Review Protocol for a Systematic Literature Review on the role of Low-Code Development Platforms for Software Development

Researcher: Lara Schwabe (lara.schwabe@s.wu.ac.at)

Supervisors: Assoc. Prof. PD Dr. Stefan Sobernig, Daniel Borst, MSc.

Institution: WU Vienna, Institute for Complex Networks

Document History

Version 1	Initial Draft, review model based on Zhang et al.	May 2025
Version 2	Adaptation of the sectioning based on exemplary SLR protocol provided by Daniel Borst	May 2025
Version 3	Change in Selection Criteria including scheme for determining relevance. Change in Time frame of review. Change in approach for QGS development	July 2025
Version 4	Pilot Study Completed; Development of initial search string for automated search; Selection of Databases	July 2025
Version 5	Finalizing Protocol: Automated Search details included; Formatting	October 2025

List of tables

I. Introduction

This protocol defines the methodology for conducting a systematic literature review (SLR) to evaluate the role of low-code development platforms (LCDPs) in software development. It adopts to the rigorous and replicable principles proposed by (Zhang et al., 2011), featuring the concept of Qasi-Gold Standards (QGS) and quasi-sensitivity to ensure search quality. Seeing that this protocol will be adapted over time, the Document History section on the first page intends to keep track of the significant changes made to this document itself as well as changes to the process.

II. Research Objectives and Questions

The SLR that stems from this protocol aims to answer the following research questions (RQs):

- **RQ1**: What are the primary drivers and barriers influencing the adoption of low-code platforms in the business context?
- **RQ2**: How do low-code platforms compare to traditional development approaches in terms of efficiency, effectiveness, and user preference based on empirical evidence?
- **RQ3**: What is the current state of research on low-code tooling, including study-design, research questions, and key observations?

III. Pilot Study Design

For the construction of the Quasi-Gold-Standard (QGS) a pilot study is conducted to suit the scope of the review. This pilot study aims to test the selection criteria and derive a high-quality QGS set which will be used to validate the automated search later via the use of quasi-sensitivity.

The procedure for the pilot study is as follows:

1. Search Execution: The initial search string is applied in Google Scholar, without applying filters. (n=22.200)
2. Result Pool: The first 40 search results from the QGS pool.
3. Application of Selection Criteria: Each study in this pool is assessed using the Selection Criteria defined in Section VII.
4. QGS Formation: All papers rated “Relevant” or higher are included in the QGS set. (n=9)

The search string used in the pilot study was derived directly from the key concepts of the research questions that guide this review:

("low-code" OR "no-code" OR “LCDP” OR “LCNC" OR "citizen development") AND (adoption OR challenges OR usability OR efficiency OR effectiveness OR comparison OR tooling OR "developer experience" OR "empirical study" OR "case study" OR survey)

