

Five Prerequisites for Citizen Development



In a previous article we talked about evolving *Citizen Developers from City Dwellers to Townsmen ... so they can become first class citizen*. We discussed the left-shift of responsibilities from pro-developers to Citizen Developers and explained that Low-Code / No-Code (L-C/N-C) platforms cannot exonerate an organization of all the responsibilities shifted to these Citizen Developers:

*It is the tooling support of the L-C/N-C platforms
that evolve Citizen Developers from City Dwellers to Townsmen.*

*It is the organization's (city) strategy and tactical focus
that turns them into first class citizens.*

Next we introduced 3 prerequisites for success the so-called **D³ or Data, Devices and Delivery**.

After some more research it became clear that still some dimensions were missing to cover all the angels of people - processes - technology.

Firstly, we want to **extend Data with Digitalization. Data is a necessary but not a sufficient condition.** Data as a collection of information is nothing without a clear view where it fits within business processes. It is here where **Digitalization comes in to add the process angle as sufficient condition.**

Secondly, in order to have an **acceptable ROI** for a L-C/N-C platform investment, **a high level of reusability is important.** Reusability focusses on technical building blocks like components and connections but also on governance and policies. The idea that there are things that will be **application-specific and handled by a Citizen Developer** and other things will be **generic and handled on an organizational level**, introduces a dimension where we look at **the distribution of reusable elements throughout the L-C/N-C platform.**

The two additions mean that we have now **D⁵** model:

Data - Devices - Delivery - Digitalization - Distribution

Data: data encapsulation/exposure is a prerequisite



For a Citizen Developer to develop applications for his business, he needs access to his business data

- **Data source isolation:** data specific for his business
- **Data availability:** data accessible through API's
- **Data storage:** data to be stored in relation to his application

Devices: suitable devices are a prerequisite



For a Citizen Developer to build applications, he needs to be able to select a suitable device for the problem at hand i.e. desktop, mobile, VR-AR, kiosk

- **Device availability:** devices made available on short-term to all the users of his application
- **Device/License sharing:** devices and licenses ad-hoc attributed and revoked to users
- **Device security policies:** devices are managed to protect against exposure and loss of business data

Delivery (Deployment): platform governance and strategy is a prerequisite



For a Citizen Developer to manage the life cycle of his application, he needs to be able to count on supporting process to be in place

- **Application Delivery TOM/SOM:** application delivery processes clearly describe roles and responsibilities for Business and IT
- **Delivered Application Support Model:** delivered application he can support directly or indirectly
- **Application DevSecOps policies:** business risks and IT risks are controlled by DevSecOps tooling

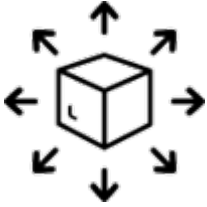
Digitalization: process oriented automation is a prerequisite



For a Citizen Developer to contribute to the digitalization he needs to be able to affect end-to-end processes he owns

- **Monetized Digitalization:** processes direct and indirect costs and revenues are quantified (FinOps)
- **Dematerialized Digitalization:** processes descriptions focus on activities (steps towards end-results) before information (content carried between activities)
- **Optimized Digitalization :** end-to-end processes are defined as a set of interrelated activities (process optimization not isolated activity optimization)

Distribution: availability of reusable building blocks is a prerequisite



*For a Citizen Developer to efficiently assembly his application
he needs to combine distributed building blocks*

- **Distributed Components:** reusable front-end components (UI elements) as well as reusable back-end components (Systems/Solutions)
- **Distributed Connectivity:** reusable means of integration between components (data exchange, network connectivity, import/export)
- **Distributed Responsibilities:** reusable cross application responsibilities (generic governance) extending application specific responsibilities of his application (app specific governance)