IDEA\MOTIVE

IM 2018

IDEAMOTIVE'S DEVELOPMENT PROCESS EXPLAINED

or what it's like to work with an agile software house.

Who we are & what we do

Ideamotive is an agile software house from Poland. Our team of ~20 (and constantly growing!) is based in a start-up and tech mansion in one of the most charming districts of Warsaw – Żoliborz.

Our main fields of expertise are **Web Development**, **Mobile Development, Custom Software Development** and **UX/UI Design**. However, the overall objective of our work is to help you **scale your business**. That's why we also offer business and technical consultancy – including optimizing business processes, audits, reviews of code and much more. We specialize in **Ruby on Rails**, **React** and **React Native**.

Our **agile development process** is a result of long years of experience in software industry. We are very proud of our methods, because they allow us to stay creative and flexible **while** working within a proven structure. We created this whitepaper because we want to share our best practices that enable us to produce highest quality software.

We believe in agile approach, as much as we believe that high standards should become ubiquitous in software industry. We hope this document can be our humble contribution to raising overall quality of software.

TABLE OF CONTENTS

What “agile” means to us

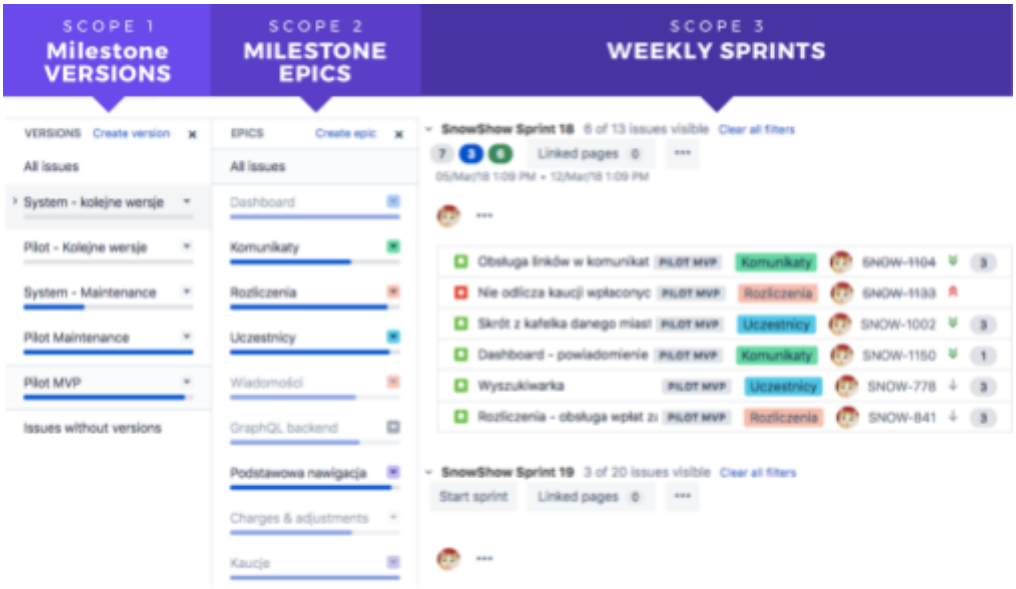


Image: Introducing the division of project into milestones, and milestones into weekly sprints

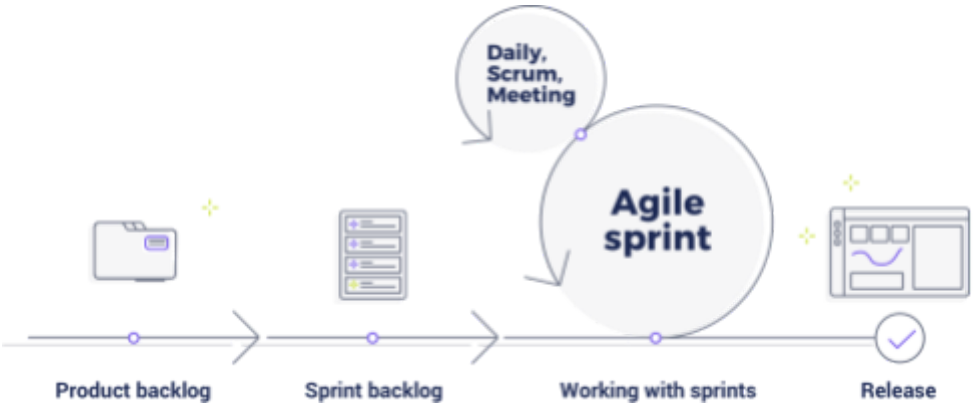
To us, “agile” is an approach to software development founded on **trust and transparency**. Focus, efficiency and flexibility are the key features of our workflow, which empower us to deliver best possible value to our clients.

