NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 3 Issue 5, Pg. No. 14-24, AIJCST-V3I5P102, 2021
https://doi.org/10.63282/3117-5481/AIJCST-V3I5P102

Original Article

How Citizen Developers Changed the Game

***Adityamallikarjunkumar Parakala[1], Aaron Bell[2]**

[1]*Lead Rpa Developer at Department of Economic Security, USA.*

[2]*Sr. Specialist SAP OTC,Microvention Inc, USA.*

Abstract:

Artificial Intelligence (AI) has come in with highly intelligent systems that continuously do ever more complex tasks. The proposed research relates to one of the newer paradigms in AI research, Agentic AI, which can be understood as autonomous, self-directed software agents that can execute goal-driven behavior via multimodal reasoning. This paper explores the design, construction, and deployment of Agentic Artificial Intelligence systems capable of synthesizing information across different modalities, including text, images, audio, and environment monitoring sensors, so as to generate intelligent autonomous choices. The main deliverable of the research is the development of a framework, which combines multimodal mechanisms of reasoning with agent-based architectures, and allows adaptive and context-sensitive behavior. To address this problem, we postulate a modular architecture that integrates the ability to learn fast and enough through reinforcement learning and profound associations throughout symbolic reasoning in this paper to effectuate decision-making in a real-time scenario and learning in a challenging arena. Our literature review is extensive and follows the development of autonomy in AI systems, the purpose of multimodal reasoning and issues in integration. The approach we use presents a layered model, which consists of perception, cognition, and action modules that accomplish specific tasks and communicate with each other using a common knowledge base. Our prototype system has been tested on various benchmarking scenarios, including navigation, task planning, and multi-agent coordination. Experience indicates a significant increase in task completion rate, awareness of context, and learning efficiency compared to unimodal and static AI agents. The paper concludes with a discussion of the ethical implications, limitations, and future trends of developing generalizable, safe, and socially agreeable autonomous agents. The study aims to develop agents that not only act intelligently but also learn and respond to new circumstances in intelligent ways.

Keywords:

Agentic Ai, Multimodal Reasoning, Autonomy, Reinforcement Learning, Symbolic Ai, Artificial Intelligence.

Article History:

Received: 11.07.2021

Revised: 08.08.2021

Accepted: 01.09.2021

Published: 06.09.2021

1. Introduction

The term "citizen developer" has quickly gone from being a buzzword to a major change in how businesses use technology today. A citizen developer is a business user, not a professional software engineer. They work in fields like finance, marketing, HR, or

operations and utilize low-code and no-code (LCNC) platforms to build applications, automate tasks, and solve business problems. These people don't write thousands of lines of code from scratch. They make good solutions for their firms by using drag-and-drop interfaces, pre-made parts & guided by their templates. These solutions are typically quite important. The citizen developer is revolutionary not just because of their technical skills, but also because they can link business needs with technology execution, which is often hard for traditional IT frameworks to do. To understand the importance of citizen development, you need to look at the history of traditional IT. For many years, only professional programmers were able to make applications. Business teams would figure out what they needed, spell out what they needed, and ask IT for it. These requests typically added to a long list of things to do, with priority based on how many resources were available, how easy it was to do the work, and how much money was available. So, it can take months or even years for a project to come up with a working solution. The corporate world may have changed by the time an application was finally put into use, making the solution less useful. Both parties were angry because the company needed things done quickly and IT was taking too long to deliver. IT staff were burdened by demands they couldn't meet quickly enough, and business units thought that their needs weren't being handled quickly enough.

Three main things are closely related to the rise of citizen developers: digital transformation, a lack of IT workers, and a growing need for quick fixes.

Digital transformation has revolutionized how businesses compete with each other. All businesses today, no matter what field they are in, are under pressure to innovate digitally. This may include making apps for customers, improving internal procedures, or using data more effectively. This change means that application development has to happen faster and in shorter cycles, rather than relying on traditional development processes.

Second, there aren't enough skilled IT professionals in the globe. Developers that are skilled in modern frameworks, cloud-native architectures, and corporate security are in great demand but hard to find. This scarcity has made the gap between what organizations need and what IT can really do even bigger. Many companies realized that their goals for digital transformation would not be reached until they found new ways to build and use applications.

