

## Manifesto

---

Let's say you have the idea of creating a mobile application – for example, an Airbnb for cat-herding.

You will probably tell yourself that you don't have the means to finance the creation of such an app. Well, it actually doesn't matter, because you can go ahead anyway. Today, anyone can quickly carry out this type of project, without having to write a single line of code.

To build your app, you would typically begin by using a tool called Bubble, which allows inexperienced users to build mobile applications using a visual-based interface. You would then use Zeroqode to create a preconceived model similar to a simplified Airbnb. To connect your workflow tools (spreadsheet, calendar, etc.), perhaps Zapier would also be of some help. These are three examples of so-called “no-code” tools.

Although these “no-code” tools have existed for many years, they have only recently reached a stage of maturity and diversity that makes it possible to adopt them widely, opening up entirely new fields of practice.

We are >Contournement>, a group of individuals, training courses, and projects that are committed to making “no-code” tools – along with the right approaches that give them their full value – accessible to everyone.

This manifesto presents our vision and our convictions regarding this recent technological revolution, which is seldom talked about.

### **The power of “no-code” technologies**

Usually, to make computers do what you want them to do, you have to know how to speak the language that machines understand. Namely: computer code.

In a world fully immersed in the technology era, at a time when “software is eating the world,” people commonly have needs or ideas that require writing code. However, most of the time, this technical constraint quickly reduces their ambitions to nothing.

**But today, non-experts who are not versed in code or computer programming can embark on projects that would have required the work of several experienced engineers just 10 years ago. No-code technologies make it possible to:**

- **Create an advanced website or a sophisticated mobile app without writing a single line of code.**
- **Design tailor-made digital tools, as a means to better manage your work, automate your everyday tasks and help you focus on the essentials.**
- **Create chatbots, connected objects, video games, augmented reality, virtual reality, and more... without having to write code. Not so long ago, these possibilities were called science fiction.**

**These increasingly powerful tools are commonly referred to as “no-code technologies.”**

Some startups have already built the success of their products and services on these technologies. Dividend Finance, which raised \$330 million, or Plato, which has thousands of paying users, both rely on applications built on Bubble, a leading no-code tool.

**Our mission: free your projects from technical complexities**

The >Contournement> initiative was born from the recognition of the maturity of these “no-code” technologies. We have the following three objectives:

- **Provide access to “no-code” technologies to all those who want to create** a company or an application, equip an initiative or a team, automate a process, optimize personal productivity, etc., all according to their needs.
- **Emancipate individuals or structures from a vision too narrowly focused on technical concerns**, allowing them to focus on their objectives, and on the right methods to achieve them.
- **Take advantage of these high value-added technologies for social and solidarity purposes** (potentializing initiatives that promote social progress, empowering individuals that face obstacles in their career, etc.) in line with TharGo’s vision, a co-founding structure of >Contournement>.

**In other words, >Contournement> aims to empower people to master contemporary web development tools by facilitating access to the best no-code technologies.**

**>Contournement>: for whom, for what needs and for what causes?**

To do this, >Contournement> is convinced that each individual and each context can eventually benefit from a format adapted to its needs:

- **For entrepreneurs or organizations who want to build their own web or mobile application; for individuals or professionals who are keen to learn and enrich their skills:** we offer shorter training and coaching sessions, as well as workshops regularly, with training modules accessible online.
- **For structures wishing to internalize these tools and skills:**
  - internal block release training sessions for employees
  - socially-conscious training sessions designed for job seekers, allowing them to be recruitable at the end of their course
- **For any person wishing to become a no-code maker,** for any reason whatsoever.

**Also, >Contournement> aims, in the long term, to take advantage of these high value-added technologies for social and solidarity purposes** (potentializing initiatives that promote social progress, empowering individuals facing obstacles in their career, etc.) in line with TharGo's vision, a co-founding structure of >Contournement>.

**As we go along, we will try to bring all these stakeholders together within**

**"Contournement Fellows"** a network designed to bring together no-code makers and enthusiasts who are willing to collaborate. Our objectives include considering how to go further in the creation of value (particularly societal, non-market, and educational value) that can result from the mastery of no-code technologies.

**Why ">Contournement>"?**

We call ourselves >Contournement> ("contournement" translates as "bypass" or "circumvention" in English) because we consider that using more accessible technologies can make it possible to circumvent many obstacles, to "bypass" certain constraints.

1/ No-code is an excellent way to do something yourself that you would not otherwise be able to do (due to lack of finances, skills, etc.)

2/ No-code is an excellent way to take the shortest path to what you want to achieve, reducing the time between the idea and the realization.

3/ No-code is an excellent way to minimize risks: you start by creating a prototype that works, you test your idea, and if it works and is used, then you invest in a coded application.

4/ No-code is an excellent way to keep control over your project's implementation, without having to depend on a professional for evolutions and modifications.

5/ No-code is an excellent way to focus on the real added value brought to the user, as well as on the right methods and approaches that make it happen (agility, lean startup approach, design thinking, etc.)

**Bypassing is the kind of approach that, combined with the method of diverting a use or a tool, resonates with the hacker mindset, focused on solving problems, building things, and often seeking the simplest solution.**

**On the other hand, we believe that if no-code is often the fastest way to achieve your objectives, it does not necessarily mean that it is the best.**

Indeed, part of our team comes from the world of web development. We are perfectly aware of the limitations of no-code in several instances, including a lack of depth of functionality and architecture advancement, compared to what experienced developers can accomplish.

