Current status of low-code/no-code platforms and demonstration of a practical example

Compact OSADL Online Lectures (COOL), December 2025

Open Source Automation Development Lab (OSADL) eG

A low-code/no-code platform goes Open Source

- What actually is low-code/no-code?
- How to select a suitable Open Source license for an existing proprietary software project?

What actually is low-code/no-code in comparison?

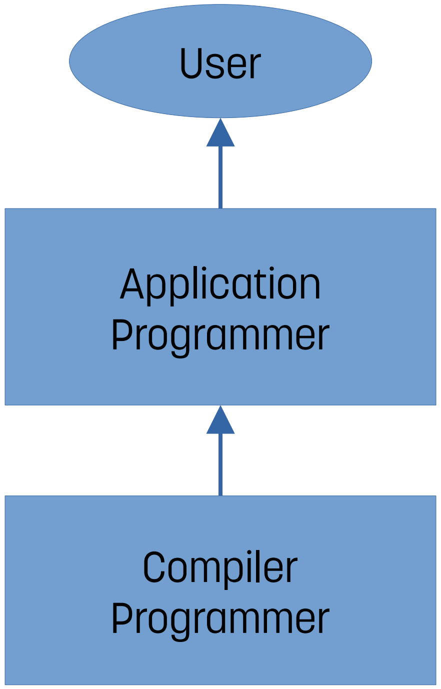
- Conventional programming
  - Write line-by-line code instructions to be compiled or interpreted and provide the program as a static executable with a predefined functionality.

What actually is low-code/no-code in comparison?

- Conventional programming
  - Write line-by-line code instructions to be compiled or interpreted and provide the program as a static executable with a predefined functionality.
- Low-code/no-code programming
  - Drag and drop prebuilt user-interface components, design a workflow and edit related data models.
  - Automate actions via workflow triggers and conditions.
  - Low-code: Optionally extend the code using conventional script languages.
  - Take care of hosting, upgrading, access control and logging via a platform-managed deployment system.

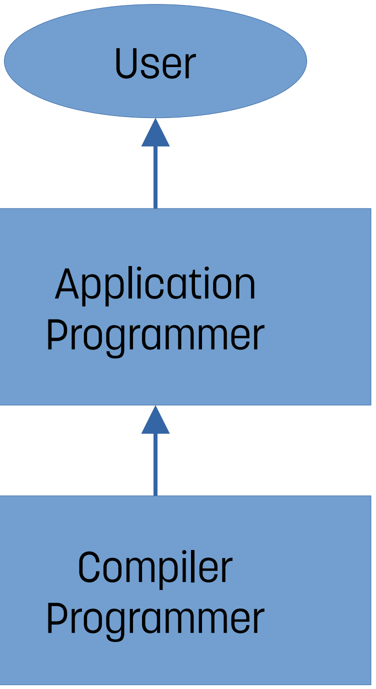
What actually is low-code/no-code in comparison?

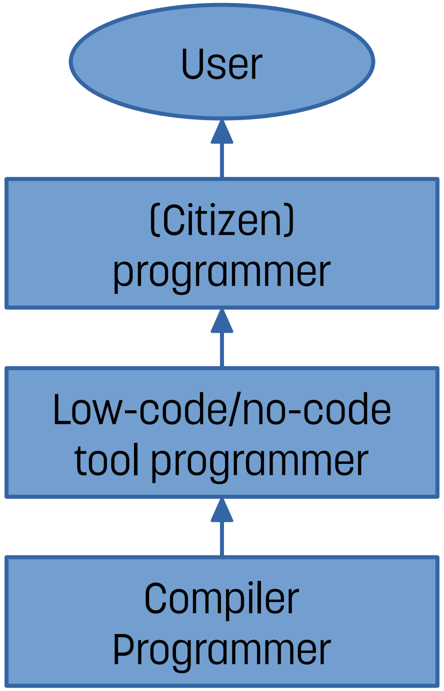
- Conventional programming



What actually is low-code/no-code in comparison?