The need for quick answers has grown. Every month or perhaps every day, new problems arise in the corporate world. A marketing team could need a campaign dashboard right once, while an HR team would need a quick way for employees to provide feedback so they can adjust to new working conditions. Waiting months for IT to provide these features is no longer an option. As a result, staff members with firsthand knowledge of the problem are stepping in to create applications on their own, utilizing platforms designed to make it easier to get started.

Figure 1. The Rise of Citizen Developers

Low-code and no-code (LCNC) systems fall within this category. Microsoft Power Apps, Mendix, and OutSystems are examples of platforms that have made it easier for everyone to build apps. They provide easy-to-use interfaces that let people who don't know how to code construct data models, connect to business systems, and start applications without too much trouble. Low-code and no-code platforms are meant to help professional developers, not replace them. They free up IT from boring, simple tasks so they can

focus on more strategic and difficult ones. At the same time, they let business users deal with problems as they come up. This change in who is responsible for developing applications has had a big impact on how companies innovate.

This article contends that citizen developers have revolutionized company innovation via the democratization of application development. Innovation is no longer limited to IT; it is now spread across the whole company. People that work in business & are closest to the problem are not just identifying these needs, but also coming up with their solutions. This has sped up the process of becoming digital, improved communication between IT and business, and let people who weren't involved in software development before be more creative. There are difficulties with this movement, such as governance, security & scalability. However, the benefits are more clear: citizen developers have changed the game. The rise of citizen programmers is an indication of a bigger change in both culture & technology. It has to do with moving from a place where IT controls innovation to one where innovation is everyone's responsibility. The democratization of development goes beyond just being more efficient; it completely changes how businesses deal with these problems & compete in the digital age. The story of citizen developers goes beyond just technology; it includes empowerment, working together & how jobs are changing.

2. The Rise of Citizen Development

The idea of "citizen development" may appear latest, but it has deep roots in a basic truth: those who are closest to the problem usually know the most about it. For decades, businesses relied on their IT staff and software vendors to create any digital solution, no matter how big or little. This caused delays, extended wait times & solutions that didn't always meet the needs of the firm. The introduction of low-code/no-code (LCNC) platforms has completely changed the game. Now, people who don't know how to code may create apps, processes & automations that solve their everyday problems directly.

2.1. Evolution of LCNC Platforms

The earliest attempts to give non-developers more control happened when Excel macros & Visual Basic for Applications (VBA) were new. These technologies gave business analysts and managers the tools they needed to automate tasks and build small apps without having to know a lot about coding. Software companies realized that they needed to provide development tools that were easier to use, so they started developing platforms that focused on drag-and-drop design, prebuilt components, and visual interfaces.

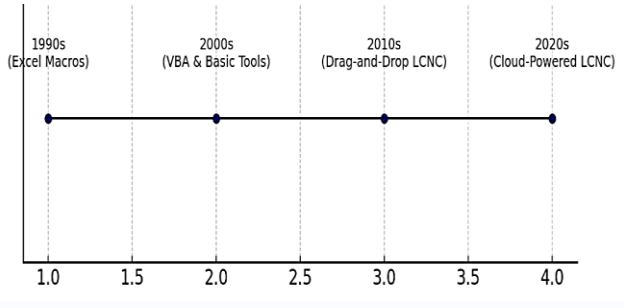

Figure 2. Evolution or LCNC Platforms

Modern LCNC platforms take this idea a big step farther. They let business users build whole applications that involve data integration, process orchestration & advanced analytics, not just little automations. Microsoft Power Apps, Salesforce Lightning, Appian & Mendix are just a few of the platforms that make it possible to build these enterprise-level apps in weeks instead of months. It's clear that there are benefits: companies may meet business needs more quickly, reduce their IT backlog & encourage staff to come up with the latest ideas. The great power of LCNC comes from its capacity to cross a long-standing division. On one side are developers and IT experts who are really good at what they do but don't have a lot of time. On the other hand, corporate users have a lot of knowledge about how things work, what customers are unhappy about, and what gets in the way of operations. LCNC tools act as a bridge, making it easier for both sides to work together. IT still provides frameworks for security, compliance, and scalability, but business users are becoming more free to solve their own problems.

2.2. Role of Cloud Adoption

The rise of citizen development is closely tied to the rise of the cloud. Before the cloud age, it was common for software to be distributed using physical servers, long setup times, and a lot of work amongst IT teams. This made it costly to try things out and kept non-technical workers from trying to come up with solutions.

