HCLSoftware

Strategic Citizen Development at Scale

Navigating Governance from Idea to Implementation

HCL Volt MX

Table of Contents

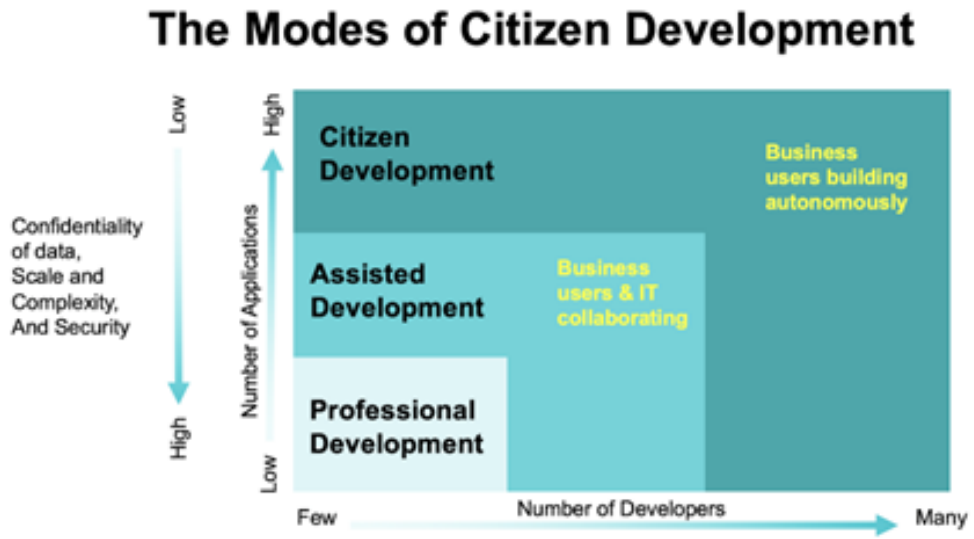
Introduction

Citizen development promises faster delivery with less skilled developers. At scale, it can foster major gains with innovation, increase efficiency and reduce IT backlog.

Is citizen development right for everyone in your organization? The answer depends on the profile of your workforce. If you're in a knowledge industry, opening it to all employees will maximize your innovation pool. Organizations with deskless workers might want to limit it to a select audience.

Whoever the audience is, the citizen development platform needs to be open so users can build apps autonomously with minimal friction. This level of democratization is key to unlocking innovation at scale.

At the same time, there should be limitations on what citizen developers can build – and who's doing the building. Think of citizen development as a self-service zone which allows users to safely create apps without IT involvement. The apps created will typically be internal facing, low on the complexity scale and limited in the amount and type of data they process. Some of these limitations need to be set in the platform, while others will be enforced by policy and inspection.



Citizen developers looking to add complexity or scale to their apps will need to enter the assisted development zone to work with IT professionals. Here it's important to have a platform which can cater to both types of developers and easily transition an app from citizen to professional development.

With the right platform, investment and governance, all these new applications and innovation can be gained without security vulnerabilities, data leakages, uncontrolled expansion or exploding technical debt.

Continued...

Introduction *(Cont.)*

This paper details the essentials to ensuring a successful citizen development program. We'll outline the two key topics which are part of this:

- Maximizing efficiency, ROI and quality of output
- Mitigating against threats, risks or other security and compliance issues that you might encounter

Maximizing Efficiency, ROI and Quality of Output

Select the right citizen development platform.

Not every citizen development platform is designed for scale. Some require IT involvement and extensive training for users to create apps. This limits how many citizen developers can start building. Here are some of the things you need for a platform to be able to scale:

Easy on ramp

Select a platform with an easy enough on ramp for everyone to be successful. Assuming a typical citizen developer has a technical proficiency level of working with spreadsheets, you'll want to pick a tool that most of your users will be able to use without extensive training.

Seeing that most use cases will be to replace paper and spreadsheet-based processes, you'll want the ability to easily transition from spreadsheets and sketch out replacements to paper or PDF-based forms. Simple visual workflow design is also a must.

