

# FROM MVP TO A FINAL PRODUCT

EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS



# Table of contents

**MVP**

**startup**

**idea**

**MVP partner**

**case studies**

1. Why you should start from MVP.
2. How to choose the right MVP partner?
3. Software development pricing models.
4. Factors that determine the MVP cost.
5. Native vs hybrid app – find your perfect fit.
6. Software development stages (with a time range).
7. Ease your pivot using MVP.
8. How to avoid common startup mistakes while developing products.
9. Estimate your MVP.





# WHY YOU SHOULD START FROM MVP

EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS



# Minimum Viable Product



**Mateusz Klimczak,**  
Partner & Project Manager  
@ bright inventions

**MVP stands for Minimum Viable Product.** It is the **first version of your product** containing **key features**. MVPs are **low-cost products** that you can show to early adopters and receive first feedback. A well-defined MVP serves as the **foundation for great software**.

Do you have an idea for software, but no clue what to do next? **Here are the reasons why you should start from MVP.**

# 1. Validate idea

MVP lets you **validate your assumptions** about the target audience in a **timely and low-cost manner**.

If the MVP turns out not to be successful, you can quickly rethink your strategy without having to click „undo” all the way. Respectively, if the MVP proves to be a great success, then you know you are going in the right direction.

# 2. Save costs

MVP is an investment in the base of your product. **Choosing to build only crucial features helps to reduce costs.** Low budget doesn't need to be an obstacle.

Low-budget MVP development has a couple of advantages:

- helps to make rational decisions,
- helps to be creative,
- protects startups from losing money on a product that won't attract clients,
- helps to create an app that users need and want.

### 3. Embrace the change

Thanks to a well-thought MVP development process you can **adjust your solution to address different needs or market challenges**. It's easier and cheaper to do it with a small product like MVP rather than with a large solution.

Read more on that in the [chapter on MVP pivots](#).

### 4. Short time to market

Developing crucial features helps to **reduce time to market** and prevent app owners from spending too much money on a product that hasn't been verified by the market yet.

Business trends and technologies transform every day, so **developing MVPs helps quickly enter the market and easily adapt to any possible business or technical changes**.

## 5. Face challenges upfront

MVP is also an opportunity to meet the first technical challenges that could have been missed during the prototyping phase such as:

- accessing geo location in the background,
- handling Bluetooth communication with other devices.

It's better to handle them as soon as i it possible.

## 6. Test and observe – don't assume

With an experience MVP partner you can derive sufficient information about your MVP users.

**Observing how users interact with an MVP allows startups to focus on what caught their users' attention the most.**

It also sets the groundwork for planning an **effective marketing strategy built around the actual value** that proved to be most useful to users.

## 7. Global solutions also started small

Amazon, Facebook, Uber, and Twitter all have kicked off with a “barebone product” before taking the world and becoming what they are today. Now, those are some MVP success stories, aren't they?

## 8. Groundwork for fruitful cooperation

You've guessed it – building an MVP lets both clients and MVP development agency test out way more than the product itself. During the time spent drafting and developing the MVP, both sides get to see what it's like working together.

By the time the MVP development agency finishes the first phase of the project, you can all picture what can be expected if you decide to work together in the long run.



## 9. Reduce risk

By launching an MVP, you can test your product or service in the market, and gather valuable feedback **without risking a significant investment.**

This helps to avoid the risk of launching a product that does not have market demand.