Cloud platforms changed everything. Infrastructure was no longer a problem all of a sudden. Low-code and no-code solutions have become web-based services that may grow as needed. A business analyst may develop a procedure in the morning and send it to coworkers across the globe by the afternoon, all without using a server or writing deployment scripts. Cloud technology has made it easier to connect LCNC systems, which has made them more accessible. Modern cloud systems typically include APIs and connections that LCNC tools may leverage via drag-and-drop integrations. A citizen developer may connect a customer relationship management (CRM) system, an ERP platform, and a communication medium like Slack or Teams without having to write complicated code. The cloud has made processing power more accessible to everyone, which is similar to how software development has become more accessible to everyone. Cloud services made it easier for new businesses to compete with big ones, while Low-Code/No-Code (LCNC) platforms have made it easier for workers to build applications that used to need a complete IT project team.

2.3. Pandemic Acceleration: Remote Work and the Demand for Automation

The COVID-19 pandemic was a major reason why LCNC grew so quickly in the late 2010s. Companies need to quickly switch to working from home, collaborating online, and using new business models. Many tasks that used to be done in person, including onboarding new employees, processing documents, or helping customers, needed to be done quickly online. IT departments, which were already stretched too thin, couldn't handle the sudden surge of requests. This is where citizen developers stepped in. Business teams began testing LCNC technology to automate repetitive tasks, digitize documents, and improve operations. A human resources manager can build a simple leave request app in only a few days. A customer service manager may develop a chatbot to answer common questions. A financial analyst may make dashboards that provide real-time information about expenditure without needing to be part of a formal IT project.

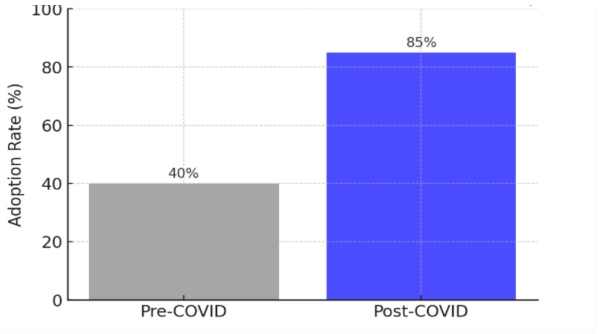

Figure 3. Impact of COVID-19 on Technology Adoption Rates

The urgency of the outbreak broke down traditional barriers. Companies that may not have wanted to allow non-technical people create apps before had few choices. The other option was inaction. The momentum grew unstoppable when they saw how quickly LCNC equipment delivered value. What began as emergency plans turned into long-term strategies. A lot of companies still use the citizen development strategy to be flexible in a world that is changing quickly.

3. Impact on Enterprises

The rise of citizen developers has changed how businesses use technology, solve many problems & encourage the latest ideas. Citizen development lets non-IT people take charge of making solutions that directly fulfill business needs, which is different from traditional IT-driven development procedures that frequently include backlogs, bottlenecks & big dependencies. This change has had a big impact on businesses by lowering expenses, encouraging the latest ideas, and making it easier for people to work together.

3.1. Reduced IT Backlog

A major & easy-to-see benefit of citizen development is that it cuts down on their IT backlogs by a lot. IT professionals at many companies are often juggling several tasks, such as keeping outdated systems running, dealing with security issues, maintaining

infrastructure, and responding to urgent requests. Putting all application development in IT creates bottlenecks by default. Citizen developers help with this by making smaller, department-specific applications on their own utilizing low-code & no-code platforms. Instead of waiting months for IT to provide them the tools they need, the finance team could create an app to keep track of expenses, or HR could set up a system for onboarding new employees in only a few weeks. This doesn't change what IT does; it just lets them focus on big, important projects while smaller, more urgent needs are addressed at the business unit level. The outcome is a better balance: IT managers spend more time upgrading infrastructure & making sure security rules are followed, while citizen developers quickly build these apps that their teams can use. Companies don't have to choose between stability & the latest ideas anymore; they may have both.

3.2. Faster Prototyping and Time-to-Market

In today's competitive industry, speed frequently sets market leaders apart from those who are behind. The traditional process of developing software—gathering requirements, assigning developers, testing, and deploying—can take months or even years. Market needs may have altered by the time a solution is put into place. Citizen coders speed up this process a lot with simple drag-and-drop tools. They can quickly make prototypes, test them with real users, and make changes almost right away. A retail team might build a form for customer feedback in a few hours, use it the following day, and improve it depending on what customers say within a week. It's hard to get this kind of flexibility in standard development cycles.