At Ideamotive, agile product development is based on breaking the whole project down into smaller, graspable chunks of work. Usually, we divide your project into milestones, marked by releasing consecutive versions of the product. First milestone is typically an MVP (Minimum Viable Product).

Each milestone is then divided into weekly iterations, also known as Sprints. Each Sprint is meant to deliver a small and well-defined set of features which can be tested and feedbacked. This way of working requires our clients to actively participate in Weekly Sprint Meetings, evaluate development progress and plan for the following weeks together with us. Such working process allows us to improve every week and evaluate features and solutions as we

develop them. This, in turn, usually helps us bring about the final product much faster and better quality.

Agile software development also means that we bill you based on **time and materials,** rather than charging a fixed rate. You will simply pay for **the man-hours** actually spent on developing your product (we track them) + **the materials** we purchase to complete the project (only after the purchase is mutually agreed on).



What you can expect from working with us

- **High level of engagement.** It's not just about coding down your idea – we also help you develop and conceptualize it in the first place.
- **Project manager assigned to your project.** We want to ensure our work on your product runs as smoothly as possible.
- **Fluid communication.** You are encouraged to participate in our internal communication and meetings, as well as talk to the developers directly.
- **Involvement in the production process.** We like to engage our clients in the development process as much as possible. This way you can be sure that the project is unfolding in accord to your expectations.
- **Addressing problems immediately.** We use our broad experience to detect and solve potential issues even before they arise.

- **Constant monitoring of our progress.** You get full access to our workflow tracking software, where you can see current tasks & time we spend on them.
- **Convenient delivery and deployment system.** Continuous Delivery is intrinsic to our working methodology. We deploy new iterations to server up to a few times a day, to keep you updated.
- **Transparent pricing system.** We track each hour worked for you and we bill you based on those hours.
- **Rights to the source code and other assets.** You own the code, as well as mock-ups, graphics and layouts we design for your project. All of the source code is stored in repository.

This whitepaper is written for you, to let you better understand the way we work and communicate with our clients. We would like to guide you through the usual steps of designing a product – from scratch to a successful business asset.

However, please keep in mind that this is more of a general framework rather than fixed steps which we always follow. First and foremost, we love agile. This means that we always do our best to **adjust the work process**, **meet your project's demands** and **offer you a tailored solution**. Treat the below document as a structure – but one that can be adjusted every time we approach a new client.

> **You bring the idea. We bring our technical expertise to make it a reality.**

PART I: Pre-Development Phase

Discovery Call

Hearing about your product for the first time is one of the most exciting things for us. We are always open to arrange a free Discovery Call to... well, discover your idea!

We usually get in touch with new clients via recommendation, receiving an email inquiry or meeting them at an event. Once you get interested in developing your project with us, we schedule a Discovery Call to better understand you and your vision. This allows both parties to determine whether we are a good match for working together.

The Scope of Discovery Call

During the Discovery Call we first give the floor to you to introduce your project. Then we usually ask you a round of questions, to get an initial idea of the scope of required work and understand your experience with creating digital products.

The first type of questions is going to be focused on understanding your vision, business model and goals you want to achieve:

- Tell us more about your business: what is the mission, vision, context in which you operate?
- What is the purpose of the product/service that you want to create?
- What market demand is it answering?
- Who are your customers?
- Who is your main competitor?

- Who is going to be the product owner representing your business throughout the development process?

Product Owner = the person on the client's side who is responsible for ensuring that the activities planned in the Product Backlog are aligned with the product's vision and goals.

The second type of questions is focused on your experience with digital products and practical aspects of our work together:

- What is your experience of working with external contractors?
- Have you ever worked with a software house?
- Are you familiar with agile methodology?
- Do you have a trusted person who can support and advise you throughout the project?
- Do you have any materials in place? (sketches, mock-ups, MVPs, working product that needs audit)
- When would you like to start?
- What is your estimated budget for this project and how are you funding it?

