



# Research Behind: A Founders' Guide to No Code

**Annais J. Paetsch**

Advisor: Morten Hansen

*annais\_paetsch@berkeley.edu*

*University of California, Berkeley School of Information*

# Table of Content

## **I. Introduction**

## **II. Analyses: Data Collection, Methods & Findings**

### **1. Analysis 1: Quantitative No Code Startup Analysis**

a. Data Collection and Method of Quantitative No Code Startup Analysis

b. Findings of Quantitative No Code Startup Analysis

### **2. Analysis 2: User Personas and Needs Analysis**

a. Data Collection and Methods for User Personas and Needs Analysis

b. Findings of User Personas and Needs Analysis

### **3. Analysis 3: A Founder's Guide to No Code**

a. Data Collection and Methods for "Founder's Guide to No Code"

b. Findings of "Founder's Guide to No Code"

## **III. Discussion**

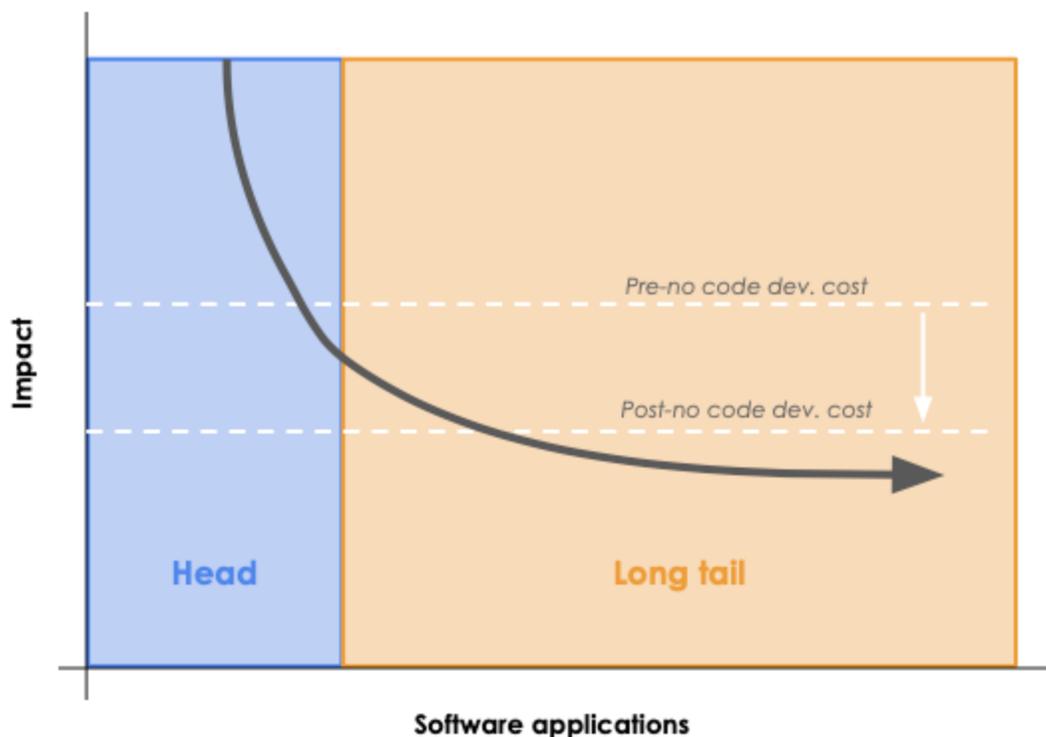
## **IV. Appendix**

## **V. Sources**

# I. Introduction

## Motivation

The No Code industry has seen a high pace of maturing in the last 3-4 years. In early 2021, Gartner projected a 23% increase for the global market and predicts 65% of overall application development activity to happen in low-code by 2024<sup>1</sup>. The capabilities of low code have recently successfully advanced to cover the customization of business workflows. With this they have enhanced customer and employee experience while achieving boosted efficiency of commercial and operational teams<sup>2</sup>. Further, no code is hypothesized to enable the build of long-tail software applications<sup>3</sup> as no code increases the supply of developers and has the potential to reduce the cost of development.



The number of software developers (full-time and part-time/hobbyist) was estimated to be 26.2 million or about 0.03% of the global population in 2020. This number is expected to grow to 43.2 million in 2025 mainly driven by the growth of part-time

developers through the enablement of low code and no code development tools<sup>4</sup>. Imagining how the global growth in literacy over the last centuries has transformed our society, leaves room to imagine how code literacy may shape society in the decades to come and what role low code and no code will play in the development of societies, innovation and global participation in the software ecosystem.

## Objective of project

Coming back to the present, low code and no code are already affecting how development is happening today. Amongst other spaces, no code is changing entrepreneurship by lowering the barrier to entry to product development. The motivation of this project at the highest-level is to understand the interplay between no code/low code and entrepreneurship.

The ultimate goal of this project is to help founders make a decision on building their product in no code or raw code. Thus, the ultimate artifact I wanted to create was “A Founder’s Guide to No Code”. To create this guide and lay out the interplay of entrepreneurship and no code/low code at large, I broke down this project into three questions and objectives:

### 1. Question: What differentiates no code entrepreneurship?

Objective: Understand what characterizes the no code entrepreneurship space in terms of industry, product types and founders

### 2. Question: For which founders is no code relevant?

Objective: Define what founder personas exist, understand their needs and motivations

### 3. Question: Should founders build in no code — how can they decide?

Objective: Propose a best practice evaluation approach and identify relevant considerations

These three questions, each building on the next, guided my analyses towards the ultimate objective.