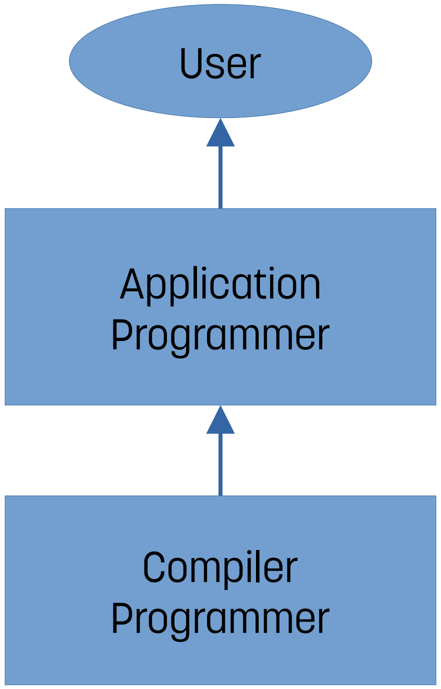
- Conventional programming
- Low-code/no-code programming



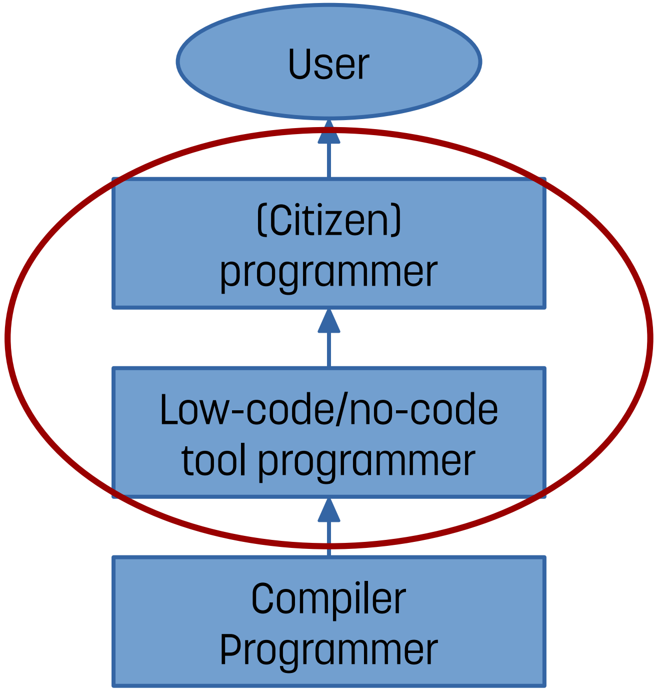


What actually is low-code/no-code in comparison?

- Conventional programming



- Low-code/no-code programming



Results of a meta study (individual results)

Dimension	LCAP-related practical experience	Impulse +1/0/-1 from source							Effect Normalized
		Luo et al. 2021	Alsaadi et al. 2021	Käss et al. 2023b/a	Al Alamin et al. 2021	Rafi et al. 2022	Martinez & Pfister 2023	Esposito 2021 (DZone)	
Time	Application development speed	+	+	+		+	+	+	*1,000*
Cost	Application development effort	+	+	+		+	+		*0.077*
	Building platform expertise	0	-	0	-				
	Provision, licensing, and operating costs	-		-			0		
	Governance-related expenses			0					
Quality	Agile integration of clients/requirement quality			+			+	+	*-0.115*
	Functional platform capabilities/customizing	0	0	-	-		-		
	Integration with third-party systems / data persistence	0	+	0	0				
	Performance / scalability	-	-				0		
	Security / Compliance		0	-		0	0		
	Software quality / error prevention					+		0	
	Maintainability / Testing / Debugging	0	0		-		0	-	
Flexibility	Enabling citizen development	0		+		+	+	+	*0.143*
	Dependence on/relief for IT service providers			+		+		+	
	Reusability of developed components							0	
	Vendor lock-in	-		-					
	Portability / Access to generated source code	-		-			-		
	Adaptability to changing circumstances								