Table a: QGS-Corpus

ID	Title	Authors	Year	Venue	Relevance Category
PS2	Practitioners' Perceptions on the Adoption of Low Code Development Platforms	Sebastian Käss; Susanne Strahringer; Markus Westner	2023	IEEE Access	Relevant
PS3	Low-code vs. the developer: An empirical study on the developer experience and efficiency of a no-code platform	Till Guthardt, Jens Kosiol, Oliver Hohlfeld	2024	MODELS Companion '24	Highly Relevant
PS7	Unleash the Power of Citizen Development: Leveraging Organizational Capabilities for Successful Low-Code Development Platform Adoption	Niculin Prinz, Melanie Huber, Justus Leonhardt, Constanze Riedinger	2024	Proceedings of the 57th Hawaii International Conference on System Sciences | 2024	Relevant
PS8	A Multiple Mini Case Study on the Adoption of Low Code Development Platforms in Work Systems	Sebastian Käss; Susanne Strahringer; Markus Westner	2023	IEEE Access	Highly Relevant
PS14	'To Code, or Not to Code': Unpacking the Understanding and Difficulties of Citizen Development Programs	Björn Binzer, Till Winkler	2024	INFORMATIK 2024 Conference Paper	Relevant
PS15	Citizen Development als Treiber der digitalen Transformation – Aktuelle Ansätze bei der Adoption von Low-Code Development Plattformen	Niculin Prinz, Melanie Huber, Constanze Riedinger & Christopher Rentrop	2023	HMD Praxis der Wirtschaftsinformatik	Relevant
PS18	Bridging Business and IT Through Low-Code/No-Code: Insights into Business-IT Collaboration in Enterprise Citizen Developer Programs	Björn Binzer, Daniel Fürstenau, Till Winkler	2025	Proceedings of the 58th Hawaii International Conference on System Sciences | 2025	Relevant
PS23	A Maturity Model for Managing Low-Code Development Platform Adoption	Niculin Prinz, Carsten Felden, Melanie Huber, Christopher Rentrop, Constanze Riedinger, Stephan Zimmermann	2025	PACIS 2025 Proceedings	Highly Relevant
PS32	Towards a Maturity Model for Adopting Low-Code Development Platforms	Niculin Prinz, Stephan Zimmermann	2024	PACIS 2024 Proceedings	Highly Relevant

IV. Initial Search String for Automated Search

With the goal of developing an effective and targeted search string for use during the automated search part of the SLR, a term frequency analysis was conducted based on the abstracts and titles of the primary studies included in the QGS, once the pilot study had been conducted. This approach aims to support the data-driven approach as recommended by (Zhang et al., 2011). The analysis follows these steps using a Python Script that can be found in the Supplementary Information section of this protocol:

1. Data Preparation: The abstracts and titles of all QGS studies are combined and converted to lowercase. Punctuation is removed and standard English stop words (e.g. “the”, “and”) are excluded using CountVectorizer from the scikit-learn library.
2. Frequency Analysis: Term Frequency (number of times each term appeared across all QGS entries) and Document Frequency (number of QGS entries in which each term appeared at least once) were computed
3. Term Ranking: The terms were ranked by total frequency, and the top 50 most frequent terms were exported to an excel file. These terms serve as a basis for identifying common phrases and words that are relevant to the research questions. Frequent terms such as platform, development, tool, developer, and application are the core in the low-code/no-code domain.

This approach ensures that the final search string is focused on the actual contents of the QGS and represents the terminology used in relevant literature. The search string that is developed with this method will be applied to the chosen databases and evaluated using quasi-sensitivity. It will then be refined, if necessary, as described in Section VI.

After the top 50 terms (by term frequency and document frequency) are extracted, terms with document frequencies ≤ 2 were removed. The remaining 37 terms are sorted by document frequency and then mapped to the three RQs based on their thematic alignment. This evaluation of terms can be found in the Supplementary Information Section of this protocol. The aim is to construct a search string that is balanced and representative of each RQ, therefore terms from each thematic group are chosen and combined into search string blocks.

The first block consists of core LC/NC terms, the second block combines terms related to the adoption, the third block consists of terms that indicate efficiency and comparison as well as the research landscape. Wildcards (*) are included to account for plural terms and variants.

("low-code" OR "lcdp" OR "lowcode")*
AND
(adopt OR implement* OR factor* OR organization OR management OR guidance OR business)*
AND
(efficien OR developer OR capability* OR develop* OR applicat* OR platform* OR tool* OR evaluation OR usability OR research OR study OR empirical OR user)*

The search string was tested in Google Scholar to ensure adequate scope. Compared to the roughly 22.000 results that the Pilot-Study-Search-String yielded, this one proved to be more precise, only returning around 8.000 studies.