While the guide is oriented towards founders considering to build their product in no code, the findings of this project overall may be relevant and interesting to a broader audience, namely investors in the no code industry, no code enthusiasts, “makers”, no code developers and more.

## What is no code? What is low code?

Before diving into analysis, it is important to lay out the distinction between no code and low code. Retool, a low code tool, defines low code in the following way<sup>5</sup>:

*“Low code is the development of software via a visual drag-and-drop interface. The logic is simple and the UI is intuitive; ideally, it offers possibilities for customizing the majority of components with code, but the idea is that most of the work can be done without spending time on writing custom code.”*

According to Retool, it has three components: A visual integrated development environment (IDE), connectors (to connect across a variety of backend-services, APIs and databases) and an application lifecycle manager (e.g. tools for debugging, deployment, maintenance).

The difference to no code is defined as having no hand-coding at all. It is purely using a visual workflow builder and IDE. The core difference that this makes is that low code allows for greater flexibility but also requires at least basic literacy of code, making the target user slightly distinct in programming experience. Further, this extra “requirement” of basic code literacy is what allows for greater customization of low code applications<sup>6</sup>.

While the definitions seem very clear in theory, in practice the lines between no code, low code and raw code can become a lot more blurry. For the purpose of this paper the difference of low code and no code is not critical to distinguish and we can treat both as one category with a spectrum. Thus, I will refer to low code and no code together as “no code” throughout.

## II. Analyses: Data Collection, Methods & Findings

As mentioned, the ultimate goal was to write “A Founder’s Guide to No Code” to help early-stage teams decide whether they should build their product in no code or raw code. Following the three questions laid out above, I aimed to create the following three artifacts to achieve the respective research objectives:

1. Objective: Understand what characterizes the no code entrepreneurship space in terms of industry, product types and founders

**Artifact: Quantitative analysis of 100 no code startups**

2. Objective: Define what founder user exist, understand their needs and motivations

**Artifact: Identified user personas and respective needs analysis**

3. Objective: Propose a best practice evaluation approach and identify relevant considerations

**Artifact: A Founder’s Guide to No Code**

Following, I will describe the data collection and method applied to create the respective artifact for each stage.

### 1. Analysis 1: Quantitative No Code Startup Analysis

#### 1a. Data Collection and Method of Quantitative No Code Startup Analysis

##### Dataset collection

The dataset consists of a database of 350+ no code startups. The database is hosted and managed by No Code Founders, which is an online community for founders who are building products without code. The community provides interviews with founders, an overview of no code tools, tutorials, discounts on no code tools, events and the

database of no code startups. A sample of 100 no code startups were the basis of this analysis.

Another dataset that was used for the later stages of the analysis was the database of YCombinator startups. YCombinator, is one of the most well-known startup accelerators that has launched more than 3,000 companies<sup>7</sup>. This database was used as a benchmark for industry breakups in the startup industry (more on this later).

### **Methods: Background research & dataset coding**

The analysis of no code startups involved a mixed-method approach. First, each startup was researched by reading the description written by founders on the No Code Founders database, reviewing the startup website and product. Based on this research each startup and its founder was coded by:

- Industry (multi-select label): Industry labels were selected from industry labels in YCombinator database to make it comparable; importantly, for B2B startups there were high-level labels of “B2B Products” and “B2B Services” and detailed-labels of “B2B Products”, e.g. “B2B Customer support”, etc. Thus, every detailed B2B product label was accompanied by a “B2B Product label” (to enable high-level or detailed cuts)
- B2B vs. B2C (single-select label): Choice between “B2B” for products catering to businesses, “B2C” for consumer facing products or “both” for products catering to both (e.g. marketplaces)
- Product type/features (multi-select label): Product type/feature labels were identified core features of the product (refined in multiple iterations, bottom up coding); for B2B startups, again B2B products had this high level or detail-level labels (analogous to industry)
- No code tools (multi-select label): No code tools used in product tech stack as indicated by founders on No Code Founder website (logos were mapped to firms)
- Stage (single-select label): Choice between “Waitlist/early sign-up”, “Beta: Product live”, “Product live”, “Website down” to denote stage of product as can be seen on startup website (e.g. waitlist signup, product live, possibility to download, etc.)
- Co-founder with CS background (single-select label): Choice between “yes” for founder who either have a CS degree, worked as developer or have technical project on personal website, “unclear / no” if engineering degree but not clearly CS

experience, “no” for founders without stated CS experience, “n/a” for founders that were not found on Twitter, LinkedIn or elsewhere

Lastly, some “startups” in the database were no code agencies displaying themselves in the database to catch interest. Agencies who do not have a product and only offer services to build no code products were excluded from the analysis. The final dataset no code startups coded with the above categories can be found [here](#).

**No Code: Taxonomy**

Table + Add view Filter Sort Q ... New

No Code tools + Add filter

Startup	Description	Industry	B2B vs C	Product type	No Code tools	Stage	Legitimacy	Co-founder
Micromega	Online community to discuss humanity & technology	Social	B2C	Online community	Bubble	Waitlist / early		n/a
Yearner	Insights for investment performance	Finance	B2C	Finance (personal)	UNIDENTIFIED: BLUE CIRCLE W	Beta: Product		n/a
Spayce	Payment management for freelancers	B2B Finance	B2B	B2B Finance	Bubble	Beta: Product	NCF tweet wi	n/a
GrowthMentor	Find a growth mentor for 1-on-1 conversations	Education Talent: Job & Cai	B2C	Matching platform	WordPress Typeform Google Sheets Notion Calendly Figma Airtable Zapier Unbounce Google Analytics Slack	Product live		No
Conversio	Customer support via Whatsapp & text	B2B Customer su B2B Products	B2B	B2B Customer support B2B Product	Bubble	Waitlist / early		No
Digist It!	Personal news summarized as email digest	Media / content	B2C	Newsletter / content	Integromat Carrd	Waitlist / early	NCF tweet wi	Yes
Plannly	Project management platform	B2B Project man B2B Products	B2B	B2B Project Management B2B Product	Bubble	Waitlist / early	NCF tweet wi	No
Fridayy	Shopping via text	eCommerce	B2C	Online shop	Bubble	Product live		n/a
scrollrack	Chatbot for Magic: The Gathering	Gaming	B2C	Productivity	Airtable Google Sheets	Beta: Product		n/a ?

