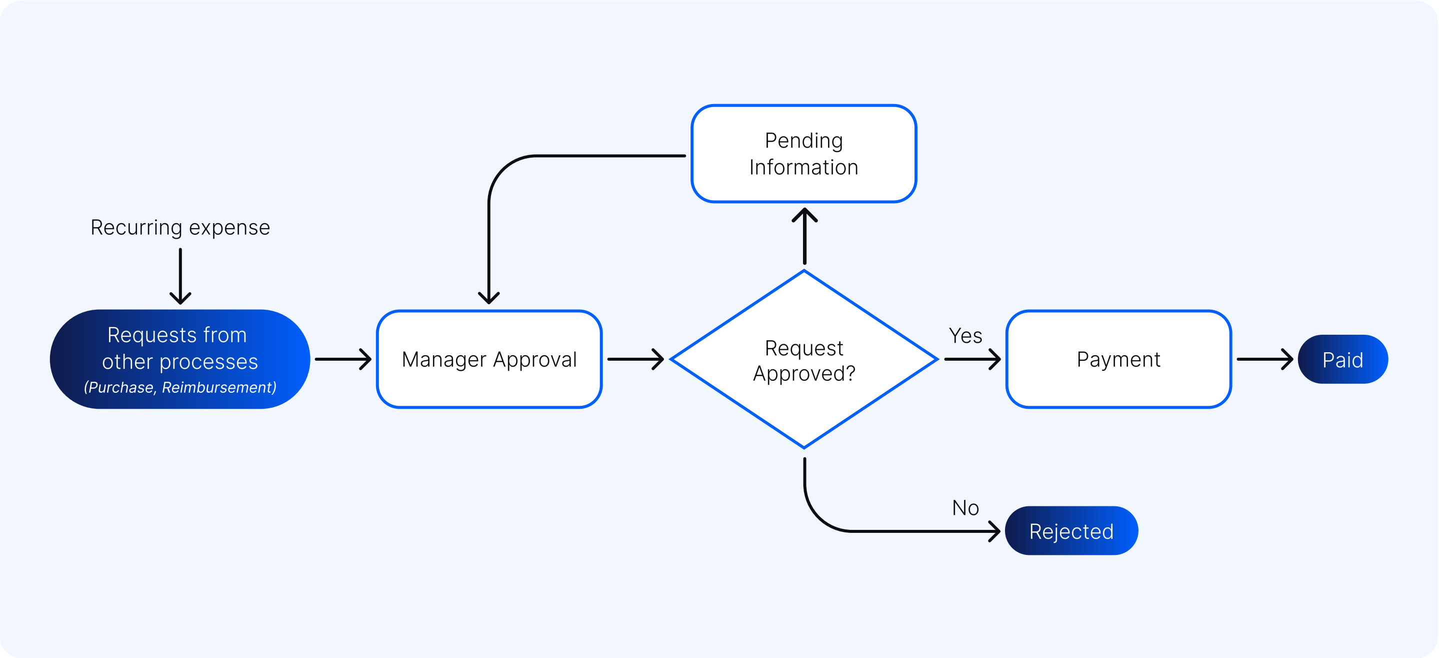
Triggering event	Starting point
Result	End point
Actors	Any person or system who holds the work
Sequence	The direction of the flow
Handoffs	Where the work moves between actors
Time	Do events have to happen within a certain timeframe?
Delays	Points at which work is slowed or comes to a halt
Task details	If necessary (and space allows) additional information about the work may be included

One last point about workflow maps and diagrams: you may want to use standard flowchart symbols to identify these different elements of your workflow. While not always necessary, using standard symbols can make it easier for yourself and others to understand the diagram.

Learn more about the standard flowchart symbols.

Example of a workflow diagram

Below is an example of an accounts payable workflow. Here, we've used the standard symbols for the starting and ending points, each activity, and the point at which decision must be made.



Accounts payable workflow.

Phase 3: Design

Now that you've analyzed the current or "as-is" workflow and identified the areas you want to improve, it's time to build the optimized or "to-be" version. This new diagram will serve as a guide to anyone involved in the workflow, or those who need to understand how the work is organized. Workflow diagrams may be part of your company's documentation, and they can help others understand how activities and information are pieced together to create a final result.

You can use the same process for this phase as you did in the previous, relying on the same tools and symbols.

> **Pro tip:** Request management workflows often depend on requests from multiple channels. For example, requests may be received through email, messaging apps, online forms, or even informally through verbal requests. These requests often arrive with missing or incomplete information. To solve, use a low-code tool with a rules feature to prevent missing information, and consolidate all incoming requests into a single stream.

Phase 4: Testing

Once you've designed the optimized version of your workflow, it's time to test it. Invite other stakeholders into the testing process to make sure that it works for them, and to gauge its impact on their user experiences.

Ask the following questions:

- Does the workflow account for all actions and activities?
- Are handoffs and ownership clear from start to finish?
- Are there any instances of duplicate or unnecessary manual work?
- Are there any data or collaboration silos present?
- Does the new workflow make their work easier?

Once you've gathered feedback from stakeholders and tested the workflow results, make necessary revisions before moving onto the next phase.