V. Database Selection for the Automated Search

A preselection was chosen based on their relevance to the software engineering domain and their proven effectiveness in previous systematic literature reviews on low-code and no-code development. In particular, the reviews by (Rokis & Kirikova, 2022), (Prinz et al., 2021) and (Binzer & Winkler, 2022) showed these databases as consistently yielding relevant studies in the field. After the pilot study, when QGS papers were determined, the below databases were chosen so that the automated search yields minimal duplicates while ensuring that all QGS studies can be retrieved.

- IEEE Xplore
- ACM Digital Library
- AIS eLibrary
- GI DL
- DBLP

VI. Evaluation of Search Performance

The performance of the search string was evaluated using quasi-sensitivity:

$$Quasi - sensitivity = \frac{\#\ relevant\ studies\ retrieved\ in\ automated\ search}{\#\ QGS\ studies} * 100$$

The previously defined initial search string will be evaluated with the QGS using quasi-sensitivity. The threshold for quasi-sensitivity is set at 70%, meaning that iterative refinement will be applied to the search string until the threshold is met when evaluating the refined search string against the QGS studies.

VI.1 Search String Sensitivity Result

The following Table b summarizes the results of the quasi-sensitivity after the initial search string was applied to each database and the total result. Since the initially defined search string (Section IV) met the quasi-sensitivity threshold of 70% both in the individual databases and in total, no refinement was conducted.

Table b: Quasi-Sensitivity Results

Applied Search String: *("low-code" OR "lcdp*" OR "lowcode") AND (adopt* OR implement* OR factor* OR organization OR management OR guidance OR business) AND (efficien* OR developer OR capability* OR develop* OR applicat* OR platform* OR tool* OR evaluation OR usability OR research OR study OR empirical OR user)*

Database	QGS papers available	QGS papers found	Quasi-Sensitivity
IEEE Xplore	2	2	100 %
ACM DL	1	1	100 %
AIS eLibrary	4	3	75 %
DBLP[1]	1	1	100 %
GI DL[2]	1	1	100 %
Total	**9**	**8**	**89 %**

VII. Study Selection Criteria

These following criteria will be applied to all literature, during the initial Pilot Study for determining the QGS, and later automated search. They are based on the goals and scope of the SLR.

Firstly, the papers yielded by the search are screened for the following objective criteria:

- The study was published between 2023 – 2025.[3]
- The paper is a primary study.
- The full text is available in English or German.
- The study is a peer-reviewed journal article, conference paper or published thesis that demonstrates academic rigor.

Secondly, the studies that remain are screened for these subjective criteria:

- The study reports on primary research related to low-code or no-code development platforms in the context of software or business application development. Prior systematic reviews are only considered for background purposes and excluded from the main synthesis.
- The context of the research is Software Engineering.
- The study addresses at least one of the three research questions defined for this study.

To evaluate topic relevance, e.g. the subjective criteria listed above, each study is categorized according to the following schema (Table c).

[1] The syntax of the search string was changed to match database specific criteria: “(low-code|lcdp|lowcode) (adopt|implement|factor|organization|management|guidance|business) (efficien|developer|capability|develop|applicat|platform|tool|evaluation|usability|research|study|empirical|user)”

[2] The syntax of the search string was changed to match database specific criteria: “("low-code" OR "lcdp" OR "lowcode") AND (adopt OR implement OR factor OR organization OR management OR guidance OR business) AND (efficien OR developer OR capability OR develop OR applicat OR platform OR tool OR evaluation OR usability OR research OR study OR empirical OR user)”

[3] The review period was set to 2023 – 2025 to ensure the inclusion of the most recent developments in the field of low-code / no-code development. Existing literature reviews in the field, e.g. (Rokis & Kirikova, 2022), (Binzer & Winkler, 2022) focus on the period between 2014 – 2022 and can be found in the Supplementary Information of this protocol. By targeting the subsequent years, this review aims to complement prior work.