And this holds true, even though some applications have shown how robust and durable no-code solutions can be, making it possible to carry out certain products and projects to an advanced stage (see the examples of Finance Dividend and Plato).

**No-code = Anti-code? The answer is no.**

Some may ask: "But if you advocate no-code as the ultimate solution, does that mean you recommend avoiding learning code? Or quit working with web developers and programmers? Or maybe you want them gone?"

To these three allegations, our answer is clear:  
"no."

### **Code VS programming**

Above all, it is essential to distinguish between the concepts of "code" and "programming," which are widely confused among the general public.

When you write a message in Morse code, you are basically coding. You are translating a message into a different form.

When you program your washing machine to wash for one-hour at 30°C, followed by 30 minutes of drying, you are basically programming. You are programming the device to execute a sequence of commands.

When you write computer code to prevent payment on your site if the shopping cart does not exceed 5€, you are programming a feature of your website by writing computer code. In other words, you are programming, through coding specific commands.

This is why we can consider that when you use a no-code tool to create an application, you are programming it, without using any code. You are simply using a visual interface instead of a computer language.

### **No-code VS code**

Does the increasing maturity of no-code technologies mean that we must stop working with programmers who code? The answer is No.

No-code cannot replace the work of experienced professionals, especially regarding:

- Advanced features and architectures,
- highly customized interface designs,
- and more generally, the know-how that falls beyond the scope of code (user experience, strategic advice, etc.)

This is why we essentially advocate the adoption of no-code solutions in frameworks guided by a lean startup approach (quickly experimenting with your idea by prototyping, testing, and improving) and frugal (doing a lot with little, achieving your objectives with the means at hand).

Moreover, depending on the need and the context, we see it as a logical continuation that tools prototyped with no-code are then redesigned by developers, once the concept has been tested and validated.

### **Does that mean we have to stop learning code?**

The >Contournement> team mostly comprises people who have been pioneers in web development training – the term ‘web development’ should be preferred to ‘code,’ in our opinion.

>Contournement> believes that there is a great deal of value in learning to write “code”, provided it forms part of a broader perspective of eventually mastering solid programming skills and good practices, combined with other key knowledge in the field (design thinking, architecture, strategy, etc.).

On the other hand, our vision is that no-code forms part of a broader set of technological



advances that invites us to question the viability and sustainability of code-writing skills that are limited to the basics.

Other technologies that threaten low-skilled code and programming skills include simplifying tools (such as front-end frameworks) and artificial intelligence.

### **Why learn no-code technologies?**

Concerning no-code learning for professionalization and occupational integration purposes, the current offer on the labor market is not mature enough in our opinion.

However, learning no-code can be an excellent way to understand how computer programming works. If you build an application with no-code tools, the concepts and logic are the same as if you chose to build it by writing code. The only difference is the mode of action: you are using a visual interface.

### **A few words of history: before no-code came along, technology was already paving the way for increased user accessibility**

During the decade of the 2000s, if you wanted to create web/mobile sites and applications, you almost had no choice but to use code-based technologies – that is, you had to write lines of HTML, CSS, JAVA, PHP or other programming languages to describe what the computer should display or execute.

To master these technologies, you had to be a trained professional (or a passionate and persevering self-taught person), with years of practice, and experience working in a team.

**Around 2010, technologies that were more accessible to non-experts have begun to mature, such as:**

- **Tools to create simple websites**, offering the possibility to manipulate only visuals, without the need for coding. Examples include WordPress or Squarespace, commonly known as CMS (content management systems.)
- **Tools to create complex sites and applications**, making it possible to program more quickly through the use of:
  - More expressive programming languages for the human being – Ruby language, for example.
  - And/or an existing structure or blocks. Frameworks such as Bootstrap, Rails, Angular, or Meteor, for example, have enabled beginners to learn more quickly how to create websites and applications, through making these skills more accessible.

However, in both cases, it was still necessary for users to master code, computer programming in the true sense of the word, and web development tools.

**“No-code” technologies gradually appeared between 2012 (approximately) and 2019. They made it possible to develop advanced websites and complex web/mobile applications without having to write a single line of code. The only requirement was to manipulate visual elements.**

And it is mainly after 2015 that applications created on app-builders (these tools made to build apps), such as Bubble.is, began to be available online, and rapidly started to host large volumes of users.

From 2018 onwards, a new generation of tools such as Coda or Airtable gradually appeared, entirely rethinking the concepts of databases and documents.

## **Conclusion**

This is our vision of no-code.

We are convinced that these technologies have the potential to unleash ideas and energies on a broad scale.

We see the potential of equalizing opportunities regarding the use of digital tools: you don't have to be technically trained to be legitimate in your project and vision.

No-code also offers the potential of widening access to these means of action, for individuals or initiatives that have hitherto been excluded from these high value-added levers.

We are convinced that self-discipline should be that everyone should have with their uses, and of the critical distance that should be systematically applied, regarding the impact that new information and communication technologies have on our societies on our world.

No-code technology is now a new component of our world, and its impact is likely to be extremely significant.

As with any other tool, these impacts can take extremely diverse forms, from the most beneficial to the most harmful.

And those who master these technologies will be able to play an active role in these impacts.

Because there is no such thing as technological fatalism. Technologies will be what we make of them.

The >Contournement> team: Erwan Kezzar & Alexis Kovalenko