Also, citizen development is about more than just convenience; it's also about importance. The people who are making the solutions usually face the same business problems themselves, which leads to applications that are more suited to real needs. People that deal with the problem every day come up with solutions, not IT's misunderstanding of requirements and the true problem. This sped up the development and implementation process, which cut down on the time it took to go to market. This gave businesses the ability to respond to customer demands, changes in regulations, and competitive difficulties with unmatched speed.

3.3. Cost Reduction and Resource Optimization

Companies are always under pressure to make the most of their resources & keep expenses down. It expenses a lot to hire experienced developers, keep up with infrastructure & manage protracted development cycles. Citizen development is a great way to save money without lowering the quality of the product. Giving employees business knowledge so they can come up with these solutions means that companies don't need to hire outside consultants or extra IT staff. Instead than paying for outside development, companies may let business users build big projects for simple tasks like automating reports, managing team schedules, or getting internal approval. Reallocating IT resources to focus on higher-value projects adds another layer of efficiency. When IT isn't busy with regular application requests, they can spend more time on important areas like digital transformation, cybersecurity, data strategy & others. This change in jobs makes sure that skilled developers & architects aren't wasted on simple apps that other people can easily handle. The overall impact is big: companies save money, make their employees more skilled & get more out of their technology investments.

3.4. Increased Employee Engagement and Innovation Culture

Citizen development has a big impact on the culture of an organization, in addition to making it more productive & very less expensive. Employees who previously felt limited in their job duties now have the chance to actively change how things are done at work. This empowerment makes people feel more like they own & are involved in something. When an operations team member creates an app to automate manual reporting, they go beyond being just an "end user" and become an invention that boosts productivity for the entire team. This recognition motivates employees, boosts morale & inspires others to come up with the latest ideas. The ripple effect might be big: when one department shows promise, others begin to try, which encourages a culture of innovation. Citizen development makes it much easier for everyone to help solve these problems. Innovation shouldn't be limited to a small group of experts or leaders; instead, it should include everyone in the company. When the right rules are in place, this shared invention becomes a big edge over the competition.

Companies have better employee retention, better collaboration & a workplace culture that encourages initiative when employees feel that their ideas are valued & they have the resources they need to make them happen. This change in culture is a big plus at a time when hiring & keeping individuals is becoming harder.

4. Challenges and Risks

Citizen development has made software development more creative, efficient, and open to everyone. But when it comes to making big changes to how a business works, it has a lot of challenges & risks. No-code and low-code platforms let employees solve many problems without having to go through busy IT departments. However, the democratization of programming also brings up huge problems that businesses need to be aware of. Understanding these issues is the first step to being able to handle them well.

4.1. Shadow IT and Governance Issues

"Shadow IT" is a big worry that comes with the expansion of citizens. This has to do with these technical solutions—apps, processes, or integrations—that are created & used outside the IT department's direct control. Even while shadow IT is sometimes done with these good intentions (such when employees want to solve problems faster), it may make the IT ecosystem less cohesive. If different teams create their own apps without following the rules, the company can end up with these duplicate tools, systems that don't work together & data stores that are separate from each other. Instead of making things better over time, this makes them very less efficient. More importantly, IT loses sight & control over the IT environment, which makes it harder to maintain their governance, ensure compliance, and help people when things go wrong.

To prevent this danger, companies need to find a balance between giving individuals more authority & keeping protections in place. Establishing governance frameworks, clear approval processes & centralized their oversight systems safeguards innovation from devolving into chaos. Citizen development should not take the place of IT; it should work with it.

Figure 4. Risk Matrix

4.2. Security and Compliance Concerns

Security is a big challenge. When applications are built outside of these standard IT frameworks, they may not have enough security, data protection, or compliance requirements. A citizen developer could accidentally connect sensitive client information to a third-party service that isn't secure, which might lead to breaches or violations of their rules. This risk is higher in fields that are very extensively regulated, including banking, healthcare & government. Regulations like GDPR, HIPAA, and PCI DSS make it very hard to maintain their information. If citizen-developed apps don't meet such standards, companies might face legal action, damage to their brand, or a loss of customer trust. Organizations need to provide citizen developers safe platforms that also include compliance features. Also, teaching people about established procedures for data privacy & the security might help prevent accidents. The goal is not to stop innovation, but to make sure it happens in a safe & the responsible way.