Table c: Relevancy Category Definition

Not relevant	No mention of LCDPs/NCDPs or completely unrelated domain.
Slightly relevant	Mentions low-code topics but lacks alignment with research questions.
Moderately relevant	Partially addresses a research question or related concept.
Relevant	Contributes to one RQ with empirical or conceptual content.
Highly relevant	Strong alignment with one or more RQs and high-quality evidence.

The ratings are based on the study's abstract, title and keywords, if present. If further information was needed for the categorization, the full text was taken into consideration for the categorization of each study. Only studies rated "Relevant" or higher are included both in the QGS-Corpus and in the final study selection.

VIII. Documentation and Transparency

All decisions regarding inclusion and exclusion of studies were recorded rigorously during the two search phases of the SLR. The metrics of the search phases can be found in the following sections. The following tables show in detail how many publications were removed from the QGS-Pool and from each database result-set. The columns map directly to the selection criteria defined for this literature review (Section VII).

VIII.1 Pilot Study

The QGS-pool consists of 40 publications from Google Scholar as described in Section III. Out of those 40 papers, 30 were excluded during the screening for objective criteria. The remaining 10 studies were checked for the subjective criteria and one more publication was removed, resulting in a QGS-Corpus consisting of 9 papers.

Table d: Exclusions during Pilot Study

QGS-pool	Date	Fulltext/ Language	Primary Study	Peer-reviewed	SE Context	Relevance	Included
40	13	4	8	5	1	0	9
40				**30**		1	**9**

VIII.2 Automated Search

In total, the database search yielded 1600 publications across the five selected databases. 1330 publications were excluded due to the objective selection criteria, leaving 270 papers to be screened for the subjective criteria. During evaluation of the subjective selection criteria, a further 204 publications were excluded. After the 66 remaining studies were identified, duplicates were removed to be included only once. The details regarding found duplicates can be found in Table f. Hence, the final number of publications included in the literature review is 62.

Table e: Exclusions during the Automated Search

Database	Results	Date	Fulltext/ Language	Primary Study	Peer-reviewed	SE Context	Relevance	Included
IEEE Xplore	283	113	47	22	3	53	35	10
ACM DL	954	433	134	272	29	1	69	16
AIS eLibrary	236	68	89	34	1	8	20	16
DBLP	50	22	16	2	0	0	2	8
GI DL	77	26	2	17	0	5	11	16
Total	**1 600**				**1 330**		**204**	**66**[4]

Table f: Duplicates found during automated search

Title	Author(s)	Venue	Database 1	Database 2
A Low-Code Approach for Simulation-Based Analysis of Process Collaborations	P. Bocciarelli; A. D'Ambrogio	Proceedings of the Winter Simulation Conference	ACM DL	IEEE Xplore
A Multiple Mini Case Study on the Adoption of Low Code Development Platforms in Work Systems	Sebastian Käss; Susanne Strahringer; Markus Westner	IEEE Access Vol. 11	IEEE Xplore	DBLP
Citizen Development als Treiber der digitalen Transformation – Aktuelle Ansätze bei der Adoption von Low-Code Development Plattformen	Niculin Prinz; Melanie Huber; Constanze Riedinger; Christopher Rentrop	HMD Prax. Wirtsch. Vol. 61	DBLP	GI DL
Practitioners' Perceptions on the Adoption of Low Code Development Platforms	Sebastian Käss; Susanne Strahringer; Markus Westner	IEEE Access Vol. 11	IEEE Xplore	DBLP

[4] Out of the final 66 included studies, 4 duplicates were removed, leaving a total of 62 publications to be included in the SLR.

IX.Data Extraction

During the data extraction phase, a standardized table was used to synthesize the literature. This table was derived from the screening documentation and the full-text papers included in the review. The table consists of all 62 papers and displays their research questions or research aim, research methods and their key findings. The extracted data from this table was also used during the final write-up of the Thesis' Chapter 4 (Conclusion). An excerpt of the data extraction table can be found below.