In the end of the Discovery Call we should have a general understanding of your idea and its business model, and be sure that you have an understanding of our working process. That's why during the call we introduce you to our **agile methodology** and explain what it means that we charge you **based on time and materials**.

You can be certain of our discretion. Our usual practice is to sign a mutual Non-Disclosure Agreement (NDA)[**request document template**], either before or after the Discovery Call. We want to ensure that both yours and our business is protected, and that no sensitive information will be disclosed to third parties.

Assuming that you are interested in working with us, during the Discovery Call we plan the next steps together to prepare for the Estimation Workshop. Those preparations usually require you to share relevant materials with us (e.g. brief, wireframes, user stories). On our side, we do initial research and prepare an agenda for our first Estimation Workshop call.

The first Estimation Workshop call is free of charge. If there are more calls required to work out the time & costs estimation for the project, we decide together with the client on whether next calls/workshops should be paid. Additionally, we may offer creation of simple assets, such as wireframes or mock-ups, to visualize the idea before entering the Development Phase.

Services We Offer During Pre-Development Phase

The deliverables we can propose during the pre-development phase depend on how advanced your business idea or product is. In general, we have three main groups of clients to whom we offer following sets of services and/or assets, before entering the Development Phase.

1. If you have a **business product idea** and a **brief…**
   ... we need to start by gathering product requirements and prepare a **product design specification.** To get there, we might offer you **technical consulting** to concretize your idea and help you solve the initial conceptual problems. Finally, we often provide low-fi mock-ups or wireframes, to ensure that you and us are on the same page.
2. If you have materials, such as **brief, wireframes, mock-ups, user stories**…
   ... we gather product requirements by looking at these materials and talking to you in-depth about your project. At this stage, we usually offer **product design specification** and provide you with a piece of **technical advice** to ensure that your business concept is solid and well thought through.

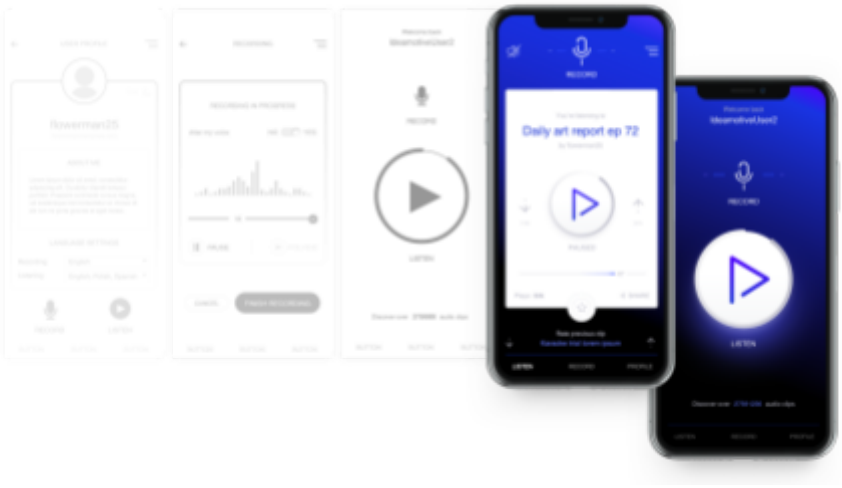


Image: VOICER: low-fi and high-fi mock-ups

3. If you have a **working product** or an **MVP**…
   ... we usually start with carrying out an **audit** to see where exactly your product needs improvement. Then we might go through a **technical consulting** workshop with you and **plan a roadmap** for actions to be taken.

> **NOTE:**
>
> Services we provide at this stage might be paid or unpaid, depending on the scope of work required to deliver them. Initially, both you and us need to invest some time to discuss the project and conclude whether we want to work together. If additional assets or services (e.g. mock-ups, audit, extensive research) are needed to reach this conclusion, we may charge you a fixed price for those.

Estimation Workshop

We believe that **concept informs implementation.** During the Estimation Workshop our goal is to help you conceptualize your idea as clearly as possible.