4.3. Scalability Limitations of No-Code Apps

No-code platforms are good for getting things done quickly, such as automating repetitive processes, making workflows more efficient, or building departmental apps. Still, scalability is still a problem. A lot of apps made by citizens are not built to handle a lot of usage, complicated integrations, or huge amounts of information. What works for a small group may not work as well, or at all, when it's used by the full company. A simple no-code inventory tracker could work well for a single branch, but it might not work as effectively when hundreds of individuals from many other different places attempt to use it at the same time. Similarly, performance

limits or a lack of integration options might make it hard to accept the system or force IT to completely rebuild it. This does not mean that no-code is inherently flawed; rather, it has constraints. Organizations should think of solutions made by citizens as prototypes or tools for departments, but IT should still be in charge of these enterprise-level needs. IT may help scale a no-code software using more advanced development methods if it shows that it is useful. So, citizen development is like a lab for new ideas, while IT makes sure that things stay stable over time.

4.4. Skills Gap: When Citizen Developers need IT's Help

No-code platforms make it easier to use technology, but they don't get rid of all of the problems. Many customers of companies don't understand how hard it is to make reliable software. Some of the problems include data modeling, process optimization, error management & working with these business systems, which may need specialized expertise. Citizen developers may run into many problems while trying to fix or keep their apps running. A tool that works well now could not work after a platform update, or it might need to be changed as business needs evolve. Without technical knowledge, non-developers may quickly run into problems. This is when you really need IT support. IT teams shouldn't consider citizen developers as competitors; they rather see them as mentors & partners. Using a "fusion team" paradigm, where business users and IT specialists work together, makes sure that solutions are both creative & technically sound. This alliance fills in the skills gap and turns citizen developers into important collaborators in the digital transformation.

4.5. Balancing Speed with Enterprise-Grade Standards

The main problem with citizen development is finding the right balance between speed & quality. The main reason people like no-code platforms is because they can quickly turn the latest ideas into tangible things. Still, speed may hurt maintainability, documentation & standards at the business level. A citizen developer could put off long-term goals like security, scalability, or interoperability in favor of more immediate needs, like fixing a problem for their team. This might result in a disjointed collection of solutions that work in the near term but create technical debt over time. Organizations need to find ways to balance flexibility with order. This may mean that citizen developers need to keep track of their apps, follow particular design rules, or go through short tests before they can be released. By adding important yet useful checkpoints, firms may quickly come up with new ideas while still being reliable.

5. Governance, Best Practices, and Frameworks

Citizen development has made companies more flexible, but this freedom means that they need to keep their systems safe, compliant & sustainable. Good governance and best practices are the basis for successful citizen development initiatives. They protect against broken solutions & hidden dangers that might come from the passion & creativity of non-technical builders. The balance between innovation and oversight is primarily based on how well IT teams, business units & citizen developers work together.

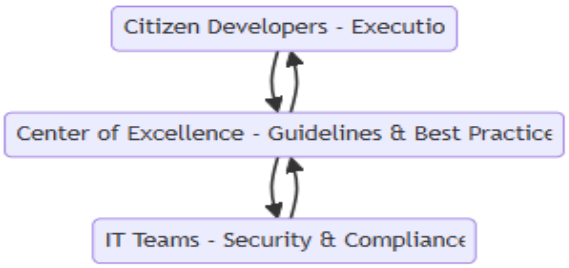

Figure 5. Governance Best Practices and Frameworks

5.1. The Role of IT in Enabling Citizen Developers

IT teams have changed from just protecting technology to actively helping people use it. Their job is to put up the right platforms, make sure that citizen developers can get to their information & tools safely, and make sure that citizens can get to information & tools safely. IT doesn't set limits on users; instead, it acts as a mentor and adviser, helping employees come up with the latest ideas while still following rules for security & compliance. IT reduces the danger of shadow IT by giving teams pre-approved parts, templates & APIs. This lets teams work quickly.

5.2. Establishing Guardrails through Policies and Security Models

Governance works best when people regard it as advice instead of a restriction. Data access rules, security models & the version controls act as guardrails, giving structure while yet allowing for their experimentation. Limiting access to sensitive customer information to only approved APIs, for example, reduces the chance of accidental leaks. Role-based access restrictions, on the other hand, make sure that people have the right amount of visibility. Clear rules let citizen developers understand their bounds without getting in the way of their work.