Professionelle Softwareentwicklung mit Low Code optimieren – eine Fallstudie, Christoph Baumgarten, Rainer Endl, Silvan Stich
https://link.springer.com/article/10.1365/s40702-024-01095-y

COOL COMPACT OSADL ONLINE LECTURES

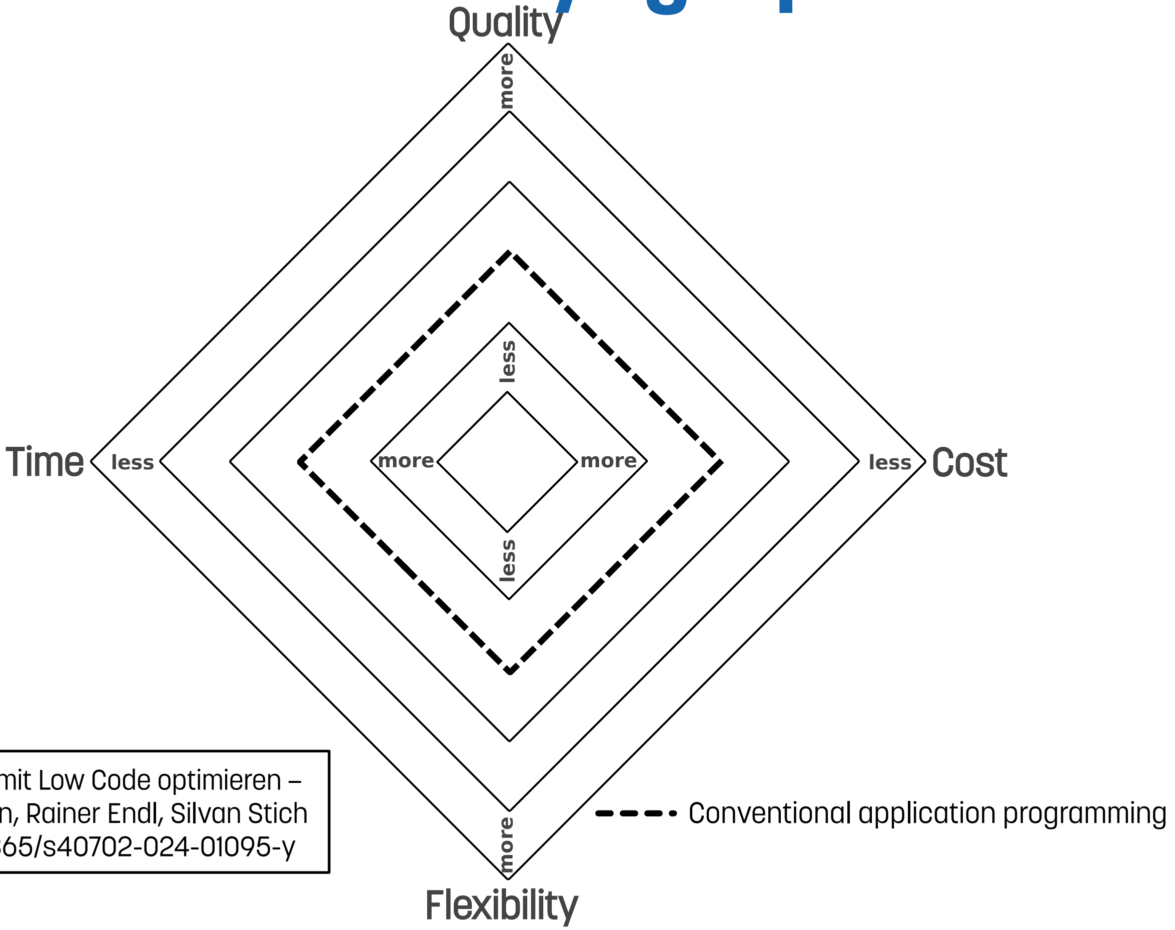
OSADL

Results of a meta study (individual results)

