

# Could No-Code Be Code?

## Toward a No-Code Programming Language for Citizen Developers

Assaf Avishahar-Zeira  
assaf@too.software  
TOO.Software  
Bnei Atarot 6099100, Israel

David H. Lorenz  
lorenz@openu.ac.il  
Dept. of Mathematics and Computer Science  
Open University of Israel  
Ra'anana 4353701, Israel

### Abstract

By 2030 for each filled position in *Software Engineering*, two positions would remain unfilled. This already apparent loss of productivity has the software industry scrambling to fill the missing positions with *citizen developers*—technical people with little or no programming skills—who would be using No-Code platforms to program various software solutions in specific domains. However, currently available platforms have fairly limited abstractions, lacking the flexibility of a general purpose programming language.

To break the No-Code abstraction barrier, a very simple yet expressive general purpose No-Code programming language might provide citizen developers with an alternative to domain-specific No-Code platforms. Unfortunately, these requirements seem contradictory. Making a language very simple and specific might render it crippled, thus limited to a certain domain of problems. Conversely, making a language very expressive and general, might render it too complicated for citizen developers.

In this work we argue that a multi-paradigm minimalist approach can bridge the gap between simplicity and expressiveness by including only abstractions considered intuitive to citizens. As a concrete proof-of-concept, we present a general purpose programming language designed for citizen developers that is on the one hand very powerful and on the other hand very simple. In fact, this language is so simple that the entire development is accomplished by flowcharts using mouse actions only, without typing a single line of code, thus demonstrating a general purpose No-Code programming language candidate for citizen developers.



This work is licensed under a Creative Commons Attribution 4.0 International License.

*Onward! '23*, October 25–27, 2023, Cascais, Portugal

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0388-1/23/10.

<https://doi.org/10.1145/3622758.3622893>

**CCS Concepts:** • Software and its engineering → Visual languages; Multiparadigm languages.

**Keywords:** Citizen Developers, Golang, No-Code Software Development, Projectional Editing, Programming Language Design.

### ACM Reference Format:

Assaf Avishahar-Zeira and David H. Lorenz. 2023. Could No-Code Be Code? Toward a No-Code Programming Language for Citizen Developers. In *Proceedings of the 2023 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! '23)*, October 25–27, 2023, Cascais, Portugal. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3622758.3622893>

## 1 Introduction

Today and for the foreseeable future, the supply of professional programmers cannot meet the demand for software engineers [7]. SlashData,<sup>1</sup> a leading analyst company in the developer economy, projects a total number of 45 million software engineers globally by year 2030. The U.S. Labor Department further estimates a shortage of 85 million engineers by that time, meaning that for each filled position two would remain unfilled, and that because of this shortage companies may lose \$8.4 trillion in revenue.

This shortage in programmers is pushing the software industry toward *No-Code tools* that enable software development by novice programmers and even non-programmers, generally referred to as *citizen developers* [29] (hereafter, *citizens*). These No-Code tools keep the promise of creating a solution without typing code, but they are by far less expressive than a full blown *general purpose programming language (GPL)*, lacking any pretension to be *Code*.

In fact, No-Code tools resemble hardware more than software development, missing the most important property of software being “soft.” In order to retain the “softness” property, they must be expressive like a programming language; that is, be *Code*. At the same time they must be simple enough for citizens; that is, to also be *No-Code*. This begs the question: *could No-Code be Code?*

<sup>1</sup><https://www.slashdata.co>