No IT decisions needed

Avoid platforms that put users in the position where they have to make IT decisions. This includes answering questions like: where and how is the data stored? How do entry widgets on a form bind to fields in the database? How will my app users authenticate? What is the dev/ops pipeline for moving my app to the production server? These are not the domain of the citizen developer. The platform you choose should take care of these questions automatically.

> Select a citizen development platform that scales effortlessly without IT hurdles.

Select a platform where the database, forms, workflow, business logic and access settings are tied together. This will make app development easier and help avoid security issues. Many low code tools split the app creation process into separate pieces – the data model, security, forms and workflows. Working with independent pieces introduces more vectors for risk and adds complexity to the app creation process. While a high degree of granularity is great for seasoned professionals, it introduces a steep learning curve for the citizen developers and increases the chance they won't get the security settings right.

Continued...

Maximizing Efficiency, ROI and Quality of Output *(Cont.)*

Unified platform for citizen and professional developers

Another key consideration is to select a platform that allows apps created by citizen developers to be extended and enhanced by professional developers. Apps developed by a citizen developer may need to be hardened and turned into an enterprise app, or citizen developers may engage in a larger effort led by IT. If they're involved with building out a successful prototype, it's turned over to professional IT developers for completion. Having this ability improves the final solution and saves time and effort by not having to re-write the app.

Make the Platform Yours

Rather than using the citizen development platform out of the box, you should enhance it. The more you enrich the citizen developer tool set, the more benefits your organization will gain, including:

- Faster time to value by enabling users to compose apps from your building blocks.
- Standardization and consistency of look, behavior and security profile.
- Avoid building everything from scratch.

You can think of the enhanced platform like a salad bar. The standard salad bar comes with base items, but users get more options – and ultimately, a better salad – when they add their own touches. The same thought applies to your citizen development platform. By adding curated access to data, templates, custom themes and widgets, users will be able to compose higher value apps, get more options and better comply with company requirements, look, behavior and security profile.

Curated access to external data

Citizen developers will need access to information from some of your organization's systems to be productive. In most cases, the access needs to be read-only and limited to a few sources.

Information about people and the reporting structure is vital for building workflow apps. For example, if you're building an approval app, you need to know the identities of the requester and their manager for approval. This information typically comes from the organizational directory or some other HW system.

Data on customers and products are often needed to populate dropdown selections in apps within sales and service. The data could be hard coded, but then you run the risk of an outdated and inaccurate list. Most organizations manage this information in their CRM system.

Performance data is often needed when creating dashboards and building apps for tracking and planning. This data typically comes from the EPR system and is often needed by those in finance, sales and operations.

Enabling citizen developers to consume external data can make a world of difference in the value of the apps they create. But you can't just point citizen developers to different REST APIs to incorporate into their apps. They may not understand REST endpoints, GETS, POSTS, request headers or the security models around those APIs.

Instead, the access to the data must be curated by someone in IT. This means that IT selects the data elements to be exposed (based on citizen developer

Continued...

Make the Platform Yours *(Cont.)*

need) and authorizes who can use them. You may want to limit access to some information and make other information available to all. It's also important to consider how the data is exposed to the citizen developer. It needs to be simple for someone building a workflow app to look up a person's manager or populate a dropdown with customer names. Otherwise, they won't use or benefit from the external data.

Customize the palette with your components

All citizen developer tools include 'out of the box' widgets that users drag and drop onto a canvas to create apps. But there are limitations to consider, like widget capabilities and compliance with your organization's specific design system.

Being able to add your own widgets to the palette lets you address these issues and mold the citizen developer tool into one where users compose apps with your components. If you have a design system, these components already exist. The citizen developer tool should provide you with the capability to add your components to the product's palette, so citizen developers can easily achieve the look, behavior and functional characteristics important to your organization.

-Add custom templates

Help citizen developers get off to a quick start, avoid duplication and provide a level of standardization by providing custom templates. Vendor supplied templates are great but often limited. The goal is to have pre-built pieces as starting points which are on target with the needs of your users.

