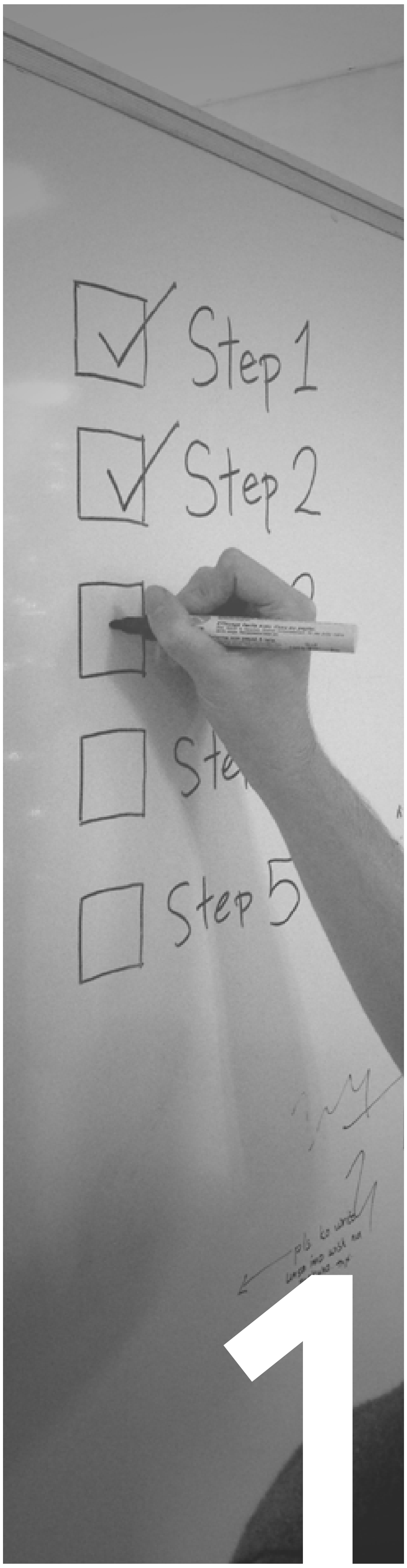
Arcanys

5 STEPS TO BUILDING THE PERFECT REQUIREMENTS



INTRODUCTION

In a recent survey conducted on over 100 software development service providers*, it has been discovered that most projects are delayed because the requirements are poorly formulated: **No one would ever build a house without having an architect create plans for it.** When you want to create your own software, it goes about the same way.

Over the past 15 years of working on software projects, we have helped hundreds of clients build their applications, but we also helped them **define their web and mobile applications, as it is the single most important step to succeed in a project.** While a conjunction of factors can make a project fail, the lack of clear requirements will make your project fail (or at least greatly suffer) in every single case. It's the lack of clear requirements.

To help you succeed with your project, you will learn the following throughout this document:

- Why good requirements are so important
- What requirements are used for
- How they will benefit your project as a whole
- How to build great requirements in 5 steps
  - Answer these general questions about your project
  - Think in a design centric approach
  - Build and describe your frontend interface
  - Don't forget about the backend
  - Think about the logistics
- **Bonus:** How to read a proposal based on your requirements

Along the way, you will discover that **building requirements is not as boring as it sounds,** and it's actually a very exciting part of the project. It will help you refine your business idea, think as the end-user and learn more about what you are going to build, **saving you tens of thousands of dollars down the road and increasing your success rate.**

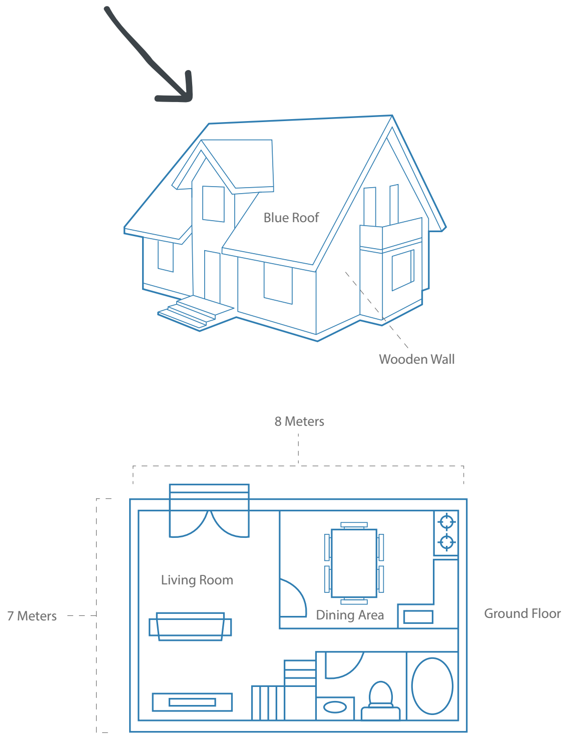
1A A good requirements document is the single most important element