COUNT 100

## Methods: Quantitative analysis

After coding our dataset, this data was aggregated and split by each dimension coded: Industry, B2B vs. B2C, product type, stage and co-founder with CS background.

For the industry analysis, the breakup was compared to YCombinator’s industry breakup to understand what characterizes No Code startups. YCombinator is obviously not a fully representative sample of the startup industry at large. Being one of the most successful accelerators with years of experience, it captures a large number of startups and specifically over-samples on startups with high success odds.

## 1b. Findings of Quantitative No Code Startup Analysis

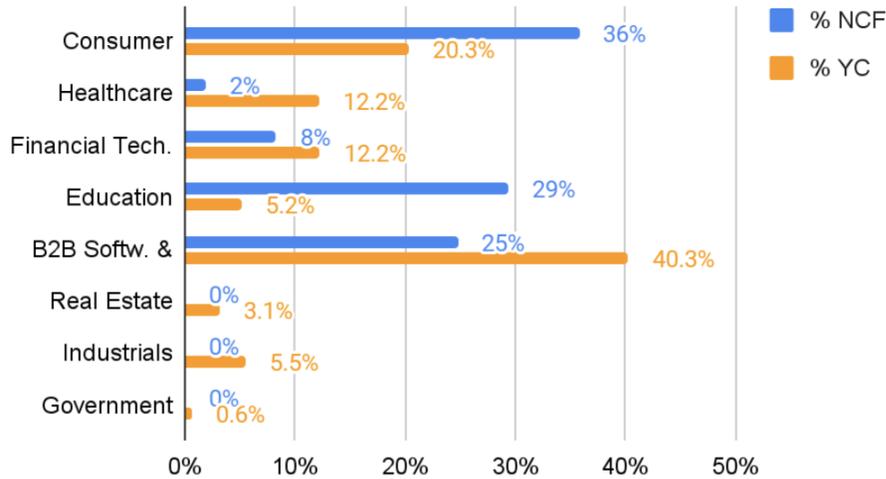
### Education and consumer startups make up biggest share in no code startups

Firstly, when comparing industry breakup of no code startups to YCombinator startups, I found that the industries that were strongly underrepresented amongst no code startups were *B2B software and services* startups and *healthcare* startups with a delta of 15% and 10% respectively compared to YC's portfolio. This delta is to be expected and in line with expectations. YCombinator is said to have a strong focus and, thereby, overrepresentation of *B2B software and services* startups. Further, *healthcare* startups are expected to be underrepresented in the no code sample as healthcare innovation expands beyond software (which is what no code enables). Biotech startups are examples of this.

*Education* and *consumer* on the other hand are strongly overrepresented compared to YC's portfolio by a 24% and 15% delta respectively. These are also the two industries most strongly represented in the no code sample overall with a 29% share for education and a 36% share for consumer. A possible explanation for the *consumer* and *education* overrepresentation is that consumer products require less professional experience (e.g. don't require knowledge over pain points that are specific to certain companies) and can be ideated from personal experience. Lowering the barrier to build a product could enable product ideas to be built that come from personal experience. It was also noticeable that the *consumer* products tended to be very niche products (e.g. gaming chatbot for "Magic the Gathering"). This is also in line with the hypothesis in the introduction: No code lowers the entry to build for the long-tail software applications. Lastly, a possible explanation for the overrepresentation of *education* is that many of no code's successful use cases are particularly well-suited for education startups (as can be seen when looking at the most common product features below).

Finally, the industries with small delta in representation are government (-1% vs. YC), real estate (-3% vs. YC), financial technology (-4% vs. YC), industrials (-6% vs. YC) and unspecified (-1% vs. YC). Generally none of these industries make up more than 5% each of YC's portfolio or in the no code sample speaking for low disruption in these industries overall. The only exception to this is financial technology which makes up 12% in YC's portfolio.

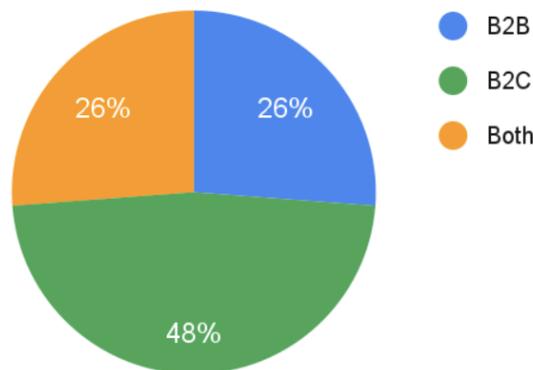
### Industry Breakup of NCF vs. YC



### B2C startups make up 48% of no code startups

As expected following our insights from the industry analysis, *B2C* startups make up a majority of no code startups in terms of customer relationship. *B2B* and *both* (e.g. marketplace involving business and consumers) make up 26% each.