The workshop usually demands 1-3 calls after which you will receive an estimation of the time and costs required to complete your project. This stage is really important both for you and for us, as this is when we are really setting the project up. The point is to be as efficient as possible once we enter the Development Phase.

Some elements of the Estimation Workshop may also appear later on **in the Scoping Sessions**.

The key participants of the Estimation Workshop include: the **client**, **project manager**, **lead developer** who is likely to take on the project and, oftentimes, our **designer**. The main objectives are to clearly **envision the product, gather requirements and estimate time and costs** necessary to complete it.

The Scope of Estimation Workshop

The format of this workshop is kept as flexible as possible. We intend it to be a **collective thinking process**, during which everyone is free to ask questions and think of possible answers. Although the process itself should be unrestrained and creative, there are some typical issues we need to address. For example:

- **Vision:** How do you envision people using the product? What is the main purpose of it? Which functionalities do you want to encourage users to play with? What might be the context for using the product?
- **Problems:** What are the development challenges? What technical/legal/marketing/usability issues are we likely to encounter?
- **Solutions:** What are the possible solutions? How to implement them? Which problems need to be solved right away, and which decisions can be postponed?
- **Technical:** How is the system architecture going to work? What kind of data structures are needed? What types of algorithms will the product be using?
- **Marketing:** What customer need is your product answering? What problems does it solve? Who are the people likely to use the product? How do you envision possible user scenarios?
- **Other relevant issues:** Which platforms are we going to target? What does the competition look like?

The above are typical examples of questions that we might be answering together during the Estimation Workshop. Of course, the questions and

specific issues can differ significantly, depending on the product you have in mind.

As a result of the Estimation Workshop you will receive your **product design specification** and an **estimation of time and costs** required to complete the project. Throughout, we may be delivering wireframes, mockups and technical consultancy, according to your needs.

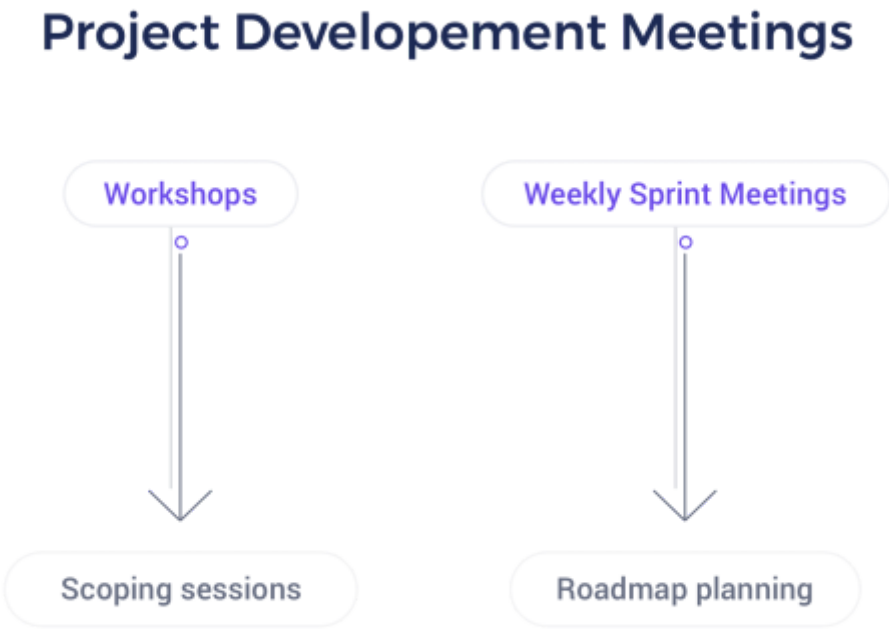
> **IMPORTANT NOTICE:**
>
> Please note that the time-and-cost estimation we make at this point is based on the **initial product requirements**. Whenever you change those requirements throughout the Development Phase, we will update our estimations to match the current scope of work. In any case, we will carefully track the time worked on your project to make sure we stay within budget. We prefer to use such transparent billing method that allows us to charge you the real costs – rather than including unnecessary contingency buffers.
>
> To learn more about our time –and-budget management, **go to PART III**.

To finalize the Pre-Development Phase and start the full-on development of your product, we need to sign a contract. You can read more about the provisions of our contract **in the Part IV of this whitepaper**.