## 10. MVP is a cornerstone of your product

Even though MVP is a basic version of the product, it should be treated as a **solid base for what's next.**

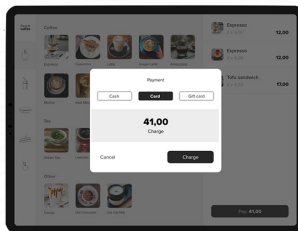
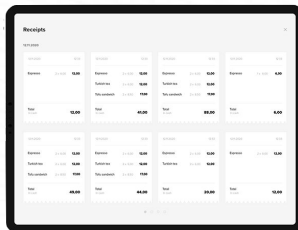
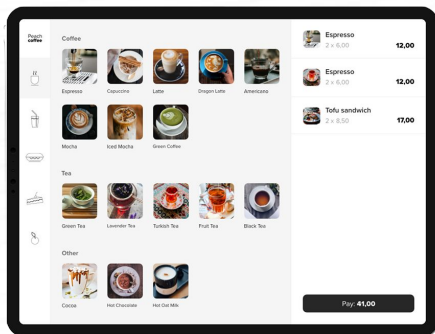
**Well done MVP will benefit in the near future** during another development stage. MVP gives a chance to create a well-thought software solution, based on actual users' needs and market trends.

# ***JUST EAT POS***

# **MVP CASE STUDY**

**RETAIL SYSTEM USED BY RESTAURANTS IN UK AND ISRAEL - SUCCESS STORY**

## b. From early-stage startup to a part of Just Eat Takeaway family



The system was built around these products:

- **ePOS iOS and Android applications** used by restaurant staff.
- **White label application** for restaurants.
- **iOS and Android driver applications.**
- **Backoffice solutions** including managing integrations and API connections.

[read a full success story](#)



# HOW TO CHOOSE THE RIGHT MVP PARTNER?

EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS



# Choosing the right MVP partner



**Michał Łukasiewicz,**  
Co-founder & Tech lead  
@ bright inventions

Deciding on the right **MVP development agency** is the first challenge to face on the road to launch your product.

Here are things to consider while choosing the right company.



# Closely located MVP partner

Nowadays we usually work on distributed teams around the world. Slack, Google Meet, Zoom etc. are great on daily basis. However, **if there's a chance, try to work with a partner that is "one flight away"**. Meeting in person might be more beneficial than you think.

Choosing a company from central Europe gives many opportunities for effective international cooperation. For example, Bright Inventions is based in Poland. Poland is in Central European Time (CET) Zone which makes it possible to organize daily online meetings with companies from all over Europe, as well as Israel or even the USA.





# Shared values

Since you're going to be working together very closely it's vital that you **both share the same values and that their working culture is close to yours.**

Do you hate when somebody is late for a meeting? Are you obsessed with keeping the discussion to the point? What is your definition of "done"?

**Vet your potential partner before any agreement is signed.** Make sure you can keep up with them on a personal level. You don't have to be best friends, but there need to be solid grounds for mutual appreciation, trust and understanding.



# Extension of your team

You run a startup and therefore things change rapidly. Each day brings new ideas, things to check out, more requests from the clients. You should be able to share it all with the developers working on your product without having to go through any sort of legal layer.

If adding even the smallest change to the feature set requires signing an annex with them – it's not worth it. If there are things that you would have no problem asking your in-house developer, but for some reason are prevented from asking the same from a developer working for your partner company, then it's not worth it either.

**You should be able to work with them as if they were part of your team.** Effectively they should feel that they are a part of your team. There's a catch: you also need to treat them as a part of your team.



# Work out a good payment model

In order to make the above possible it's good to **work out a simple to-understand payment model**.

You might like the benefits of a **fixed-price approach**. Yet in reality that would quickly limit the scope of changes that you can reasonably request from them.

**A sensible approach might be to work in sprints and have them charge you a fixed amount after each sprint.** It lets you refocus after each short iteration and nobody will make any fuss over requirements changing from sprint to sprint. Your vision will evolve and your partner should adopt.

Read more on that in the next chapter.



# Argues with you

It's easy to find a partner who would just blindly follow your requests. **It's much harder to find one that would put your ideas to the test and challenge you.** One that would **truly care about your ultimate goal** and would openly discuss their insight until your (or their) point is validated.

Let's say that you want to add full offline support to the app you're building (which is usually a tall order but doesn't seem so). They can either blindly follow or convince you that there are more pressing issues in the backlog that would benefit your users more.