Table g: Excerpt of Data Extraction Table

Study ID	Title	Authors	Year	Venue	RQ / Research Aim	Research Method	Key Findings
DB16	Practitioners' Perceptions on the Adoption of Low Code Development Platforms.	Sebastian Käss; Susanne Strahringer; Markus Westner	2023	IEEE Access Vol. 11	RQ1: What are drivers and inhibitors of LCDP adoption? RQ2: What is the importance of these drivers and inhibitors for LCDP adoption?	Semi-structured qualitative interviews and a ranking-type Delphi study with 17 experts (consultants, line managers, and sales executives) to empirically identify, categorize, and rank drivers and inhibitors for LCDP adoption	Identified 12 drivers and 19 inhibitors for LCDP adoption, including six new drivers and six new inhibitors not previously covered in literature, and revealed that consensus among experts is highest for the most and least important drivers/inhibitors while relative importance for others is context-dependent (e.g., organization, adopted platform, expert background); findings provide empirical validation and a nuanced ranking to support decision-making for practitioners considering LCDP adoption.
AS41	Teaching Tip: Using No-Code AI to Teach Machine Learning in Higher Education	Leif Sundberg, Jonny Holmström	2024	Journal of Information Systems Education	RQ1: How can no-code AI be used to teach machine learning in non-technical educational programs?; RQ2: What are the benefits and challenges of using no-code AI in education?	Qualitative case study in a master's level "AI for Business" course at a Swedish university; students with diverse educational backgrounds participated in a group-based ML assignment using a no-code AI platform, and data was collected through observations, student feedback, written assignments, and evaluations; analysis followed thematic coding to extract learning outcomes, benefits, and challenges	No-code AI platforms facilitated meaningful hands-on ML learning for non-technical students, allowing focus on practical tasks like data collection, modeling, and evaluation; key benefits included lowering the coding barrier, enabling problem- and case-centered learning, and fostering data awareness and collaboration; challenges included the need for thoughtful case and platform selection, user management complexities, and ensuring all students could engage fully; the approach empowered non-coders to experiment with ML, but instructors should provide scaffolding for group work, realistic data tasks, and critical reflection on ML's applications and limitations.
GI39	Die vier Phasen von Citizen Development-Initiativen: Treiber, Herausforderungen und Handlungsempfehlungen	Binzer, Björn; Winkler, Till J.	2024	HMD Praxis der Wirtschaftsinformatik: Vol. 61, No. 5	(1) What drives companies to launch citizen development initiatives with low-code/no-code platforms? (2) How can the different phases of a citizen development initiative be characterized and differentiated?	Qualitative empirical study involving 21 semi-structured interviews with citizen development leads from 16 early-adopting companies (mainly large organizations across Germany, the UK, the US, etc.), analyzed with open and axial coding to develop a four-phase model of citizen development introduction and scaling. Interviews explored strategy, onboarding, training, platform choices, governance, and cultural/organizational context.	Companies start citizen development to boost digital skills, use existing IT tools, reduce shadow IT risks, modernize the workplace, and increase business unit agility. Implementation progresses through four phases: conception/planning, standardization, digital program integration, and making it a core competency. Major challenges include balancing governance, consolidating platforms, ensuring data quality/access, integrating training, and managing licensing. Practical recommendations involve careful platform choice, fostering IT and business partnerships, balanced governance, data/training focus, and supporting career paths. Additional challenges include cultural change, scaling initiatives, and adapting to new technologies like AI.