We like the analogy of building a house when talking about the requirements of a project. The requirements are basically the plans the architect will draw for the house of your dreams. You would never just go to a contract or and tell him: I want to build a house, how much does it cost? Or if you did, he'd tell you: "Let's go talk to the architect to find out."

Do you want to build your house

with this?

or this?



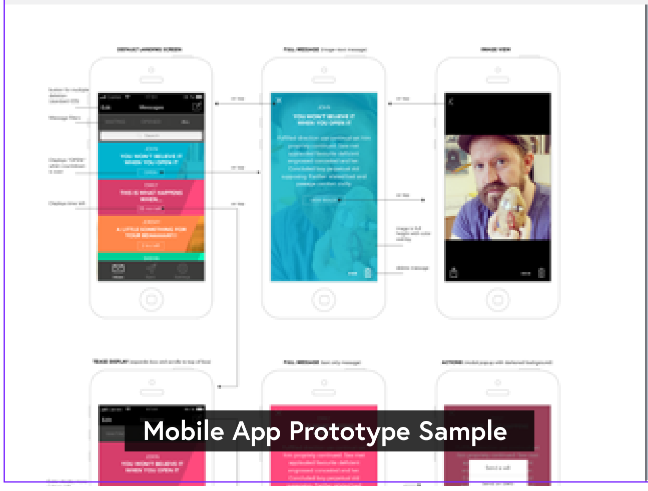
Architectural Requirements

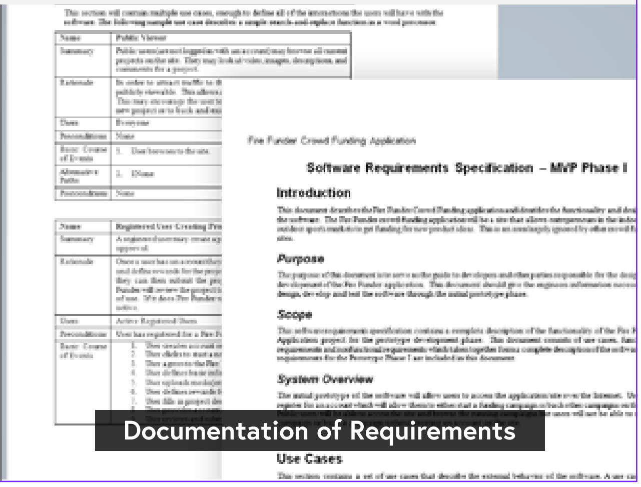
Building requirements for your project is done the same way with designers and business analysts, talking to you and trying to understand what your project is all about. **It is an important step to eliminate uncertainty.** If you have a project that has been started already and you expect another provider to carry on, the requirements document of what remains to be built is also necessary.

The actual requirements are ideally done in two forms, complementing each other:

1. A design centric approach with screen wireframes or designs with a short description of every functionality on each screen. This is more often the job of the UI / UX designer, who focuses on the usability of the application / web interface.
2. A more thorough text approach which describes in details the general information of the document, the business rules and notifications of the project as well as any other considerations you deem important for your provider to understand the project in an exhaustive manner. This is usually the job of the business analyst, who focuses on the functionalities of the application. Both the UI / UX designer and the business analyst work hand in hand during this process to come to the best possible understanding of your project.