If you've chosen the right partner, they will surely have more **technical experience** than you have and they will also have **some business experience** from the past projects.



## Stable team

Before you start relying on a MVP development agency you absolutely need to make sure that it's a **company that maintains a stable team of high-quality programmers**. You don't want them to restaff your project team every few weeks.

You need to have a sense of stability and be sure that the people who work on your project won't get reassigned willy-nilly to another gig.

Also, if there's any way you can get more familiar with the individual developers who will work on your project (their personal blogs, conference talks, etc.), it's a great plus.



# Estimates

People who will work on your project will have to come up with different sorts of estimates. It starts with a **high-level estimate before the kickoff**, i.e. how many sprints might we need to accomplish a given feature set.

Then there are **estimations made during sprint planning** and estimates made on the go when you check with developers how much time they need to complete a given task. It's very hard to check but you'll be much better off if you can trust the estimates coming from them.

If you realize after the first couple of sprints that their estimates are way off, you have to openly discuss it and expect them to be more realistic. You make business decisions based on the estimates you get from them. Obviously, you always have to plan for some buffer but you can't accept things taking twice the time they estimated. **An estimate is not a promise and it's not a commitment but it should mean something.**





# Progress

You need to be able to track progress easily and on a regular basis. It starts with the source code repository. Be sure that the **code gets pushed to your GitHub or BitBucket account.**

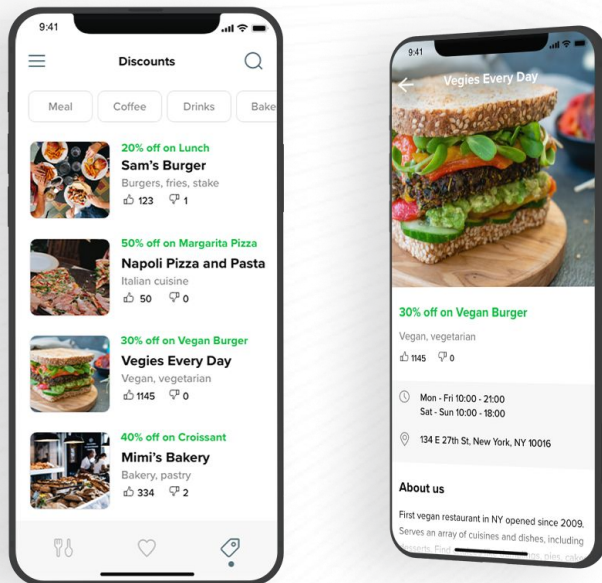
Make sure there's a **continuous integration system** that builds all of the components after each code change and that automatically deploys it or pushes for testing.

Ensure that the **continuous integration system stays deployed to machines that you control**, so that you can still use it if you stop working together.

# ***FINEBITE* MVP** **CASE STUDY**

**A BEACON-BASED MOBILE LOYALTY APP - SUCCESS STORY**

# b. Discover the best restaurants in the biggest Polish cities



## Key factors of the MVP success:

- Finebite originated from a loyalty app – **Everytap**.
- Everytap was a popular app around consumers and restaurants.
- While launching MVP of Finebite app owners already had a loyal group of users who were interested in dining booking solution.

[read a full success story](#)



# **SOFTWARE DEVELOPMENT PRICING MODELS**

**EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS**



# Pricing models



**Mateusz Klimczak,**  
Partner & Project Manager  
@ bright inventions

**Fixed price or Time and Material?** What is the best pricing model for software development services? Or **maybe there is a third option** to consider?

Let's dive into the **pros and cons of popular pricing models for software development.**

# Fixed-price model

A fixed price model is a situation when you (as a client) agree with a MVP development agency a price set for a job to be done in a specific period of time. For example, you pay a fixed price for Minimum Viable Product (MVP).

# Time and Material (T&M) model

You pay only for the real time spent on the project and the resources (materials) needed to complete the job.



