Chapter 5
The Democratization of Software Development and Design Thinking:
Governing Design Quality in Citizen Software Development

Maaike Stoops
https://orcid.org/0009-0003-5139-1806
LINKIT B.V., The Netherlands

Pablo Alfonso Aguilar Calderón
https://orcid.org/0000-0002-4939-0383
Universidad Autónoma de Sinaloa, Mexico

Óscar Manuel Peña Bañuelos
https://orcid.org/0009-0007-1241-9051
Universidad Autónoma de Sinaloa, Mexico

ABSTRACT

The ongoing democratization of software development, fueled by the proliferation of low-code/no-code platforms and AI-assisted tools, is fundamentally transforming the technological landscape by enabling non-technical citizen developers to create functional applications with minimal coding expertise. As organizations increasingly adopt these accessible development approaches to accelerate digital transformation, critical questions emerge regarding the potential erosion of established design principles, including user-centered methodologies and rigorous usability testing. This chapter systematically examines both the opportunities and challenges presented by this democratization movement, analyzing its impact on

DOI: 10.4018/979-8-3693-9531-8.ch005

software quality, maintainability, and user experience. The discussion culminates in practical recommendations for balancing accessibility with excellence, ensuring that democratized software ecosystems produce solutions that are not only rapidly deployable but also robust, scalable, and genuinely user-centric in an increasingly competitive digital economy

1. INTRODUCTION

In today's fast-evolving technological environment software development is becoming increasingly accessible to a wider audience. The rise of low-code and no-code platforms, combined with artificial intelligence (AI) integration, has allowed individuals with little formal training—referred to as "citizen developers" (Kirvan, P. et al., 2023)—to design and create software solutions. A practice previously reserved for specialists. As a result, almost anyone can now develop and deploy working software. This process of making a technology or practice available to anyone is known as democratization. In this case we are talking about the democratization of software development.

With the rise of citizen developers, the dynamics of the software development process and the roles within development teams change. Software development teams traditionally include multidisciplinary roles like developers, testers, a scrum master, and product designers. However, now that citizen developers can develop and deploy full software applications, employees from different parts of organizations are empowered to make software solutions. As described by Redhat in 2018 low-code platforms not only make software development more attainable, but also speed up the time in which working software can be developed. As a result, organizations that practice low-code development tend to follow fewer formal processes. Compared to traditional software development, I have also experienced that low-code development teams often have fewer and less formalized roles and responsibilities. Teams might lack a product designer, as low-code platforms provide many User Interface elements out of the box (Rokis & Kirikova, 2023).

Besides a shift in roles, the process of software development also changes. Traditionally software development teams used the Waterfall method, where the scope, budget and timeline are fixed upfront. To make software development more iterative, teams are trained to work through Agile frameworks and methodologies. However, since one person can fulfill different the roles in the team, the agility of software development reaches a new level. When one person can execute multiple, if not all, roles within the software development team, there is less inherit focus on methodologies and alignment. Such citizen developer can design, develop, test and deploy new functionality in a matter of hours or days rather than weeks or