AD1	Low-code vs. the developer: An empirical study on the developer experience and efficiency of a no-code platform	Guthardt, Till and Kosiol, Jens and Hohlfeld, Oliver	2024	Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems	(1) How accepted are LCDPs in small and medium-sized enterprises (SMEs) and what barriers prevent adoption; (2) What level of experience is necessary to use no-code tools efficiently and do citizen developers differ from professional programmers in terms of task correctness or completion time when using a no-code platform	The authors conducted a two-part empirical study; first, they performed semi-structured interviews with representatives from six SMEs to gauge attitudes, perceived barriers, and usage of LCDPs; second, they carried out a controlled experiment with 10 students (five with software development experience, five without), asking them to complete standardized business application tasks using a custom no-code builder, and statistically compared their task success and processing times between groups;	SMEs expressed keen interest in LCDPs but reported concerns about training requirements, tool functionality, and flexibility; in the experimental study, participants with and without programming experience were able to achieve nearly identical high rates of task correctness and similar task completion times, with no statistically significant difference; technology affinity correlated with success more than prior coding experience; findings suggest that fears about excessive training needs for citizen developers may be overstated, at least for simple platforms and use cases; the authors caution that results may not generalize to large-scale or more complex scenarios but support the potential for LCDPs to democratize software development for simple business purposes within SMEs.
AD15	AI for Low-Code for AI	Rao, Nikitha and Tsay, Jason and Kate, Kiran and Hellendoorn, Vincent and Hirzel, Martin	2024	Proceedings of the 29th International Conference on Intelligent User Interfaces	RQ1: How do LowCoderNL (the language model-powered interface) and other features help participants discover previously-unknown operators for building ML pipelines?; RQ2: Are participants able to compose and then iteratively refine AI pipelines using LowCoder?; RQ3: What are the benefits and challenges of integrating language models with visual programming for low-code, especially for users with different AI expertise?	The study involved developing LowCoder, a tool combining block-based visual programming (LowCoderVP) and AI-powered natural language programming (LowCoderNL) for sklearn pipelines; 20 participants with varying AI expertise performed four ML pipeline creation tasks (with either NL or keyword-based search condition), allowing quantitative assessment (operator discovery, task completion, iteration) and qualitative feedback, coded via open and axial methods; language model components were rigorously benchmarked using both automatically extracted and real user data for various operator-invocation generation tasks	With LowCoderNL, participants discovered previously unknown operators in 75% of tasks (vs. 22.5% via web search or scrolling; NL condition enabled significantly higher discovery, especially for non-experts); 82.5% of participants completed pipeline-building tasks, with 72.5% further iterating, showing that both novices and non-novices could compose and refine ML pipelines, though non-novices performed more targeted iterations; qualitative analysis found that LowCoderNL was especially valuable when users "knew what, not how"—having a goal but not the operator's name; however, novices struggled when they lacked clear intent or understanding, suggesting that LowCoder benefits those with some AI domain knowledge, and template/contextual suggestion features would further help total beginners; overall, integrating visual and NLP interfaces advances discoverability, supports iterative exploration for both experts and non-experts, and points toward further "democratization" of AI programming, though clear ML design intent remains a barrier for total novices.

Supplementary Information

The following sections contain supplementary information that emerged from the creation of this review protocol.

Time-frame of previous SLRs

Title	First Author	Year Published	Timeframe Covered	Method used	Identifier
Challenges of Low-Code/No-Code Software Development: A Literature Review	Rokis	2022	2019 - 2021	Literature review	(Rokis & Kirikova, 2022)
Low-Code Development Platforms – A Literature Review	Prinz	2021	all until end of 2020	Literature review	(Prinz et al., 2021)
Democratizing Software Development: A Systematic Multivocal Literature Review and Research Agenda on Citizen Development	Binzer	2022	2014 - 2022	Multivocal Literature Review	(Binzer & Winkler, 2022)
Exploring Low-Code Development: A Comprehensive Literature Review	Rokis	2023	2019 - 2022	Literature review	(Alumni of Riga Technical University., 2023)
FROM USERS TO CREATORS: HARNESSING THE POWER OF CITIZEN DEVELOPERS	Hedlund	2023	n.a	Literature review & Case Study & Interviews	(Johansson & Hedlund, 2023)
Low-code/No-code Development : A systematic literature review	El Kamouchi	2023	Jan 2014 - Jul 2023	Literautre review	(El Kamouchi et al., 2023)
Low-Code/No-Code Platforms and Their Impact on Traditional Software Development: A Literature Review	Upadhyaya	2025	2020 - 2022	Literature review	(Upadhyaya, 2025)
Systematic Literature Review of Low-Code and Its Future Trends	Velásquez	2024	2018 - 2023	Literature review	(Velásquez et al., 2024)
What about the usability in low-code platforms? A systematic literature review	Pinho	2023	n.a.	Literature review	(Pinho et al., 2023)