## b. **Fixed price pros**

- It gives you (a client) a sense of security.
- You know how much the development process will cost.
- You don't have to worry about any surprises on the road e.g. budget changes (in theory).
- It is convenient at the beginning of the cooperation when you want to test the software development company and still are building trust toward a software vendor.

## b. **Fixed price cons**

- It is not a flexible option. It is in opposition to Agile methods.
- Changing plans during the development process is a very common practice and the fixed price model doesn't work well in these cases.
- If you need to change the tasks or need to create features that weren't estimated earlier as soon as possible, then it may cause some pauses in the development process. Due to the fact that you and the software development agency have to go back to the valuation stage.
- You receive only the final price of all tasks. You don't know the separate costs of every task. As a result **you are unaware of a very important aspect of software development – which features cost you the most.** If some of the features are particularly expensive – you could skip them if they are not crucial for the app to succeed.

## b. **Time and Material pros**

- It is a model price compatible with the Agile methodology.
- It is flexible – you can quickly change the development process while the work is on. Adding new tasks usually doesn't interfere with other tasks.
- You understand much better what time, effort and money is behind every feature. You know the price of every feature and based on that knowledge you can optimize the development process according to your budget and business plan.
- It also works perfectly during the maintenance phase when the software development team should be available if something happens with the product. That service can be easily estimated thanks to the T&M pricing model.

## b. **Time and Material cons**

- You never know the final price of the development process. You get the price after some features are done (depending on the number of hours and resources used on the tasks).
- You have to trust the software development agency that the time they spend on the project is accurate and reasonable.
- Sometimes you have to involve an expert that will verify the number of working hours established by the software development studio.

b.

**Fortunately, there is the third option that combines prones of both systems.**

## b. **Hybrid pricing model**

- MVP agency presents the fixed price, but for a smaller piece of the software development cake.
- The fixed price contains the job planned to do within the sprint (usually a sprint lasts 2 weeks).
- Before each sprint there is a planning meeting with the client and software development company. Both sides establish what needs to be done during the upcoming sprint.
- According to that software development company creates a budget proposal. When the client accepts it, they know what is going to be delivered within the next 2 weeks and how much it will cost.
- When the plan has to be changed because, for example, other features have to be developed asap, we have flexibility to quickly adjust to the expected changes.



# **FACTORS THAT DETERMINE THE MVP COST**

**EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS**



# The cost of MVP



**Michał Łukasiewicz,**  
Co-founder & Tech lead  
@ bright inventions

There is no easy answer to that question. It's like buying a car. You can buy an operating car for less than 10 000 dollars but you can also buy a Ferrari for over 1 million dollars. It depends on how advanced the product you have in mind.

**Check out 4 main factors that affect the cost of developing MVP.** Taking all of them into consideration helps to answer the big question – how much will my app cost me?



# 1. How advanced your app should be

Well-advanced apps like Uber or Google Maps may cost hundreds of thousands of dollars. Fortunately, startups usually don't have to begin with that budget.

MVP helps you invest less money to check if your idea has market potential. **You start something small and develop it according to your users' needs.**

It's a bit like buying a cheaper car just to see if driving to work is fine by you. Let's assume that you just passed your driving test and want to check if driving to work will be better for you than riding a bus. If you're not sure what you want, the smart way is to buy a cheap car. Eventually, if you rather commute using buses there is a chance to sell your car without a feeling of a bad investment. However, if driving a car turns out to be a perfect solution for you – you can start to save money for a better vehicle.



## 2. The cost of app infrastructure

When startups try to estimate the project, they sometimes overlook the fact that the **budget should include not only an app itself but also the whole infrastructure** behind it.

Software needs to coexist with outside systems that generate extra costs. The most important is the server infrastructure needed to exchange data.

It is impossible to predict the cost of server infrastructure without going into detail. The price depends on what server provider you would choose and how big the infrastructure needs to be. It's like the budget for gasoline. Every car has a different fuel burn efficiency.