### NCF startups by customer relat.

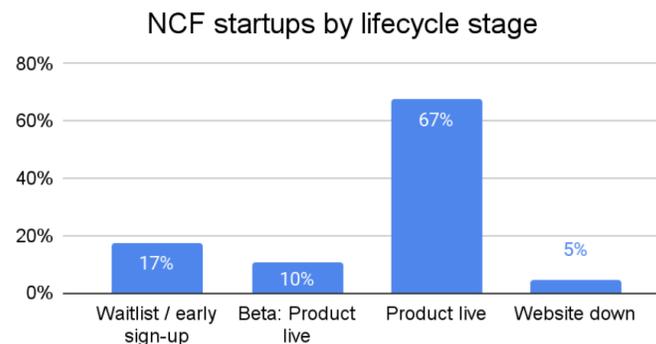


### 2/3 of no code products live, 1/4 of products in idea validation

17% of products with *waitlist/early-sign-up* access and 10% with a *beta product* roughly equate to 1/4 of the products in this sample being in the idea validation phase. 2/3 of the

products appeared to be *live products*. However, within the live category it is hard to differentiate products that looked live but effectively acted as beta products, too. Thus, it is hard to draw conclusions on what no code facilitates more strongly: Idea validation vs. product build.

Notably, only 5% of startups had websites which were *down* and invalid. However, it is likely that a few products which still appear live are also invalidated products. Thus, again it is hard to draw specific conclusions here.



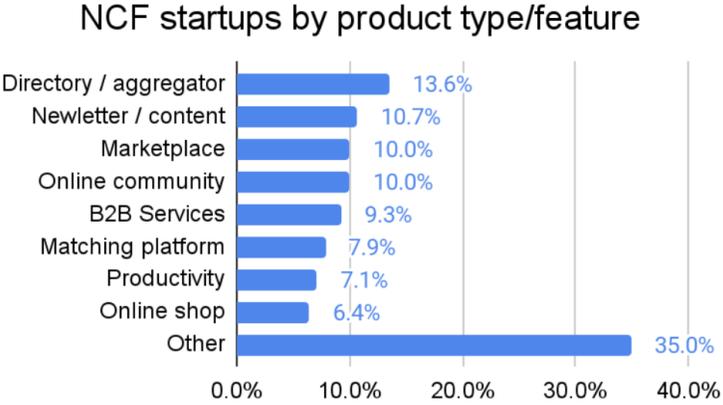
### Five dominant, well-defined product features observed in no code startups

When analyzing the sample of 100 no code startups these product features were represented most frequently:

- Directories/aggregator features
- Newsletters/content/blog post features
- Marketplace features
- Online community features
- Online shop features

Other “feature classes” with high representation were more heterogenous within a class. For example, productivity features appeared in 7.1% of no code startups but were implemented in very heterogenous ways across products. Other diverse but big “feature classes” were B2B service and matching platform. Other features, aggregating to 35% of representation in the graph below, independently were represented in less than 5% of startups. Lastly, it is noticeable that there is not one core feature that dominates across

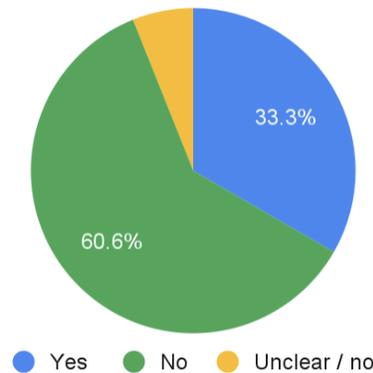
startups but rather a broader selection of features. This speaks for the path of use-case oriented innovation occurring within the no code industry.



**Two thirds of no code founders without CS background**

About two thirds of no code startups has at least one CS background founder vs. none, implying they have no computer science degree, software engineering projects or programming projects posted online. No code particularly lowers the barrier to building products for people without deep computer science experience. It is not necessarily an intuitive result as developers also use no code tools despite being able to code (e.g. early evangelists). However, it speaks for the adoption progress the no code industry has seen beyond the developer world. Further, another difference to be noted is that most no code founders of startups in my sample were solo founders, when the “standard” preferred model is not being solo. A study by MIT found that in a kickstarter sample 28% were solo founders vs. 31% two-people teams vs. 41% three or more-people teams<sup>8</sup>.

NCF startups by founder with CS background



## 2. Analysis 2: User Personas and Needs Analysis

### 2a. Data Collection and Methods for User Personas and Needs Analysis

#### Dataset collection

The data used to create the user personas and conduct a needs analysis was collected from 10 user interviews. Specifically, seven interviews with no code founders, one CTO and two no code developers. Of these interviewees five were recruited through cold outreaches on LinkedIn and Twitter, three through warm introductions and two from my personal network. Geographically, they were very dispersed with two interviewees based in Germany, one in Hong Kong, two in Greece, three in the United States, one in Tanzania and one in Mexico. Notably, I did not aim to reach a global audience, but rather this organically happened and the referrals for further interviews were all cross-border introductions, too, This may perhaps be a characteristic of the no code community, too.

#### Methods: User interviews, qualitative coding of interviewees & creation of personas

The process of this analysis involved multiple stages: Creation & refinement of an interview guide, qualitative coding of interviewees and creation of the user persona

artifacts.

To understand what users may most benefit from “A Founder’s Guide to No Code” (artifact #3) and what the guide should cover, the interview guide focused on covering personal introduction to no code, process on no code go/no-go decision, relevant considerations, unforeseen changes and limitations.