Custom templates let you cater to your citizen developer's unique needs. For example, the marketing team may be interested in building apps for customer offers and lead generation. Templates for this audience can be set so that they're pre-wired to your organization's marketing systems. Sales may need templates which are set up with access to the CRM system for account planning. Users building workflow apps need templates that easily assign tasks to people from the company directory. Putting the right templates in place that users can review and instantiate as the start of their app will pay big dividends.

-Add custom themes

Putting custom themes in place will save citizen developers the time and effort involved with setting up the look of their apps. Just like creating a presentation, when you start with your organization's theme you can be assured you're using the right color palette, icons, logos and more. If everyone

Continued...

Make the Platform Yours *(Cont.)*

uses the themes, they'll consistently adhere to the organizational branding.

Create a Center of Excellence

The Center of Excellence acts as the command center for apps. This is the hub where citizen developers can access assets that maximize low code capabilities, ask questions, share ideas, share applications and more. It's also a place where IT can provide information on the platform and guidelines for using it.

-Set up a knowledge base

Initially setup by IT, the knowledge base contains information on the platform, guidelines for using it, tips and examples. This can include use cases for apps, UI standards, integrations and security requirements. Once set up, community contribution is encouraged so it evolves with the maturity and needs of the citizen developer community.

-Set up a forum

The forum is the primary place for citizen developers to go to ask questions. IT monitors the forum to ensure questions get answered, and users can answer each other's questions as the community gains experience and skill with the citizen developer platform. IT takes the lead on issues that cannot be answered by escalating them to the vendor forum and support channels.

-Provide options for training

This can be as simple as pointing to the vendors publicly available training or something developed in-house.

-Set up an internal user group

As usage of the platform picks up, more and more citizen developers will want in-person or online live opportunities to meet and exchange ideas. Meetings should happen on a published cadence.

Launch and promote it

You'll want to promote the platform like you would any other internal service targeted at all employees. It's important to raise awareness and let employees know that the citizen development platform is sanctioned by senior management and the IT organization. Communicate internally with emails and blogs. The goal is to scale the service so more users benefit from it.

Mitigating Risks

Create acceptable use policies and terms of service

Think of this as the contract the IT sponsor of the platform has with the citizen developer. It needs to outline what types of apps and data are prohibited and describe usage scenarios which aren't suited for the platform due to poor application performance (e.g., an app which processes millions of records or has highly confidential data). It should also set proper expectations for the platform's availability, performance, backup and recovery.

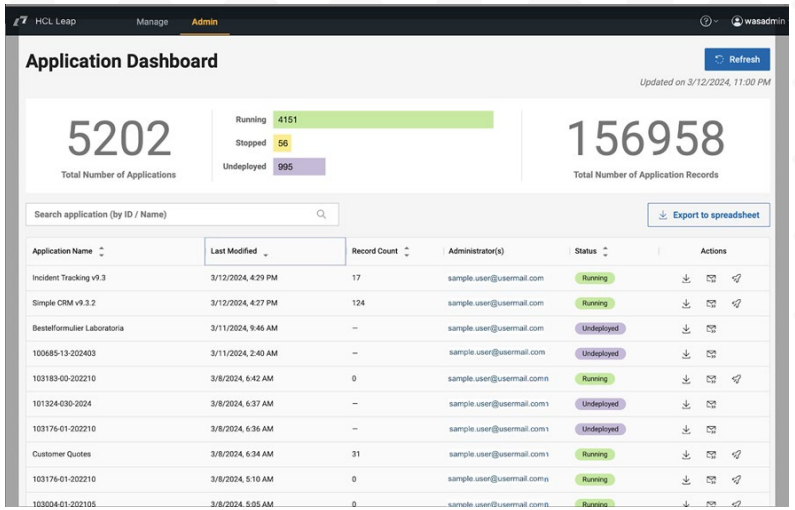
Whoever owns the citizen developer app needs to agree to acceptable use policies and terms of service. The app owner needs to be held responsible for everything specific to the app including its support and updates. They're responsible for determining the correct security settings for the app, the data collected by the app and the periodic backup of the app and its data. IT on the other hand is responsible for the platform. They ensure platform availability, enhancements, backup and recovery.