### 3. The cost of app maintenance

Guess what? It's like owning a car. ;) Buying a vehicle is probably the biggest, yet only the first payment of the new owner. Every car needs regular services and sometimes repairs. It relates to the IT world.

**Both Google and Apple regularly upgrade their systems. On top of it, new mobile devices are launched every year. Every app needs to be adjusted to these changes.** That brings extra costs such as developing and testing the new versions of the app.

What is more, Google and Apple provide changes on their app stores that often affect every app owner. As an example, Google demanded from app owners information about the age range of their app users. It is not easy and time-consuming to meet these types of requests but it is crucial to keep an eye on them otherwise your app may be even banned from the store.



## 4. Developing an app for iOS and Android doubles the price

It sometimes comes as a surprise to startups that developing an app for Google Play and App Store comes with 2 separate development processes which double the budget.

When you want to develop an advanced product it isn't advisable to do one software for both stores. Apple and Google have their different requirements, processes, and tools. Thus, you need two teams with different coding skills to create products for both top app stores.

However, there are situations when it might be more efficient to build your MVP as a hybrid solution dedicated to AppStore and Google Play. More on that in the [next chapter](#).

**Building two products for iOS and Android isn't a must!** You can first release MVP for Android or iOS users and then develop it for another platform.

## b. Do you want to estimate your MVP?



Mateusz Klimczak  
Partner & Project Manager  
@bright inventions

**let's talk about your MVP**

**Book a free consultation** with our team at **Bright Inventions**.  
Based on your requirements we will:

- Estimate your solution.
- Establish the best pricing model for both sides.
- In case of mobile solution, we will decide if it is necessary to create both versions for iOS and Android.

Without client's requirements it's difficult to present any price estimations up front, that's why **contact us** and we will get back to you within 48 hours.



b.

# **NATIVE VS HYBRID APP – FIND YOUR PERFECT FIT**

**EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS**

2023



# Native vs hybrid



**Mateusz Klimczak,**  
Partner & Project Manager  
@ bright inventions

What is the difference between a hybrid app and native one? Which mobile approach is better for your MVP?

Let's look closely at the mobile "battle" between native and hybrid apps.

# Hybrid app definition

A cross-platform app that **works for iOS and Android**.

In order to build it you use frameworks that transform your code into a solution for Android and iOS as well. One of the most popular frameworks are: **Flutter** (by Google) and **React Native** (by Facebook).

# Native app definition

An app developed **only for a specific platform**: iOS or Android.

You build it with native technologies dedicated to that process such as **Java or Kotlin for Android** and **Swift or Objective-C for iOS**.

b.

# Hybrid apps pros

- Hybrid apps are **easier and faster** to build and launch as MVP.
- They can work for MVPs with **no complex functionalities**.
- They are the **answer for startups with limited budget**.
- If you want to **quickly launch the MVP** to get first users and then start to look for investors, it might be a great option.

b.

## Hybrid apps cons

- You are dependent on other providers like Flutter or React Native. For example - Android will release an update, then you have to wait for React Native to adjust to the update. While building native app, you could immediately adjust to Android updates.
- They help to save money at first, but **in the long terms it might not pay off**. Adding more complex functionalities will be problematic. Eventually you might need to do the whole refactoring.
- iOS and Android has different requirements and characteristics, so **a hybrid app will never perform as well as an native mobile solution**.
- UI of hybrid app will always be **less user-friendly compared to native**. You must compromise to meet both platforms' requirements.
- You will never escape from native development anyway. **Features like geo localization or Bluetooth have to be built natively**.
- Also, when your audience is only on iOS or Android there is no point in going hybrid.

## b. **Native apps pros**

- You work with native technologies **limiting any dependencies**.
- You can **quickly adjust** to any Android or iOS updates.
- The community built around native development is bigger. There are **more resources and libraries for developers**.
- Native **apps are more user-friendly**. Your software is fully supported by Android or iOS. You have access to API, your code answers to every little aspect related to the software. Apps are easily optimised for different resolution screens or other specifics related to the new smartphones.