Dimension	LCAP-related practical experience	Impulse +1/0/-1 from source							Effect Normalized
		Luo et al. 2021	Alsaadi et al. 2021	Käss et al. 2023b/a	Al Alamin et al. 2021	Rafi et al. 2022	Martinez & Pfister 2023	Esposito 2021 (DZone)	
Time	Application development speed	+	+	+		+	+	+	*1,000*
	Application development effort	+	+	+		+	+		
Cost	Building platform expertise	0	-	0	-				*0.077*
	Provision, licensing, and operating costs	-		-			0		
	Governance-related expenses			0					
	Agile integration of clients/requirement quality			+			+	+	
	Functional platform capabilities/customizing	0	0	-	-		-		
	Integration with third-party systems / data persistence	0	+	0	0				
Quality	Performance / scalability						0		*-0.115*
	Security / Compliance		0	-		0	0		
	Software quality / error prevention					+		0	
	Maintainability / Testing / Debugging	0	0		-		0	-	
	Enabling citizen development	0		+		+	+	+	
	Dependence on/relief for IT service providers			+		+		+	
	Reusability of developed components							0	
Flexibility	Vendor lock-in	-		-					*0.143*
	Portability / Access to generated source code	-		-			-		
	Adaptability to changing circumstances								

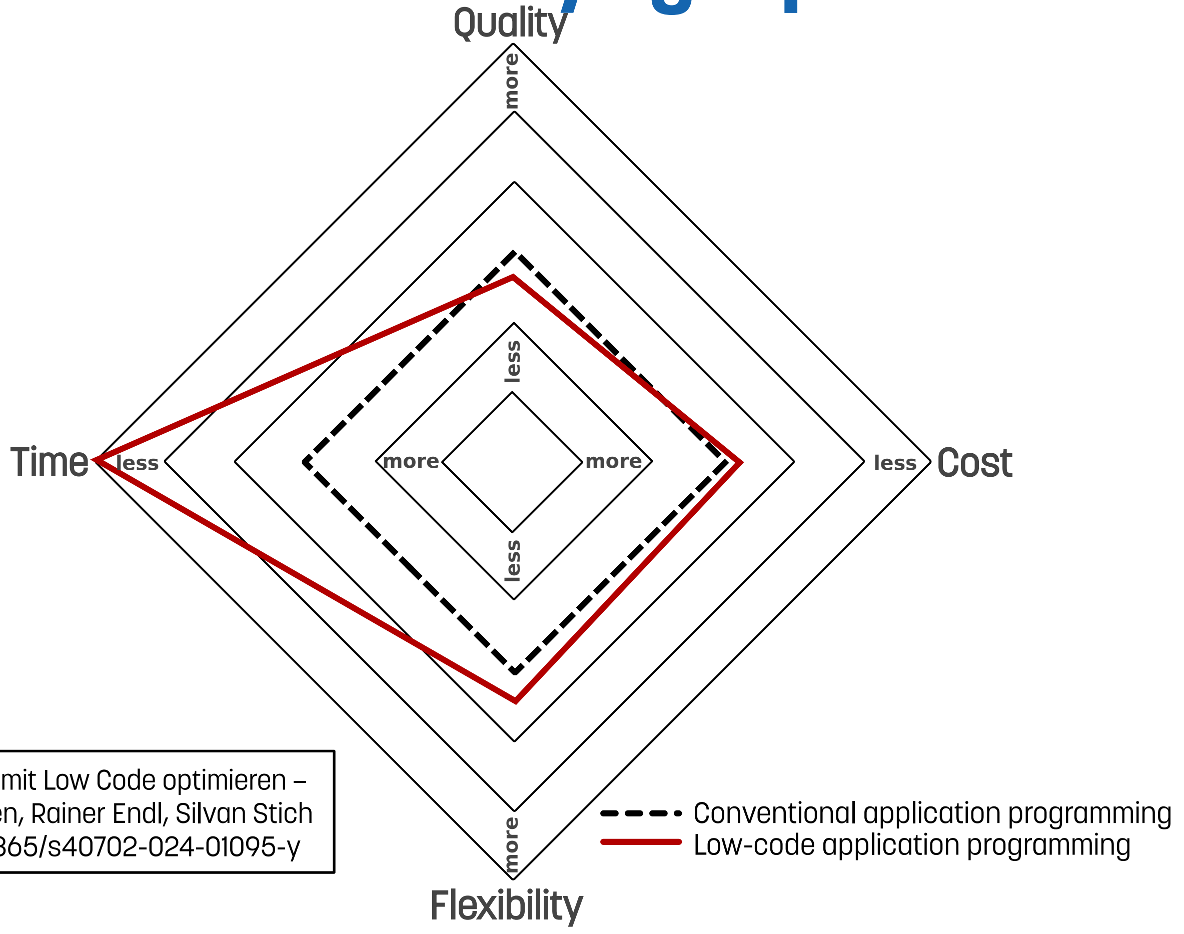
Professionelle Softwareentwicklung mit Low Code optimieren – eine Fallstudie, Christoph Baumgarten, Rainer Endl, Silvan Stich
https://link.springer.com/article/10.1365/s40702-024-01095-y