Manage application lifecycle

Applications created by citizen developers could have a lifespan of a few days to many years. You want to encourage all employees to develop and learn. Some users may come in and create a bunch of apps as part of a learning exercise, or for fun, just to get acquainted with the platform. Others will build apps that stay in use for months or years.

Keep track of what users are deploying

It's vital that IT keep track of the apps that have been created. This is more than simply acquiring a list of the different apps on the platform and who created them. IT needs to know the type of app, anticipated level of usage, expected life of the app, the owner (which may differ from who initially created it) and who's responsible for supporting the app.



To help with tracking apps the platform should have an admin dashboard and API for gathering the needed information. Above you see an example of one which is in the citizen development environment of Volt MX.

Continued...

Mitigating Risks (Cont.)

Put an attestation process in place

A process needs to be put in place for IT to check-in with the app owner on the items above. Putting a gate in front of them before they build the app creates friction and could lower adoption, but immediately after deployment may also be too early as well. From our experience, the best time is a couple of weeks after the app has been deployed.

New apps

This process should be automated and triggered by the app's deploy event plus a 2-week interval. A script should send an email notification to the app owner with a link to an attestation form. The form needs to collect information from the app owner and also serve as a contract with the app owner so that they agree to the terms and abide by company policy. If the form isn't returned, IT needs to remove the app from the platform. This ensures that IT has a record of all apps in production and an agreement with the app owner.

Existing apps

After 6 months, the app owner should get a follow up notification to re-register the app. In this case, the form needs to capture:

- Do you still need the app? If so, how long?
- Has ownership changed?
- Have support contacts for the app changed?

If the re-registration form isn't returned – or if the app is no longer needed – then IT should remove it from the system.

Audit application security settings

Periodic inspections are necessary to check the security setting in apps. The goal is to see if access has been set wider than what it should be. For example, did they set it to "all authenticated users" when it should be set to a specific subgroup? Although app owners agree in the attestation process to properly set access permissions, periodic audits will help IT gain confidence that they are doing so and that risk of data leakage is mitigated.

To help facilitate audits, admins should have the privilege to open and inspect security settings for any app and an easy to select apps for inspection. The audit can be done from the Admin dashboard where apps are chosen and inspected.

Intranet vs Extranet

An important decision when setting up the platform is whether the platform should be exposed to the

Continued...

Mitigating Risks (Cont.)

internet. Setting it up within your firewalls will limit apps to internal facing but creates a layer of safety. Exposing it outside the firewall opens it up to external anonymous users but exposes the platform to new threats. In some cases, you may want external anonymous access for customer facing apps. These tend to be the exception and often warrant their own deployment which is separate from the one open to all employees.

Setting user controls and guard rails

There are many settings that should be available on the platform to help guard against malicious activities and ensure performance. Some of these include:

- Limiting the types of files that can be added as attachments: By creating a whitelist, you can only allow what is deemed safe.
- Limiting the maximum size of file attachments: Excessive file sizes overtime may cause poor performance and bloat storage capacity.
- Setting the maximum size of queries and downloads: Apps that filter millions of records can put a drain on the database.
- Determining which domains are allowed for images, videos and services: internal or outside the organization.
- Using Strict CSP to guard against script injections: This is recommended in all external facing scenarios.
- Allowing anonymous access: Restricting app usage to those in your organization's directory provides a layer of security.
- Restricting whether the apps created can be embedded into something else.
- Setting safe limits on the level of JavaScript used when building apps: Advanced users may have skills which could create security issues, so it's best to give them a safe level to use.
- Authorizing which users can build apps with specific integration services, as well as which app users can run apps with specific integration services: You may need to put some safeguards on some of the external data which is being introduced to your citizen developers.