## b. **Native apps cons**

- You have to **double your budget to hire two separate teams** (iOS and Android developers) if you want to build apps for both platforms.
- It **usually takes more time to develop** native solution compared to cross-platform one.
- They require to **hire a team with mobile development knowledge and experience**. Hybrid apps can be develop by web developers who don't specialize in mobile development.

## b. **What to choose?**

**We are native app enthusiasts because we want to look at MVP as a cornerstone of your product.** Therefore native approach gives you a better ground to improve your product in the future, adding more complex functionalities.

However every MVP is different. **In some cases hybrid is a better choice.**

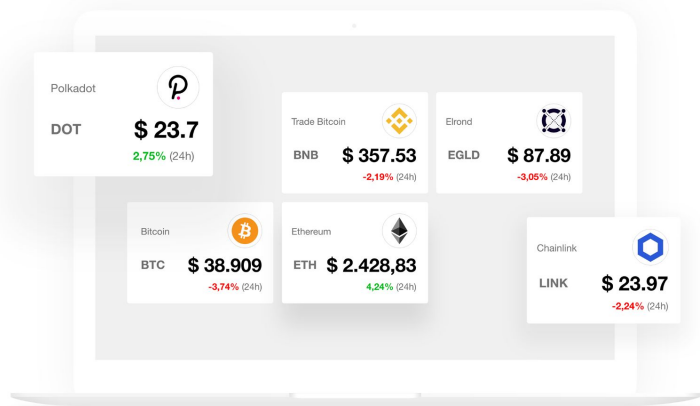
In the end you should talk it through with your software development team or software agency and choose the best option.

# **CRYPTOATLAS**

# **MVP CASE STUDY**

**SOCIAL PLATFORM FOR CRYPTO COMMUNITY - SUCCESS STORY**

## b. Connecting people passionate about cryptocurrency



CryptoAtlas is a social platform What was unique about this MVP? From the beginning it was designed and built with according to users feedback and needs.

Founder of CryptoAtlas has close relations with crypto community leaders which helped him gather feedback and plan MVP that the community wanted.

[read a full success story](#)



b.

# **SOFTWARE DEVELOPMENT STAGES**

**EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS**

2023



# Software development stages



**Mateusz Klimczak,**  
Partner & Project Manager  
@ bright inventions

From an idea to...? There are 4 main stages of building an app. **Learn more about all of software development steps.**

Although, remember that **the final flow always depends on the project.** You should establish it with your development team or MVP agency beforehand.



# Software development stages



## b. Idea

- Even the most unique idea doesn't guarantee success. This is only the beginning.
- This phase helps **quickly verify if the idea has some business potential**.
- Also, during this stage, you can **start looking for a suitable software development company**.
- Software development companies often conduct product workshops with the client to verify idea's potential.
- This stage helps to **answer another big question – how much would my app cost?** If the idea is too expensive to develop, you can modify it or even just give up on it.
- This is a moment to **establish goals** to measure your MVP success. However, these goals might change. While you are getting to know the industry, you might better understand what you need to succeed.
- Ideation is a significant time for product designers. During workshops with the client, designers can better understand the whole solution. This brings us to the Prototyping phase.