Phase 5: Implementation

By this point, you've done the work of analyzing your existing workflow, and building and testing the new version. Congratulations! Now you're ready to implement your workflow, and you can do so with confidence.

The key to successful implementation is communication. Follow these three steps to ensure a successful launch for your new workflow.

- Training: make sure everyone who will be impacted by the new workflow understands what has changed.
- Awareness: give everyone (including IT) access to any relevant documentation, including the workflow diagram you created.
- Timeline: communicate the official start date and send reminders when the day arrives.

Phase 6: Monitoring

Once your new workflow is in motion, monitor the results. Look for consistency, quality, and errors. Your workflow software should make it easy to monitor key metrics and KPIs, to measure your success and identify potential points of improvement.

It's also important to understand user experiences in your new workflow, from all stakeholders. Plan on collecting and analyzing feedback on a regular basis, using a survey tool that allows you to organize your feedback.

Phase 7: Continual optimization

As business needs and strategies change, workflows will need to evolve. Remember to revisit your workflow on a regular basis to ensure that it's still aligned with business goals and producing the desired results. Anytime a new tool is added to your tech stack, review its impact on your workflow. New apps and systems can also be opportunities for further optimization.

Essential elements of workflow optimization

There are a lot of moving parts to workflow optimization. To simplify, we've put together this visualization to make it easier for citizen developers to approach workflow optimization with confidence.

Automations	Features	Capabilities	Integrations	Security	Benefits	Solves
			1 ERP Enterprise software			
1 Rt Repetitive tasks	1 Lc Low-code	1 C Centralization	2 L Legacy systems			1 Ss Spreadsheet sprawl
2 E Email	2 T Templates	2 Wv Workflow visualization	3 D Databases	1 SSO Single Sign-on		2 Ee Endless email
3 N Notifications	3 Cf Customizable forms	3 Do Documents	4 Ma Messaging apps	2 2FA Two-Factor Authentication	1 We Workflow efficiency	3 Ds Data silos
4 A Approvals	4 R Rules & conditionals	4 M Metrics & KPIs	5 Ct Collaboration Tools	3 Pm Permission management	2 Ae Avoids errors	4 B Broken handoffs
5 S Status updates	5 Gp Guest portal	5 S Standardization	6 Ds Document signing	4 Up 99.9% or higher	3 Mc More control	5 M Manual work

Low-code workflow optimization software

Citizen developers need the right tools to optimize their workflows. Low-code options are often the tool of choice, because they are designed to be used by those without coding experience, and because they help ease the pressure on the IT backlog.

Look for workflow optimization tools that include the following:

Automation

One of the most important problems citizen developers can solve is time wasted on repetitive or manual tasks.

By automating these tasks, citizen developers help their teams **increase speed and efficiency, avoid errors, and produce more consistent results**.

Features

Citizen developers need the right tools to work. For optimizing workflows, this means using a low-code platform with features that make it easy to build and modify any type of workflow. **Low-code also helps your business conserve IT resources**.

Capabilities

Workflow software should support each phase of the workflow optimization cycle, **empowering citizen developers to visualize, standardize, and centralize their activities**. Look for low-code solutions that make it easy to visualize workflows and monitor KPIs.

Integrations

Most workflows depend on a wide range of components in the existing tech stack. This includes apps, systems, databases, and legacy software. **Choose tools that integrate with — and extend — the components of the existing stack**.

Security

Any no/low-code software citizen developers use will need to meet IT team requirements. **Look for tools that simplify security, governance, and compliance**, and which meet or exceed industry standards. These include SSO, 2FA, Permission Management, and compliance requirements such as GDPR, LGPD, ISO 27001, and SOC2.

Benefits

The primary goals of any workflow optimization effort are to increase efficiency and speed, reduce errors, and achieve deeper levels of control and visibility. **Low-code workflow tools speed up time to value, conserve IT resources, and keep business teams agile**.

Solutions

Some common problems citizen developers encounter in workflows include:

- Process gaps
- Broken handoffs
- Lack of centralization
- Frequently missed deadlines
- Over reliance on spreadsheets

Our vision

Citizen developers are the future

According to Gartner, by 2025 70% of new applications will be developed using a low-code solution. That means businesses will be increasingly reliant on citizen developers to help bridge the gap between business team needs and IT resources. Citizen developers understand the problems and opportunities better than anyone, and low-code software will empower them to become agents of change in their organizations.

Operational efficiency is the watchword in this economy

After years of unprecedented growth, businesses are shifting their focus toward achieving operational efficiency across the enterprise. That means every department, every team, and in every process and workflow. Citizen developers will become the vanguard for operational efficiency and workflow excellence.

People are the most important element in every equation

We believe that people are always priority number one. We make this ideal a reality by solving the problems that people find in their work. That means building and improving workflows in order to make work and life better for everyone.

Every possibility begins with people

The future of work is about seeing people for who they are, then creating the tools and workflows to help them thrive. We invite you to see what's possible with Pipefy.

Works cited

Sharp, Alec. *Workflow Modeling: Tools for Process Improvement and Application Development*. 2nd Ed. Artech House: Boston. 2009. pp 40 - 57, 246.

"Critical Capabilities for Enterprise Low-Code Application Platforms." © Gartner, September 2021.

pipefy