Python Script for deriving the Top 50 terms across QGS Abstracts/Titles

```
pip install pandas scikit-learn openpyxl
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
import re
file_path = 'Selection Criteria Check_Template.xlsx'
df = pd.read_excel(file_path, sheet_name='QGS Details')
df = df.dropna(subset=['Title', 'Abstract'])
df['Combined'] = df['Title'].astype(str) + " " + df['Abstract'].astype(str)
df['Combined'] = df['Combined'].apply(lambda text: re.sub(r'[^\w\s]', '', text.lower()))
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['Combined'])
term_names = vectorizer.get_feature_names()
term_frequencies =X.sum(axis=0).A1
document_frequencies = (X > 0).sum(axis=0).A1
term_stats = pd.DataFrame({
    'term': term_names,
    'frequency': term_frequencies,
    'document_frequency': document_frequencies
})

top_terms = term_stats.sort_values(by='frequency', ascending=False).head(50)

from IPython.display import display
display(top_terms)

output = 'qgs_tf_top_50.xlsx'
top_terms.to_excel(output, index=False)
```

Top 37 terms from QGS Abstracts/Titles

Term	Frequency	Document Frequency	Mapped RQ(s)	Justification
development	40	11	RQ1, RQ2, RQ3	general term
lcdps	22	9	RQ1, RQ2, RQ3	core topic
study	18	9	RQ3	reports on research design/methodology
research	15	9	RQ3	inidcates academic study
platforms	14	9	RQ1, RQ2	Technical focus
lowcode	17	8	RQ1, RQ2, RQ3	core topic
adoption	31	7	RQ1	barriers and enablers
lcdp	30	7	RQ1, RQ2, RQ3	variant of LCDPs
citizen	16	6	RQ1	related to end-users, developer roles, democratization
application	11	6	RQ2	links to usage of LCDPs
digital	11	6	RQ1	digital transformation context
practical	7	6	RQ1	real-world use, implemenation
conducted	6	6	RQ3	empirical reporting
organizations	12	5	RQ1	context for adoption barriers/enablers
organizational	8	5	RQ1	complementary to organizations
practitioners	7	5	RQ1	relates to end-users/stakeholders
findings	6	5	RQ3	empirical results, synthesis
applications	6	5	RQ2	tied to use cases, implementation context
approach	6	5	RQ2, RQ3	method/development model
using	5	5	RQ2	tool usage, practical application
time	5	5	RQ2	eficiency/performance comparison
results	5	5	RQ3	outcome focus
platform	5	5	RQ1, RQ2	overlaps with "platforms"
understanding	8	4	RQ3	theoretical or conceptual frameworks
efficiency	7	4	RQ2	central metric for comparing development approaches
capabilities	7	4	RQ1, RQ2	technical or business context
paper	7	4	RQ3	general academic term
code	7	4	RQ2	low-code or full code
developers	7	4	RQ1, RQ2	stakeholder group
develop	6	4	RQ2	technical comparison
guidance	4	4	RQ1	organizational strategy, barriers/enablers
factors	9	3	RQ1	barriers/enabler context
business	6	3	RQ1	domain where adoption is evaluated
management	6	3	RQ1	organizational
model	5	3	RQ2	design implication in comparison
implementation	5	3	RQ2	practical use
important	4	3	RQ3	general term in reporting key insights