Results of a meta study (graphical overview)



Professionelle Softwareentwicklung mit Low Code optimieren – eine Fallstudie, Christoph Baumgarten, Rainer Endl, Silvan Stich
https://link.springer.com/article/10.1365/s40702-024-01095-y

Results of a meta study (graphical overview)



Professionelle Softwareentwicklung mit Low Code optimieren – eine Fallstudie, Christoph Baumgarten, Rainer Endl, Silvan Stich
https://link.springer.com/article/10.1365/s40702-024-01095-y

How to select a suitable Open Source license for an existing proprietary software project?

How to select a suitable Open Source license for an existing proprietary software project?

Criteria

- The type of software
  - Handling
  - Connectivity
- The owner's expectation
- The future users' expectation

The handling of the software

- Will the software only be used at the location to that it is downloaded?

or

- Is the software – maybe after modification – primarily intended to be copied and distributed as or along with a product?

The connectivity of the software

- Is the software a stand-alone program that can be licensed independently?

or

- Is the software a library that needs to be linked to other software in order to be deployed?

The owner’s expectation through Open Source

- Higher acceptance of the software due to low entry barriers
- Wider distribution of the software
- Higher speed of evolution

The future users' expectation through Open Source

- Better software at a lower price
- Easier installation of the software
- Possibility to contribute to the software
  - Bug fixing
  - Enhancements
  - New features
- Independence from the provider

Acceptance criteria by the community

License	License obligations fulfilled by provider	Connectivity	Handling	Acceptance
GPL type	**No**	Doesn't matter	Doesn't matter	**None**
GPL type	Yes	Library	Distributed	**None**
GPL type	Yes	Library	Used locally	Low
LGPL type	Yes	Library	Distributed	Less low
GPL type	Yes	Stand-alone	Doesn't matter	High
LGPL type	Yes	Library	Used locally	**High**
MPL-2.0/EPL-2.0	Yes	Doesn't matter	Doesn't matter	**Very high**
Permissive	Yes	Doesn't matter	Doesn't matter	**Highest**

Acceptance criteria by the community/owner

License	License obligations fulfilled by provider	Connectivity	Handling	Acceptance	Owner's preference
GPL type	**No**	Doesn't matter	Doesn't matter	**None**	**Very high**
GPL type	Yes	Library	Distributed	**None**	**Very high**
GPL type	Yes	Library	Used locally	Low	High
LGPL type	Yes	Library	Distributed	Less low	High
GPL type	Yes	Stand-alone	Doesn't matter	High	High
LGPL type	Yes	Library	Used locally	**High**	High
MPL-2.0/EPL-2.0	Yes	Doesn't matter	Doesn't matter	**Very high**	High
Permissive	Yes	Doesn't matter	Doesn't matter	**Highest**	**Very low**