5.3. The Center of Excellence (CoE) Approach

A Center of Excellence strategy has been adopted by several groups to appropriately grow citizen development. A Center of Excellence is a place where you may find best practices, templates, training, and governance frameworks. It brings together information while allowing creativity to happen in different places. Instead of doing the same thing again, teams may utilize the CoE's tools to speed up development and improve assurance. Over time, the CoE becomes the cultural foundation for citizen development, making sure that everyone in the firm is on the same page while also encouraging innovation.

5.4. IT-Business Partnership Models

When IT and business departments work well together, citizen development thrives. This way of working together recognizes that each side brings something different to the table. IT brings technical oversight and security expertise, while business users provide subject matter expertise and firsthand understanding of what customers need. Joint governance committees, collaborative planning sessions, and regular reviews are good ways to bring both sides together. This collaboration goes beyond the "us versus them" mindset and creates an environment where everyone is responsible for coming up with new ideas.

5.5. Training, Certification, and Upskilling

An effective citizen development program puts as much money into its people as it does into its technology platforms. Training programs & credentials help citizen developers get better at their jobs & make them feel more responsible. Workers may improve their skills in application development, data analysis & the automation via workshops, online courses, and peer-to-peer learning groups. Upskilling makes sure that solutions are the latest, long-lasting, and ready to grow & that they follow company rules.

5.6. Bringing It All Together

In citizen development, governance is more about creating a safe space for the latest ideas to grow than about controlling people. Companies may protect the integrity of their systems & data while getting the most out of their employees by utilizing IT as a framework, setting clear rules, keeping a strong Center of Excellence & giving employees ongoing training. Citizen developers can come up with the latest ideas in a way that lasts since they are both empowered & supervised.

6. Case Studies

Citizen development is not just a passing trend; it has become a practical approach that companies in many fields are embracing to solve real problems more quickly and cheaply. The next four case studies show how low-code/no-code (LCNC) technologies have made it possible for people who don't work in IT to come up with important solutions.

6.1. Case Study 1: Banking Sector – Customer Onboarding Automation

6.1.1. Context/Problem:

Retail banking is a highly regulated field that frequently needs a lot of steps to get a new customer, such as verifying their identity, following KYC (Know Your Customer) rules, doing background checks & setting up an account. In the past, these steps needed to be documented by hand, departments had to work together & IT personnel had to be heavily relied on to keep an eye on these workflow systems. What is the result? It might take days or even weeks for a new customer to be set up, which can make them angry & cost you business.

6.1.2. Solution via Citizen Development:

A well-known retail bank began a low-code platform & told a small number of these operations managers to create and carry out onboarding processes on their own. They were able to create an app that automatically collected their documents, worked with compliance databases & gave workers and customers actual time updates on progress, all without having to write complicated code.

6.1.3. Impact/Metrics Achieved:

The results were quite important. The onboarding process became shorter by nearly half, going from an average of 10 days to less than four. The technology automatically found missing information or differences, which made compliance inspections easier. Employees used to have to manage a lot of operations by hand, but now they can focus on helping their customers. Most importantly, the IT group didn't have to build and maintain custom onboarding solutions anymore, so they could concentrate on big-picture tasks. The bank said that customer satisfaction scores had gone up and that a lot more people were opening new accounts.

6.2. Case Study 2: Healthcare Sector – Appointment Scheduling with No-Code

6.2.1. Context/Problem:

Healthcare organizations typically have trouble organizing appointment scheduling. People often handle reservations by hand or using old procedures that are hard to adapt and don't allow for much flexibility. Patients may have to wait a long time, have problems rescheduling, or not know enough about the appointments that are available. On the IT side, requests for changes to the system pile up, causing delays & unhappy patients.

6.2.2. Solution via Citizen Development:

To solve this problem, a hospital let a small group of administrative staff utilize a no-code platform. In only a few weeks, they built an online scheduling software that lets patients book, change & cancel appointments directly via a web interface. The technology was connected to the hospital's electronic health records (EHR), which made sure that physicians were constantly available. Automated alerts helped people keep their appointments, and staff could simply update the system without having to ask their engineers for help.