Example of UI / UI design of the interface and of the requirements document




Documentation of Requirements

Fire Funder Crowd Funding Application

Software Requirements Specification – MVP Phase I

Introduction

[illegible]

Purpose

[illegible]

Scope

[illegible]

System Overview

[illegible]

Use Cases

[illegible]

Your requirements are basically the central piece of information your software development provider will follow to build your application from the ground up. It is:

- The overall expression of your idea ad your needs in a structured way
- The basis for communication between you and the team to collaborate on the same grounds
- The basis for communication to investors if you are trying to raise some funds to get your application developed
- The most important piece of information to get an estimation of the time, personal and financial resources needed to get your project off the ground. You will have an idea about the cost and the timelines to make your plans a reality

1B How requirements will benefit your project as a whole

Building good requirements is never easy because it involves a lot of work adding and removing features, thinking about the users and the value you want to deliver to them, and even how your business model fits into what you are trying to achieve. **It really helps you put everything in perspective.**

You are taking ideas from your mind to put them down on paper. It's a necessary process that will strengthen your whole project, and help you:

- Think about everything necessary and get rid of useless features
- Prioritise the key features for your MVP
- Be able to show your project to other people in a way that is easy for them to understand, and provide feedback.
- Allow all key decision makers to agree on functionalities (too often down the road a decision maker who wasn't consulted may object and force important changes)

MVP or Minimum Viable Product
In product development, the minimum viable product (MVP) is the product with the highest return on investment versus risk. It is the sweet spot between products without the required features that fail at launch and the products with too many features that cut return and increase risk.

A minimum viable product has just the core features that allow the product to be deployed, and get user feedback as early as possible. Truth of the matter, most people inventing software cannot always put themselves into the shoes of their target users, and getting early user feedback allows you to change the strategy or modify functionalities that you hadn't considered.

Basically, it is a strategy that avoids building products that customers do not want, to maximize the information learned about the customer per dollar spent.
The process is iterated until a desirable product/market fit is obtained, or until the product is deemed to be non-viable.
until a desirable product/market fit is obtained, or until the product is deemed to be non-viable.

1C What requirements are used for

A well done requirements document will give you access to critical information to get your project moving:

- A clear understanding of the project through wireframes or screen designs
- Detailed explanation about how things are expected to work (workflows, written descriptions)
- Sometimes, a non-functional prototype allowing you to browse through your application
- An estimation in terms of cost and time needed to build your application (you can hardly make a sensible estimation without good requirements)
- Information about technological choices (programming language, frameworks, etc.)

HOW TO BUILD REQUIREMENTS

So now what? This sounds like a daunting task?

Well not really. **It's actually one of the most interesting and creative parts of your journey.** And as you might have guessed, there is not just one way to build requirements. There are many, depending on your project. We will present here the most common ways to build your requirements, and what we really need to get a decent understanding of your project.

Functional and non-functional specifications

Along the way, there will be a few terms you will need to get familiar with, which are divided into 4 main categories that will cover the functional and non-functional specifications.

We think that an illustration is worth a thousand words, and design-driven specifications are in most cases the most comprehensive and efficient way to go.

Along with the designs / wireframes, the functionalities of the application (web or mobile) should be described in order to make sure nothing important is forgotten, as it might have an impact on the development time and the cost of the project.