After completion of the interviews, I coded interviewees by “goal of build”, “side project vs. main project”, “role” (in startup), “Attitude towards no code exploration/decision”, “Informal technical knowledge”, “Formal CS knowledge”, “Stage of funding”, “# of employees” to characterize interviewees but also the maturity of the startup. It became apparent that the most characteristic difference in approach was driven by the objective of the build. More specifically, whether they were building this as a side business or main business and for what time frame they were making a decision (prototype vs. next 1-2 years). Thus, the three no code decision maker personas that became apparent were:

1. Side project founder
2. Prototype-stage founder
3. Core product-build founder

## Interviewees ...

Goal of build	Sideprojec...	Role	Explora...	Informal...	Formal ...	Stage	# of emplo...
Sideproject	Sideproject	Founder	Explorative	No	No	No funding yet/	Solo/founder onl
Sideproject	Sideproject	Founder	Unclear	Unclear	No	Funding unknowr	Not relevant
Prototype	In-between	Founder Head of Bus I	Explorative	In-between	In-between	No funding yet/	1-10 employees
Prototype Build lasting product	Main project	Founder	Explorative	Yes	No	Seed	1-10 employees
Prototype	Main project	Founder	Explorative	No	No	Pre-Seed	1-10 employees
Prototype Build lasting product	Main project	Founder	Analytical	Yes	No	No funding yet/	Solo/founder onl
Build lasting product	Main project	Founder	Analytical	In-between	No	No funding yet/	1-10 employees
Build lasting product	Main project	CTO	Analytical	Yes	Yes	Funding unknowr	11-50 employee
Other	No code develo	No code deve	Unclear	Unclear	No	Not relevant	Not relevant
Other	No code develo	No code deve	Analytical	Yes	No	Not relevant	Not relevant

Taking these three personas, I next collected key insights from interviews for each persona to create the personas:

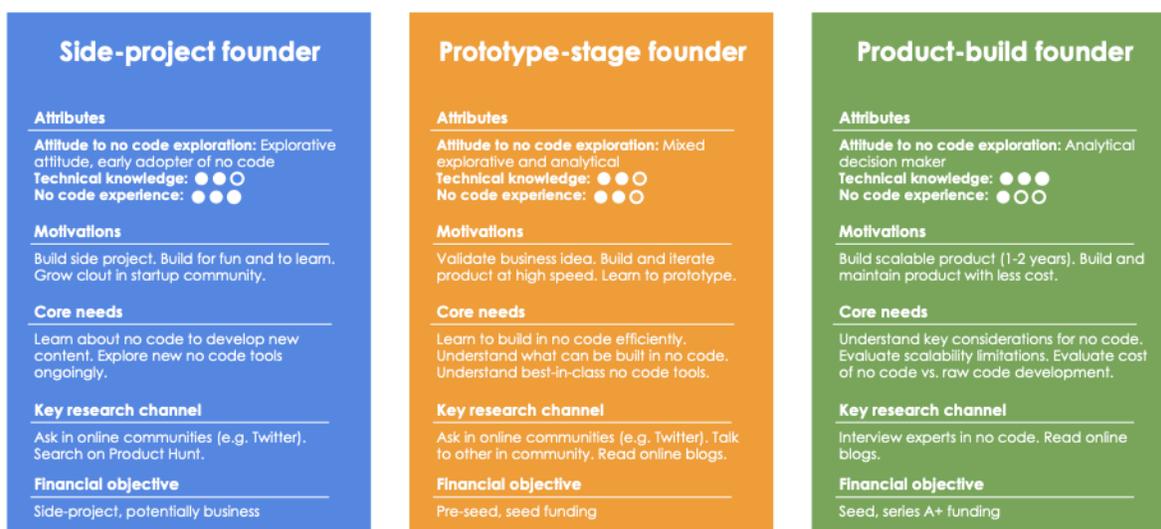
- Founder attributes: Attitude to no code exploration, technical knowledge (formal or informal), no code experience at decision time
- Motivations for using no code
- Core no code related needs
- No code learning channels
- Entrepreneurial objective

## 2b. Findings of User Personas and Needs Analysis

### Three target personas differentiated by no code build objective

As mentioned above the distinct approaches to no code build decisions - and consequently the three personas that emerged - from the interviews were (double click image to expand):

### Interviews uncovered three decision maker personas



### Founder personas differentiated by attitude, technical knowledge & no code experience

Interestingly, the no code experience at the time of product build was almost inverse to the size of the entrepreneurial ambition. Or put differently, the founders making the most long-term no code decisions were not necessarily the ones who had the most experience with no code. The side project builders tended to have more experience with no code as they had commonly built several no code projects before. Further, no code experience at time of no code decision making did not correlate to computer science experience either.

Main project founders (“prototype-stage founders” and “product-build founders”) tended to be more analytical about the decisions (e.g. research and structure decision) compared to side-project founders, who were often early no code adopters and more explorative (e.g. build and learn). When they decided to build a business later, they then

already tended to have a lot of experience. Side-project founders also cared to build for the sake of learning and engaging more broadly in the no code community (e.g. on Twitter). Main project founders - especially core product-build founders - were more concerned about “doing the right thing” and getting an answer as quickly as possible, as they were concerned with putting their business at risk with the wrong decision.

### **Prototype-stage and product-build founders can benefit most from a guide**

The main project founders - prototype-stage founders and core product-build founders - make up for the perfect target persona of a no code guide as the gravity of their decision is largest, they tend to have the least no code knowledge upfront and other technical knowledge (e.g. computer science degree or developer experience) only helps understand the basics of no code.

Having identified the persona that can most benefit from the guide, I consequently also focus my guide on the motivations and needs of these two groups covering the following topics:

- Learn to build in no code efficiently/quickly
- Understand what no code can or cannot do (e.g. what common no code products)
- Understand key considerations and risks to evaluate no code
- Evaluate scalability limitations
- Evaluate development and running cost of raw code vs. no code

Lastly, these two target personas commonly read “Medium-style” or “LinkedIn-style” blogs written by people who have faced similar situations and can share experiences. They especially seek actionable knowledge for their decision. Thus, my guide aims at a similar writing style and aims to provide actionable examples and heuristics from experience.