6.2.3. Impact/Metrics Achieved:

There was no delay in the change. About 30% fewer patients missed their appointments, and the average wait time for each appointment dropped by 50%. The hospital's IT personnel thought it saved hundreds of hours of development time since citizen developers could handle most of the changes on their own. The new system was easy to use, according to patient satisfaction surveys, and doctors said it made scheduling easier. The trial showed that giving healthcare management the right tools might improve the patient experience without spending a lot of money on technology.

6.3. Case Study 3: Government and Smart Cities – Enabling Public Service Apps

6.3.1. Context/Problem:

Local governments regularly get requests to speed up the delivery of services to residents, such as issuing permits and setting up waste pickup schedules. Still, money problems and long procurement processes make it hard to put new digital services into place. It might take months or years for traditional IT development to finish, which means that important improvements that communities really need will be delayed.

6.3.2. Solution via Citizen Development:

In a smart city initiative, city workers who didn't know how to code were taught how to make applications utilizing low-code platforms. A transportation official devised an app that lets passengers keep track of bus delays and get real-time updates. Another small team built a service request system that lets customers report pothole concerns, keep track of repairs, and receive alerts. These programs made in the U.S. avoided costly vendor contracts & were made to fit the needs of the community.

6.3.3. Impact/Metrics Achieved:

The outcome was twofold: citizen services were delivered faster & expenses were cut significantly. It used to take several months to make new applications, but now it just takes a few weeks. Citizens liked that actual time information was exposed to everyone, which made them trust the government's reaction more. The city cut its IT costs by moving money from outside development projects to educating and helping citizen developers. This strategy also made government institutions more creative, since employees felt free to solve these problems on their own.

6.4. Case Study 4: Retail – Supply Chain Tracking Dashboards

6.4.1. Context/Problem

The success of retail businesses depends a lot on how well their supply chains work. Any problem, such as delayed shipments, changes in demand, or issues with these suppliers, may affect inventory levels & customer satisfaction. In the past, supply chain data

was spread out across these several systems and spreadsheets, which meant that IT teams had to develop reporting tools that didn't always meet actual time needs. Managers didn't know much about how quickly things were changing, which made it hard for them to respond quickly.

6.4.2. Solution via Citizen Development

A big retail firm chose low-code solutions so that its supply chain managers could make their own dashboards. These displays combined actual time information from sales systems, warehouse & logistics partners into one view.Managers may set up alerts for delays, check on how inventory is moving, and adjust their purchase strategy on the fly. We could make modifications to the dashboards straight away since we created them ourselves instead of having to wait for IT cycles.

6.4.3. Impact/Metrics Achieved

The outcomes were great. The business asserted that it could now deal with market changes, including abrupt increases in demand or supply delays, 20% better than previously. As management learned more about how much stock they had and stopped purchasing too much, the cost of inventory went down. Allowing non-technical workers to create and alter their own dashboards made the company more adaptable, especially during peak shopping seasons. IT leaders saw that the strategy made the supply chain more flexible and encouraged people from different departments to work together since managers were in charge of solutions that directly influenced their performance.

7. Conclusion

Citizen developers have changed the way businesses come up with the latest ideas in a big way by showing that anybody can make these solutions, not just IT staff. Their rise shows that workers who are closest to the problems can also come up with answers if they have the right tools, training & help. This change has not only sped up innovation, but it has also made it more useful since the people who make these procedures & applications know what actual problems their teams & customers are having. One important thing to learn from this movement is that giving people authority makes them more flexible. When workers have the freedom to try the latest things and come up with the latest ideas, they do so that they would not have thought of before. Citizen development has made it easy for everyone to produce technology. This allows companies to operate quicker without having to rely on a limited number of IT specialists. This change won't be easy. It is important to find a balance between freedom & control since too much freedom may lead to these security holes, broken systems & problems with following the rules.

Companies need to carefully balance giving citizen developers the tools & freedom they need with making sure that IT puts in place strong protections. It's clearer for businesses: to be competitive in a world where speed and flexibility are important, they need to support citizen growth. Companies that offer their workers greater power and push them to act ethically will unearth hidden talent and create a culture of creativity that lasts. In the end, citizen developers are more than just a passing trend; they are the future of work. They are a place where many people work together to create their technology instead of just a few. By intentionally embracing this change, companies may create a more inclusive, imaginative & more resilient approach for handling future difficulties.