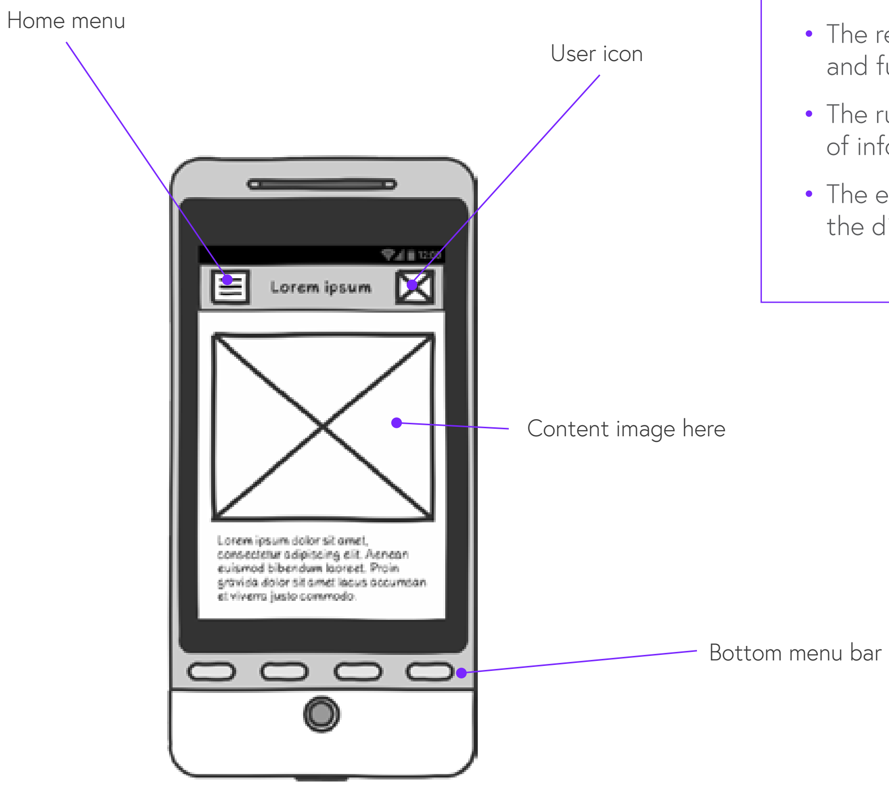
Wireframe

A website wireframe, is a visual guide that represents the skeletal framework of a website or an application. Wireframes are created for the purpose of arranging elements to best accomplish a particular purpose. The wireframe depicts the page layout or arrangement of the website's content, including interface elements and navigational systems, and how they work together.

The wireframe usually lacks typographic style, color, or graphics, since the main focus lies in functionality, behavior, and priority of content. In other words, it focuses on what a screen does, not what it looks like. Wireframes can be pencil drawings or sketches on a whiteboard, or they can be produced by means of a broad array of free or commercial software applications, such as Balsamiq®

Wireframes focus on:

- The range of functions available
- The relative priorities of the information and functions
- The rules for displaying certain kinds of information
- The effect of different scenarios on the display


Home menu
User icon
Lorem ipsum
Content image here
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.
Bottom menu bar

Mobile Wireframe done in Balsamiq®

The 5 steps to perfect requirements

Step 1. Answer these general questions about your project

It's important for anyone working on your project to understand a few basics about who you are, what you want to achieve, and for who. Tell us more about you:

- The goals you'd like to achieve
- Who is this project for? Who will use your application?
- Do you have any competition?
- What are the timelines? When would you like to launch your application?
- What is your realistic budget for your application? Is there a budget for the MVP and is there a budget for additional developments?
- And any other information that can help your provider to understand the project

Step 2. Think in a design centric approach

The design encompasses various terms related to how your application will look like, and how users (clients and administrators alike) will interact with your application. Usually, we will talk about designs to describe screens of the web or mobile application, that will be used and implemented as final design of the application.

Most people will start with drafting some wireframes (for example a wire framing tool balsamic, see in the resources section what tools we recommend) to explain the rough idea about their project before they turn to designers to actually create the actual design of all the screens of your application in specific tools made for this (Photoshop, InDesign, etc.) and will deliver the designs to you in PSD files and sometimes in prototyping tools such as Proto.io.

The final designs will then be used to be implemented in the application by the software developers as they build the frontend and the backend.

Step 3. Build and describe your frontend interface

The frontend is the visible part of the application for all the users. It is the graphical interface and its behaviour with all kind of users (customers, visitors, administrator, etc) using your platform. **This is the center piece of the specifications you need to work on as it will determine later how the other parts of the project will be defined (backend and logistics).**

Defining the way your application is structured and will look like for all users is the best and most efficient way to tell the stories of your users, as everyone will be able to envision how things work in the glimpse of an eye. You don't need to explain it with a 50 pages document, **you just have to show it to us.**

The first step is to imagine a **site map** or a **flow map** of how the users will navigate in the application. It will basically tell you **how the users will navigate through the application and inform you about how many screens you will need to build** for your users to be able to use it properly.

However, besides the wireframes / designs of the screens and the flows between the screens, an explanation about the functionalities is a very important aspect of your specifications.

This is the step where **UX / UI Experts and Business Analysts will add a lot of value to help you build a world-class fronted that will be both intuitive and pleasing to the eyes of your users** to increase their adoption and engagement with your product if you are not too sure about how to make it work best.