# 3. Analysis 3: A Founder's Guide to No Code

## 3a. Data Collection and Methods for "Founder's Guide to No Code"

### Data collection

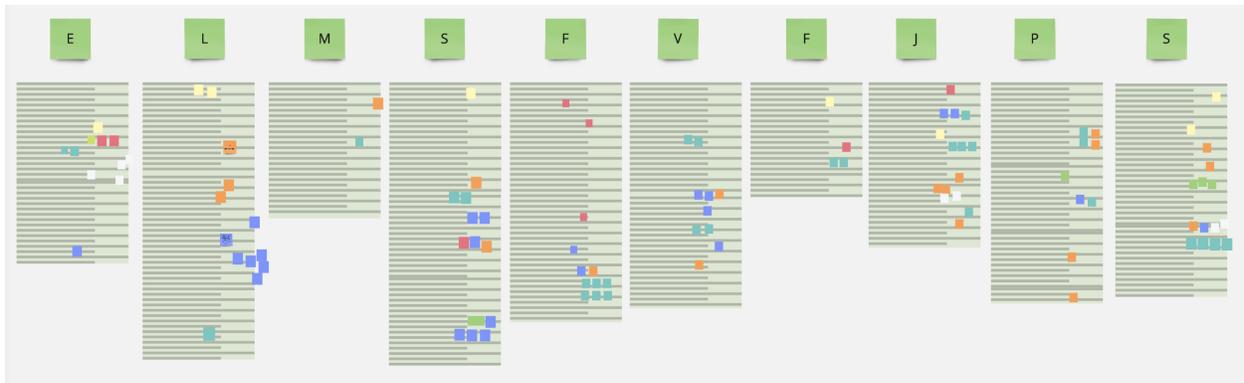
The data collection for this last part stems from the same user interviews as for the "User Personas and Needs Analysis (artifact #2). Please refer to [section 2a](#) for details on the collection process.

### Methods: User interviews, qualitative analysis and guide writing

The process of this analysis involved multiple stages: Creation & refinement of an interview guide, qualitative coding of interviews, synthesis of insights and outlining & writing the guide. Lastly, the guide was sent to founders who are the target persona facing a no code build decision to gather feedback, which was incorporated in the guide.

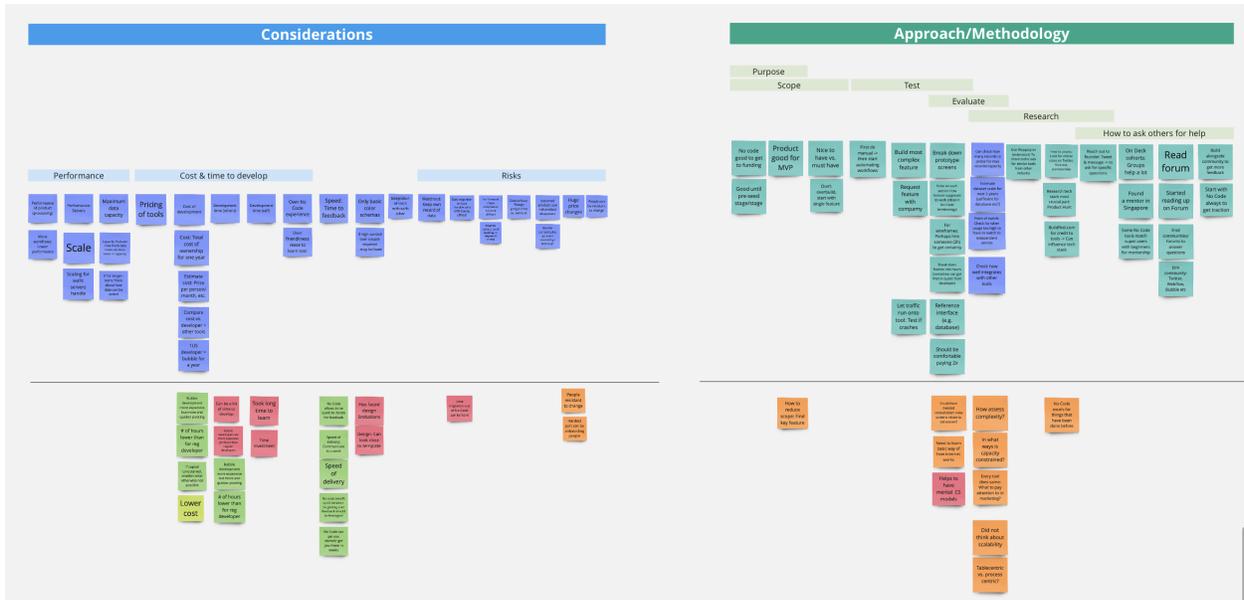
As mentioned for analysis 2, I created an initial interview guide covering questions on personal introduction to no code, process on no code go/no go decision, relevant considerations, unforeseen changes and limitations. The core focus of the interview was spent on the decision making process and teasing out the limits of the possible in no code. For special cases I "double-clicked" on special circumstances to understand considerations more deeply. An example of this was a startup built in raw code and then completely re-built in no code. For the "base" interview guide, please see the [appendix](#). Finally, I tested this interview guide structure in one dry run interview.

After completing all ten interviews (including documentation), I coded key insights across different dimensions on sticky notes next to the respective part in the interview documentation. These dimensions were "approach to no code building", "considerations", "limitations", "benefits", "introduction to no code" and "other important info" (each dimension representing a different color).