## b. **Prototyping & Product design** **[1-4 weeks]**

- The stage when a product designer creates the **first visual representation of the idea**.
- Prototyping phase contains usually **3 stages: wireframes, mockups, prototypes**.
- **Wireframes are very simple designs** that show the basic application flow.
- Then it's time for mockups that derive from wireframes. **Mockups ensure the client that all requirements are met**. Also, provide developers with info about UI elements.
- After that, it's time for **prototypes**. They are **basic clickable mockups or wireframes without any backend**. The product designer creates mockups to show testers how the solutions might work.
- This stage helps to eliminate mistakes that we weren't aware of till this point. You can even come to the conclusion that your app is too difficult or users would use it differently than expected.

b. ■

## Building MVP [4-20 weeks]

- It's **time to show first users your solutions**. No more theory and prototyping. Time to launch!
- MVP is a step towards successful product development.
- MVP is also an opportunity to **meet first technical challenges that could have been missed during prototyping phase** such as: accessing geo location in the background or handling bluetooth communication with other devices.
- Even though MVP is a basic version of the product, it should be treated as a **solid base for what's next**.

b.

## Development, QA & Maintenance [Infinity]

- At this stage, a software development company **regularly deploys some new features** to production.
- QA team makes sure that your software is secure, up-and-running and offers the best possible performance.
- **Every change in the product will affect your users.** Thus at this stage the priority for developers is to maintain the app stability. All development updates should be tested in a staging environment.
- The crucial part is to **analyze users' behavior** to see how they respond to new features.
- Also, at this stage mobile app developers have to regularly update the apps to cope with the Android and iOS changes.





# **EASE YOUR PIVOT USING MVP**

**EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS**





# Top reasons to pivot



**Izabela Pawlik,**  
Digital Marketing Specialist  
@ bright inventions

Sometimes **MVP launch** shows that **your product should pivot**. Although app pivoting feels scary, it isn't such a unique thing. It was done by the greatest ones.

Why did tech giants like Twitter or Instagram decide to change their product and strategy?

**Check out the reasons to pivot.**

# 1. Too strong competition [Twitter]

Before the **Twitter** team started to develop a popular microblogging they had been working on **Odeo**. That was a place where people could listen to podcasts and share music.

**Odeo** struggled to compete with **iTunes**. So the team pivoted and started to develop a new solution. They also decided to rebrand and that's how **Twitter** was born.

# 2. Different users needs [YouTube]

**YouTube** was a video dating website first. However, people didn't feel like uploading videos to attract date partners. At the same time they were eagerly uploading videos about their passions, holidays, family or friends.

In **YouTube's** case users practically did the whole pivoting for the owners.

### 3. Long for simplicity [Instagram]

Developing too many features may be confusing for users. Instagram learned that at the beginning. It was supposed to be a photo&note sharing solution and loyalty app called Burbnb.

The number of features made the prototype too complex, so the owners decided to focus on photo-sharing.

### 4. Adapting to market trends [Netflix]

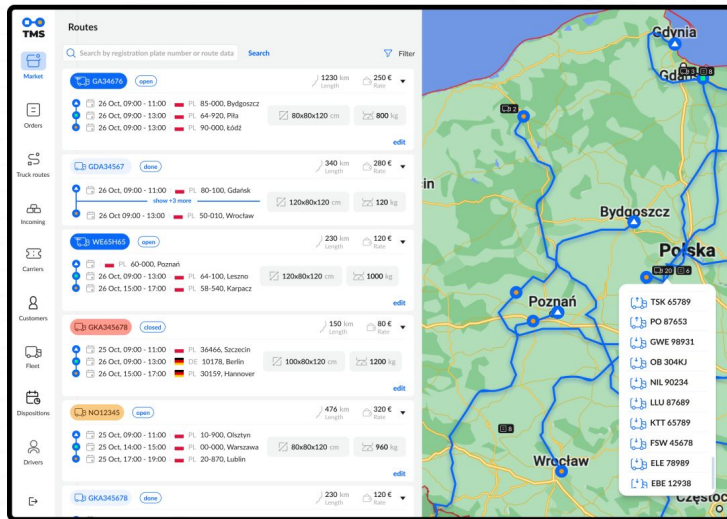
At first **Netflix** was an online rental platform for DVD movies. DVDs were mailed to American households.