If your application is bigger than 10 screens, it is necessary to build **user stories**, which describe the steps a user goes through while using the application. This allows, in addition to the designs, to **clarify how the application will work** and provide an exhaustive list of the application functionalities. User stories are described fol lowing a specific formulation:

- **As a Visitor**
  - I want to register to the website
  - So I can access my profile section
- **As a Manager**
  - I want to access the reporting protected area
  - So I can view the monthly turnover

With these 2 sentences, we can identify:

- 2 roles (visitor and manager)
- 2 screens (profile and reporting)
- 2 actions (register and login)
- 2 functions (calculate the monthly turnover and access the secure area)

Not bad! You can now see how much value is created for the project with only 40 words.

If you do not know how to build user stories, we twill help you out with this part, as this is an essential part for us to translate your visual information and descriptions into a valid estimation of your project.

People who will work on this part are:

Prepare the specifications and design:

- UX / UI Designers to design your application
- Business Analysts to help you out with your functional and non-functional requirements (and user stories)

Implement, code and test the software:

- Slicers and HTML, CSS, bootstrap experts to integrate your design into a responsive or non responsive design (mobile compatible web pages)
- Frontend developers (Javascript, Angular.JS, etc.)
- Testers to find bugs and validate the functional aspect of the application

What is required to define the frontend are the following:

- Sitemap
- Wireframes / designs of your screens
- Descriptions of each function of each screen
- Bonus: User stories to help us understand who does what

Step 4. Don't forget about the backend

The backend is basically what's under the hood, the engine. **It's everything you don't see that makes your platform run without issues.** It goes from the database, to the users and **rules management** as well as **notifications** and other **business processes** you will need. The backend does not relate to the admin part of your application, which is also part of the frontend. A few examples below:

Notifications

Description	Type	Condition
Registration confirmation	✉ Email	Valid registration
Order confirmation	✉ Email	Payment accepted
Login from new device notification	💬 SMS	Login into the account from new device

Business Rules

Module	Rule
E-commerce	A product cannot be ordered without approval if the quantity of the stock is below 3 units
	Orders above 10,000 USD should not suggest Paypal as a payment option
	A user cannot place an order if there are missing information in the address field

Who will work on these to:

Prepare the specifications and design?

- Business Analysts to help you out with your functional and non-functional requirements
- Software Architects to design all complex issues such as databases (MongoDB, MySQL, etc.) and hosting (AWS, etc.) or backend as a service (Parse)

Implement, code and test the software?

- Software Architects to implement all complex issues such as databases (MongoDB, MySQL, etc.) and hosting (AWS, etc.) or backend as a service (Parse)
- Backend developers to build it (Node.Js, Frameworks such as Symfony2, Drupal, etc.)
- Testers to find bugs and validate the robustness and the rules of the application

Step 5. Think about the logistics

This part includes everything that will help you get your project running smoothly, such as:

- Compatibility
- Project management methodology (mostly Agile, Scrum mode)
- Code repository and management (GitHub, Bitbucket)
- Technological choices: Programming language, frameworks
- Hosting, etc
- Project phases: do you want to build everything specified, do you have an MVP in mind, and develop the rest when you have validated your hypothesis or reached a certain user base, or waiting to get additional funding after showing investors you have a tangible product already?

Based on the project information about the goals, user base, etc, as well as the frontend description, the documentation of your internal processes and business rules, the team will be able to determine how to build and size the engine you will need, as well as the integrations it would need with existing server / client services or new additional applications (APIs). We will be able to make technological recommendations for your needs.

Final note: We believe it is also very important for you to know what you are NOT doing. You have to exclude a whole bunch of stuff you know you don't want to do for your business at the start. It will help you clarify your mind on what is essential and non-essential for you to succeed. Trimming down and simplifying is healthy and can save you a lot of hassle, money and time down the road.

BONUS: HOW TO READ A PROPOSAL FROM A PROVIDER, AND WHAT MISTAKES TO AVOID

A lot of the items you should have in the proposal have been discussed in the points above, but there are another few important items to make sure you fully understand before signing a proposal from a provider.

Ideally, you want to have 2 to 3 providers you discuss with for your project, so that you can compare the quotes.

It is important that you don't have too many providers bidding on one project, as this will be extremely time consuming for you to analyse the proposal thoroughly, and you might get confused by having to process too much information. It's better to have a good feel of who you want to closely look at your project, as a lot of work is usually involved (between 15 and 100 hours depending on the projects).

3A Budget and deadlines

Needless to say: **When it's too good to be true, it probably is.**

If you are looking at building an application that works by a professional team, thinking you can get something done for below USD 3,000 is usually an illusion. Developing software is the same as building a house or a car: it's a complex process that has to be done by people who (should) know what they are doing, and usually takes longer than 1 week.