Screenshot of Miro digital board (zoomed out, thus not legible here)

Thereafter, I took all color-coded insights (sticky notes in image) and started clustering these insights within each dimensions. Specifically, I was most interested in laying out a methodology on how to test whether to build in no code or raw code and collecting limitations of no code building. Thus, I organized and synthesized insights around “Approach/Methodology” and “Considerations” and mapped respective benefits (light green), downsides (red) and other insights (orange) under the matching “consideration” or “approach” (column-wise).



After this, I took the “approach” steps (turquoise) and ordered them in accordance with the process described in user interviews. This became the skeleton of the final evaluation methodology and the ground structure for the guide. Using this as a base, I could fill in the the “considerations” (blue) at the point of the process where they can be

best evaluated. And finally, I wrote the guide based on this outline and filled in examples from user interviews whenever possible to make decisions more concrete for the target audience. Finally, this research from user interviews was augmented with deep-dive internet research on no code tool companies' online content to give concrete examples (e.g. pricing structures in relation to scalability).

### 3b. Findings of “Founder’s Guide to No Code”

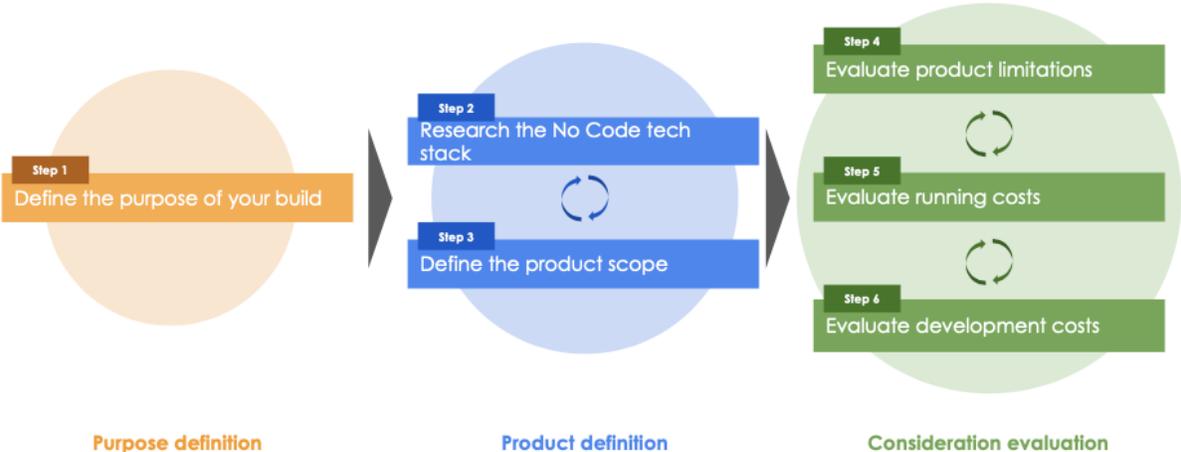
#### Findings in guide format

As mentioned above, the target user of the guide often reads digestible, actionable articles on Medium, LinkedIn, etc. for advice on decisions. These articles are often a way of how insights and learnings circulate within the startup community. Thus, it was a natural fit to format my learnings on methodology and considerations as a guide in the style of a Medium article. Lastly, in the spirit of no code, I decided to publish this article online using a No Code tool (Notion + [super.so](https://super.so) for hosting).

#### Founder’s guide to no code product decisions

The distilled best practice approach at a high-level involved three phases: Purpose definition, product definition, consideration evaluation.

### Founder’s approach to no code build decisions



The guide goes through each of these phases outlining the sub-steps, process and considerations and gives concrete examples and benchmarks for decision making along the way.

## A Founder's Guide to No Code

### III. Discussion

#### Summary

In this project, I set out to answer three questions. The first goal was to understand what characterizes no code entrepreneurship. The qualitative coding and quantitative analysis of the 100 no code startups featured on No Code Founders uncovered that *education* and *consumer* startups were strongly overrepresented compared to YCombinator startups. This hints at no code potentially being especially well equipped to cover education use cases, which require less complex technical features and focus on content instead. It also suggests that it lowers the barrier to entry (or build) for *consumer* and *education* startups.

We found that 2/3 of the no code startups were founded by people without computer science background and mostly run by solo founders, which typically is the less common model as explained above. This could perhaps be seen as evidence that no code is reducing the barrier to entry to founders by not requiring a CS background and enabling founders to take on multiple hats alone (marketer, developer, user researcher, designer).

Lastly, we identified the most common features / product types amongst the 100 no code startups to be directories/aggregator features, newsletters/content/blog post features, marketplace features, online community features and online shop features. The strong representation of these features in no code products hints at no code being use-case optimized as opposed to horizontally usable (e.g. unlike raw code which one use to build any feature).

The second goal of this project was to understand who the founder user personas are to whom no code is relevant. From the 10 interviews I conducted, I found three distinct founder personas: Side-project founder, prototype-stage founder and product-build

founder. Of those three, the prototype-stage and product-build founders often had least experience (e.g. did not experiment and “play” with no code as much upfront) and for whom decision making had the biggest impact (e.g. build product for next two years vs. side-project). Thus, for my guide I most focused on the needs and goals of these personas in my guide.

The third goal of my project was to provide guidance to early-stage founders in determining whether to build their product in no code or raw code. Gathering experiences and insights from my interviews with 10 no code founders, no code developers and CTOs, I distilled a best practice approach to this no code build evaluation: Purpose definition, product definition and consideration evaluation (at highest level). Lastly, I laid out the key considerations for this evaluation regarding product (design limitations, scalability limitations, storage capacity, performance limitation), development costs and running costs.