References

[1] McKendrick, Joe. "The rise of the empowered citizen developer." *New Providence, NJ: Unisphere Research* (2017).
[2] De Lange, Michiel. "The playful city: Using play and games to foster citizen participation." (2015): 426-434.
[3] Lerner, Josh A. *Making democracy fun: How game design can empower citizens and transform politics*. MIT Press, 2014.
[4] Ng'ambi, Dick. "A Case of Citizen Developers." *Critical Mobile Pedagogy: Cases of Digital Technologies and Learners at the Margins* (2020).
[5] Patel, Piyushkumar, and Hetal Patel. "Lease Modifications and Rent Concessions under ASC 842: COVID-19's Lasting Impact on Lease Accounting." *Distributed Learning and Broad Applications in Scientific Research* 6 (2020): 824-41.
[6] Gaventa, John. "Finding the spaces for change: a power analysis." *IDS bulletin* 37.6 (2006): 23-33.
[7] Allam, Hitesh. *Exploring the Algorithms for Automatic Image Retrieval Using Sketches*. Diss. Missouri Western State University, 2017.
[8] Green, Duncan. *From poverty to power: How active citizens and effective states can change the world*. Oxfam, 2012.
[9] Guntupalli, Bhavitha. "Clean Code in the Real World: Principles I Actually Use". *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 1, no. 1, Mar. 2020, pp. 66-74
[10] Cornwall, Andrea. "Making spaces, changing places: situating participation in development." (2002).
[11] Patel, Piyushkumar. "Remote Auditing During the Pandemic: The Challenges of Conducting Effective Assurance Practices." *Distributed Learning and Broad Applications in Scientific Research* 6 (2020): 806-23.
[12] Mohammad, Abdul Jabbar. "Blockchain Ledger for Timekeeping Integrity." *International Journal of Emerging Trends in Computer Science and Information Technology* 1.1 (2020): 39-48.

[13] Greif, Avner, and David D. Laitin. "A theory of endogenous institutional change." *American political science review* 98.4 (2004): 633-652.
[14] Guntupalli, Bhavitha. "How I Debug Complex Issues in Large Codebases". *International Journal of Emerging Research in Engineering and Technology*, vol. 1, no. 1, Mar. 2020, pp. 67-76
[15] Anthropy, Anna. *Rise of the videogame zinesters: How freaks, normals, amateurs, artists, dreamers, drop-outs, queers, housewives, and people like you are taking back an art form*. Seven Stories Press, 2012.
[16] Anand, Sangeeta. "Integrating Blockchain for Securing and Auditing Patient Eligibility Data in CHIP." *International Journal of Emerging Trends in Computer Science and Information Technology* 1.1 (2020): 57-65.
[17] Shaik, Babulal. "Network Isolation Techniques in Multi-Tenant EKS Clusters." *Distributed Learning and Broad Applications in Scientific Research* 6 (2020).
[18] Juul, Jesper. *A casual revolution: Reinventing video games and their players*. MIT press, 2012.
[19] Arugula, Balkishan, and Sudhkar Gade. "Cross-Border Banking Technology Integration: Overcoming Regulatory and Technical Challenges". *International Journal of Emerging Research in Engineering and Technology*, vol. 1, no. 1, Mar. 2020, pp. 40-48
[20] Bennett, W. Lance. "Changing citizenship in the digital age." 2008,
[21] Patel, Piyushkumar. "Bonus Depreciation Loopholes: How High-Net-Worth Individuals Maximize Tax Deductions." *Distributed Learning and Broad Applications in Scientific Research* 5 (2019): 1405-19.
[22] Castronova, Edward. *Exodus to the virtual world: How online fun is changing reality*. MacMillan, 2008.
[23] Guntupalli, Bhavitha. "Code Reviews That Don't Suck: Tips for Reviewers and Submitters". *International Journal of Emerging Research in Engineering and Technology*, vol. 1, no. 2, June 2020, pp. 60-68
[24] Andrews, Matt. *The limits of institutional reform in development: Changing rules for realistic solutions*. Cambridge University Press, 2013.
[25] Gabrys, Jennifer. "Programming environments: Environmentality and citizen sensing in the smart city." *Environment and planning D: Society and space* 32.1 (2014): 30-48.
[26] Mathie, Alison, and Gord Cunningham. "From clients to citizens: Asset-based community development as a strategy for community-driven development." *Development in practice* 13.5 (2003): 474-486.