Project Type		Budget (USD)	Deadlines
	Smartphone app	5,000 to 50,000	1 to 4 months
	Web application	10,000 to 100,000	3 to 12 months
	Website	3,000 to 15,000	1 to 3 months
	UX / UI Design (no coding)	1,000 to 10,000	1 to 2 months

3B Fixed price vs hourly based work

This choice can be made mostly depending on 2 factors: the provider, and the project type. Some providers work on fixed price, some others work on an hourly basis. In addition, some projects setups / technologies may allow a fixed price (when the scope is very well defined and the requirements leave no room for interpretation) while some other not (big evolving projects, or with unclear scopes, with a lot of R&D, etc.).

It is very important to clarify exactly what is included in the proposal in both cases. Read the fine print of everything included and excluded in the proposal. Sometimes, you might have an attractive fixed price, but a lot of exclusions and an expensive hourly rate for everything that is excluded from the initial scope.

3A Payment terms

There are risks for both the client and the provider with payments. **The ideal situation is where the risk is spread fairly between the parties.** In our case, we require monthly upfront payments, but deliver the code as soon as the payment is done, on a daily basis. This is probably the best option as milestones are scheduled and the work is delivered continuously for the client to verify the quality and quantity of the code produced at anytime.

3D Ownership of the code

Clients paying for an application to be developed for themselves should own all rights linked to the work done and should own the exclusives rights linked to it (meaning the provider cannot reuse the code to sell it to other clients). If you are buying an off-the-shelf solution (such as Microsoft Office), it is a different story, since you are purchasing only a license to use the software.

3E Technologies involved

There are many technologies out there. It is important for you to check how dependant on your provider you are given the technology they suggest. **We discourage any client to pay a provider to build solutions on a proprietary framework for custom applications** (unless it is extremely specific, and done by a renowned provider), as it will be very hard for another provider to work on a framework they have no idea about.

As an example: You can't really bring a Ferrari to any garage for repair, as those beautiful beasts are extremely delicate and require specific knowledge. But a Ford, for example, share a lot of same technologies with other common brands, meaning that any kind of mechanic is more or less able to repair them. It goes the same for proprietary frameworks vs open source (and well documented technologies).

Resources:

If you feel like you want to start building parts of your requirements on your own to facilitate the evaluation of your project, here are some free resources you can use:

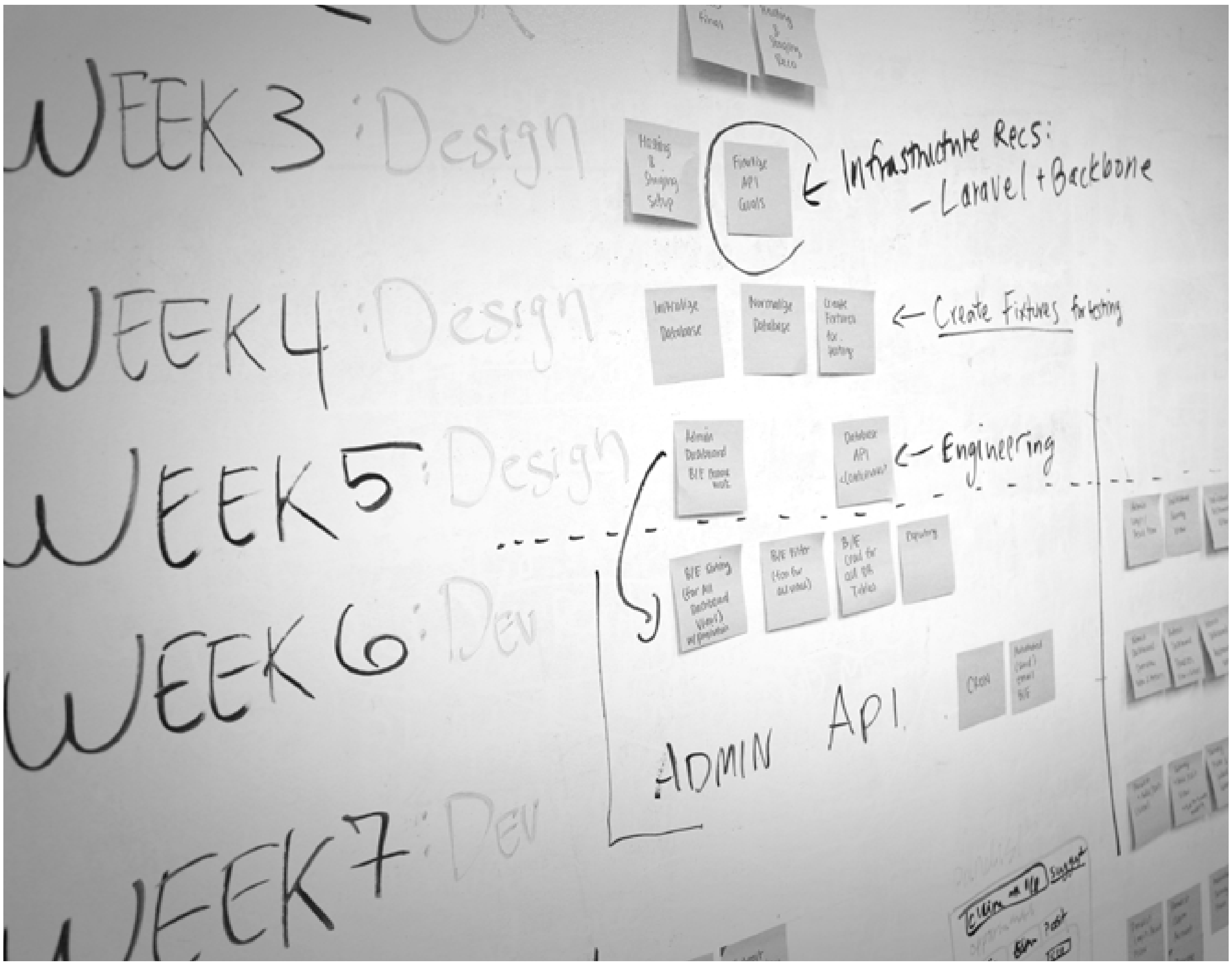
Wire-framing tool: https://balsamiq.com

Mind maps and brainstorming tool: https://bubbl.us/examples

Site maps: http://slickplan.com

Processus: http://lucidchart.com

Remember that the design is one key element of the project and the user adoption of your application. You want to have a professional and knowledgeable UX / UI designer to work on the screens of your application.



CONCLUSION

We hope this eBook clarifies all the steps that are necessary for you to specify and start your project right. You have probably heard a thousand times the saying that "failure to prepare is preparing to fail", and it has never been more true when it comes to build software.

Remember, those 5 steps will help you a long way to prepare your project:

1 Answer important general questions about your project
2 Think in a design centric approach
3 Build and describe your frontend interface
4 Don't forget about the backend
5 Think about logistics

Don't let your business or startup ambitions die before you give them a chance to become a winner on the market. Use a professional software development and design firms to get an A-team working on your project.

If you are interested in knowing more about how to avoid an application development horror story, **contact me** to get more insights!

We can get you started by a free strategic analysis of your project and provide you with additional insights such as:

- Outsourcing 101: How, when and where to outsource.
- 5 key benefits of outsourcing for startups / businesses.
- 7 dangers of outsourcing and how to avoid them.
- How to structure outsourcing contracts & protect your IP.
- Outsourcing 102: Manage, communicate and release effectively.

Arcanys

ABOUT ARCANYS

Arcanys is a Swiss software development company with a delivery center in Cebu City, Philippines. Our focus is in the development and implementation of great and innovative ideas into smart software solutions for enterprises and startups.

Our core business is to help our clients from the design of the specifications to the release of a software project. We provide and manage teams of full-time software developers collaborating with offshore IT teams who are primarily based in North America, Europe, and Australia. We are dedicated to helping clients spot areas for improvements in their innovation process, from solid specifications to limiting errors and bugs, with a keen interest in delivering projects on time; just as you expect it.

With our extensive experience in dealing with onshore and offshore clients, you can be assured not only with the quality of the output, but also with the quality of your outsourcing relationship with us. Talk to us today to find out how we can help you achieve your software development goals.

CONTACT ARCANYS

For business inquiries, drop us a line at **fred@arcanys.com** or leave us a message through our contact form at **https://www.arcanys.com/contact/.**