Eventually, **Netflix transformed into a digital content service**. As you know, they don't just gather content, they create their own movies and TV series. **They pivoted their product following or even starting marketing trends.**

# ***TRANSPORT MANAGEMENT SYSTEM MVP*** **CASE STUDY**

**INTERNAL FREIGHT MANAGEMENT SYSTEM - SUCCESS STORY**

## b. MVP built to increase productivity within the company



The MVP owner – MAKOTSL needed a customized solution that would meet their needs in 100%.

The goal of the MVP was:

- To verify the potential of the system to be developed on a bigger scale.
- Transform employees work into a less repetitive, faster, and more automatic job.
- Effective utilization of vehicles' spare capacity.

[read a full success story](#)





# HOW TO AVOID COMMON **STARTUP** MISTAKES

EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS





# Common startup mistakes



**Mateusz Klimczak,**  
Partner & Project Manager  
@bright inventions

We've listed some of the most **common startup mistakes** in mobile app development.

There are **5 mistakes you definitely should and can avoid with our bright tips.**

# 1. Not verifying market potential

Always verify your app's potential on the market. There are many ways to do this: market research surveys, or consulting a professional agency, to name a few.

If you don't determine that your app addresses a genuine need on the market, you run the risk of failing.

# 2. Not establishing (or ignoring) MVP

MVP helps to establish a list of critical features necessary to launch a first version of the app.

Yet what about all those unexpected circumstances that might push you to consider updating your MVP halfway through? **Although pivots might be beneficial, it doesn't mean you should agree on every change.** Ensure it won't disrupt the development process.

### 3. Developing app on too many platforms

Sometimes it's worth focusing on one platform at the initial phase. **Instagram** was one of the successful mobile apps that was **first launched only on one platform (App Store)**.

This is likely because market research showed that potential users were using one platform more than the other.

### 4. Micromanaging (or neglecting) developers

You might feel tempted to stay in touch with your team 24/7, which can draw your developers' attention away from tasks. Conversely, you might be so confident that your team is handling everything well that you won't show up to meetings.

**Speak to your team about the frequency of contact** that will be most beneficial for the development process.

## 5. Not paying attention to app analytics

Once the app goes live, a lot of early-stage startups make the mistake of focusing their entire attention on App Store reviews, while **underestimating the role of app analytics.**

In fact, monitoring app performance, user behavior, and install and retention rates (among many other factors) should all be treated as detrimental to product development in the post-MVP phase.

## Bonus – analytics tools to consider

There's a whole array of analytics tools on the market. Here are examples of tools we use at Bright Inventions:

- Google Analytics 4
- Mixpanel
- Firebase Analytics
- DataDog
- Kibana
- AWS CloudWatch
- Graphana

## b. Do you want to estimate your MVP?



Mateusz Klimczak,  
Partner & Project Manager  
@ bright inventions

**let's talk about your MVP**

Book a free consultation with our team at **Bright Inventions**.  
Based on your requirements we will:

- Estimate your solution.
- Establish the best pricing model for both sides.
- In case of mobile solution, we will decide if it is necessary to create both versions for iOS and Android.

Without client's requirements it's difficult to present any price estimations up front, that's why **contact us** and we will get back to you within 48 hours.



**Bright Inventions** is a software consulting studio based in Gdansk, Poland, **operating since 2012**. Our expertise in **mobile, web, blockchain and IOT** systems has been highly appreciated by our clients from **UK, Germany, Netherlands, Norway, Israel** and more.

If you want to estimate your software solution, drop us a line.

[brightinventions.pl](https://brightinventions.pl)

[info@brightinventions.pl](mailto:info@brightinventions.pl)





# If you have any questions, contact



Michał Łukasiewicz,  
Co-founder & Tech Lead  
@ bright inventions



Mateusz Klimczak,  
Partner & Project Manager  
@ bright inventions



Izabela Pawlik,  
Digital Marketing Specialist  
@ bright inventions

**EBOOK FOR STARTUP FOUNDERS, CTOs & PRODUCT OWNERS**