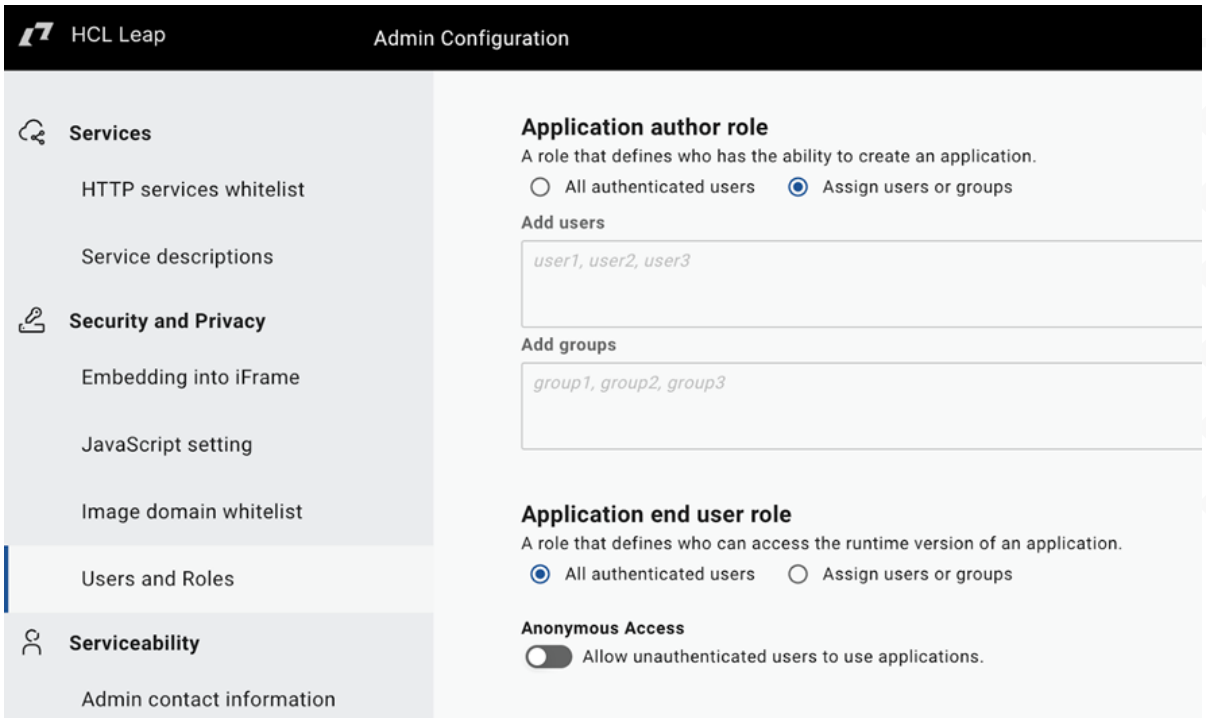
Set expectations for the platform's availability, performance, backup and recovery

Continued...

Mitigating Risks (Cont.)

The citizen developer platform should have a capability which allows the admin to configure these various controls. To the right, you'll see an example of one which is in Volt MX.

The admin section of the product should have its own security context (e.g. not exposed to the internet) and be limited to a few people in IT who are responsible for the platform.



Summary

Citizen developers need their zone where they can experiment, learn and build apps to increase efficiency, quality and innovate. Set up properly, they'll flourish and allow the business to do more without incurring additional risk. They also need a path to collaborate with IT on projects.

IT professionals need to take the lead by setting up citizen developers with the right platform, extensions and governance programs. Senior management needs to back the effort and ensure that all employees are aware of the service. If you're not embracing citizen development, or doing it properly, you risk losing out to shadow IT.

IT should also plan for the projects where they can engage citizen developers for contribution to the overall effort. Not everything needs to be done solely by a professional developer. Final assembly from IT supplied component parts of certain pieces of an effort is a way to gain speed and agility.

HCL Volt MX empowers both professional and citizen developers to build impactful applications across various digital channels. This industry-leading, low-code platform fosters innovation by simplifying development tasks with AI-powered assistance and drag-and-drop functionality. Even business users with minimal coding experience can contribute to process automation and workflow digitization - all within a secure, IT-governed environment.

Unleash the potential within your organization and embark on your Volt MX journey today.

HCL Volt MX empowers both professional and citizen developers to build impactful applications across various digital channels.

Try Volt MX for Free

HCLSoftware

hcltechsw.com