## **Challenges**

One of the main challenges I encountered in my analysis of the 100 no code startups was that I could not uncover the tech stack more reliably on the backend. While each startup reported the no code tools they used, these were not always fully reliable (as I found when comparing reported frontend tools with frontend tools identified by [builtwith.com](https://builtwith.com)). Further, it was also hard to clearly identify the stage the startup was in: idea validation (beta and waitlist) vs. market validation (live product) vs. failed product. To get reliable data of the no code tech stack creating a survey and sending this to no code founders for responses may be more effective. This could also help uncover “operational product types” and features (e.g. common workflows, etc.) across no code startups, which are more hidden in the backend. Lastly, it is also hard to find benchmarks for a truly global population of startups as every list or directory of startups has its own bias (as does YCombinator).

Another challenge I faced for the third artifact - the “Founder’s Guide to No Code” - was to write a guide for people with heterogenous technical (e.g. different levels of understanding on how websites are built) and product experience (e.g. different levels of understanding of what mockups, wireframes, etc. are). Thus, it is a tradeoff to write for a larger, more heterogenous audience (different levels of technical expertise but similar questions around no code) vs. for a very narrowly defined, homogenous audience.

## Future work

In future work, I see a lot of potential in writing “deep-dive” chapters of the initial guide I created. Specifically, I believe it would be relevant to founders to get more specific on limitations (e.g. scalability, performance) on a tool level and analyze this systematically. Further, an in-depth evaluation of tools and what they excel in can help people select the right tool better as well (e.g. evaluate automation workflow tools by ease of use, scalability, time to build, etc.). This data could be collected with an online survey (challenge perhaps: getting traction on Twitter). Another helpful piece of guidance, could be to create templates for founders to fill in their research and accompany them in the recommended approach, I laid out in the current guide. It could be helpful to have breakdown chapters of specific assessments for founders who have less product or technical experience. For example, when saying one should pick the most complex feature first, one could guide a founder through an exercise of how to assess “feature complexity”.

Lastly, a very relevant group of users I did not cover in my research are people working at enterprises. Big companies are likely to see huge changes to operations and software development due to no code, as well. Understanding the different personas to whom no code is relevant in corporation, analyzing their needs and motivations, uncovering enterprise-specific opportunities (e.g. no code tools effectively to automate & improve their operations) and writing an “Enterprise Guide to No Code” would definitely provide a lot of value to companies at a time when the no code industry is seeing a lot of advancement.

# IV. Appendix

## A. Interview guide questions

### Intro

- How did you get into No Code?
- How did you learn more within No Code at first?
- At what stage did you decide/think about building the product? At what point did No code become relevant?

### Doing

- At what stage did you decide/think about building the product?
- What was your decision process for building in No Code the first time? Can you guide me through the steps?
  - Was it always clear you would build in No Code? What was the trigger?
  - Was it a decision or default to build in No code?
  - What research did you do on No Code upfront? What did you learn?
- How do you determine whether to build in No Code now?
  - What are your considerations? What considerations were relevant?
  - How did you assess scalability and potential performance issues? How relevant was this?
  - How do you test this? What did you test? What was the outcome?
- After having built in No code, what did you not foresee/ what would you do differently?
- After having built in No Code, what were your biggest learnings? What do you see the role of No Code as?
- What does your decision to build in No Code ultimately come down to? How do you choose/test tooling?
- Did you build yourself or get an agency? How did you decide that? Time? Cost?

### Longlist: Detail

- What does the No Code tech stack look like?
- What part is No Code vs. Low Code vs. Raw Code? How did you decide that?

## V. Sources

1. *Gartner forecasts worldwide low-code development technologies market to grow 23% in 2021*. Gartner. (n.d.). Retrieved May 2, 2022, from <https://www.gartner.com/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021>
2. Kostereva, K. (2022, April 14). *Council post: No-code is the future of software: Here are five critical things to drive success in 2022 and beyond*. Forbes. Retrieved May 3, 2022, from <https://www.forbes.com/sites/forbestechcouncil/2022/03/18/no-code-is-the-future-of-software-here-are-five-critical-things-to-drive-success-in-2022-and-beyond/?sh=12b309a96bd1>
3. *No-code and the rise of the "long-tail" software startup*. Blog. (n.d.). Retrieved May 5, 2022, from <https://blog.airdev.co/posts/no-code-and-the-rise-of-the-long-tail-software-startup>
3. *Understanding the significance of the worldwide developer forecast, 2020–2025: Part-time application developers lead growth*. IDC. (n.d.). Retrieved May 3, 2022, from <https://www.idc.com/getdoc.jsp?containerId=US47199121>
4. Garcia, K. (2021, July 30). *What is low code? A comprehensive guide*. Retool Blog. Retrieved May 3, 2022, from <https://retool.com/blog/what-is-low-code/>
5. Garcia, K. (2021, July 30). *What is low code? A comprehensive guide*. Retool Blog. Retrieved May 3, 2022, from <https://retool.com/blog/what-is-low-code/>
6. Wikimedia Foundation. (2022, March 12). *Y combinator*. Wikipedia. Retrieved May 3, 2022, from [https://en.wikipedia.org/wiki/Y\\_Combinator](https://en.wikipedia.org/wiki/Y_Combinator)
7. Somers, M. (2018, May 30). *2 founders are not always better than 1*. MIT Sloan. Retrieved May 5, 2022, from <https://mitsloan.mit.edu/ideas-made-to-matter/2-founders-are-not-always-better-1>