PART II: Development Phase

The Development Phase starts when we have all agreed on the product requirements, acknowledged estimated time and costs and signed the contract. From now on, the main pillar of our work with you will be **different kinds of meetings**.

There are three main types of meetings we hold with our clients throughout the Project Development phase:



Scoping Sessions

A typical Scoping Session is dedicated to **discussing a particular feature of your product**. It usually happens once every 1-3 months throughout the Development Phase – but this can differ depending on the type of your

product or unforeseen issues. Some elements of a Scoping Session may also appear earlier, **during the Estimation Workshop**.

Formula & Preparations

We usually meet through Google Hangouts Meet, where we can work together on a shared screen. This way you are able to see all the work results achieved so far. A Scoping Session typically lasts somewhere between 4-6 hours. The key factor of a successful Scoping Session are **the people participating in the call**.

The representatives of Ideamotive are usually the **project manager**, **lead developer** and the **designer**. The participants from client's side should include someone who **supervises and understands the project in detail** (i.e. product owner) and, ideally, one or two **end-users**. The goal is to involve different people who can look at the product development from various angles.

We communicate with you before the call, in order to prepare for it. Ideamotive is usually responsible for the **meeting agenda**, along with **summarising** the discussed feature(s) and undertaking necessary **research**. As a client, you are in charge of specifying the **requirements**, preparing **questions** and bringing **relevant people** to the meeting.

What happens in the meeting?

A typical Scoping Session may consist of the following steps:

- **Brainstorming** on what needs improvement, what is the current business goal, what are the possible solutions, etc.
- **Defining problems** by breaking them down into smaller, more graspable sections – for example, coming up with specific user stories.

- **Deciding on the best solutions**, which includes prioritizing action steps (client) and assessing what is feasible (Ideamotive).
- **Creating documents** such as **roadmaps**, **mock-ups**, **specifications** or **application flows**, as a base for further work.

Scoping Sessions are the quintessence of our agile approach to software development (along with Weekly Sprints – read more below). They let us constantly update our understanding of the project as it unfolds, instead of remaining stuck in the initial assumptions. They also allow you to give us your feedback and point out arising problems in real time.

As much as we like being flexible and agile, we still need some structure and a work plan. This is why we periodically hold another type of workshop with you – Roadmap Planning.

Roadmap Planning

Image: Roadmap. Epics oriented in time.

Roadmap Planning usually happens once every 1-3 months and lets us review the tasks necessary to complete the next milestone. As a client, you are

actively involved in the Roadmap Planning with us. Simultaneously, we make sure to update the Roadmap internally, within the development team.

The goal of a Roadmap Planning session is to combine your business knowledge and our technical expertise to **come up with a detailed plan of action** for the following weeks or months. This ensures that we work within a structure and according to the project requirements – while also leaving space for adjustments and flexibility.

To prioritize next action steps and put them in the right order, we usually create a **value/cost matrix**. It serves as a visualization of the most cost-efficient and impactful actions to be taken. Additionally, it allows you to see what is the fastest way of moving your business forward.

Image: Example of Cost/Value matrix

Weekly Sprint Meetings

Every milestone that we specify together during Roadmap Planning is then broken down into graspable and specific chunks of work. They are what we

refer to as Weekly Sprints, or “iterations”. These weekly periods are our most basic working modules, which allow for careful planning and monitoring the progress.

You can read more about our workflow management in Part III.

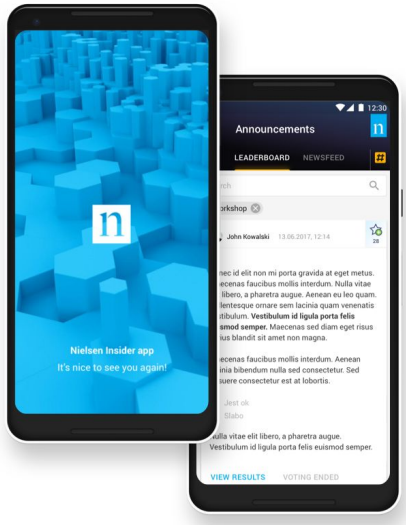
What happens in the meeting?

Weekly Sprint Meetings are dedicated to **summarizing previous week’s work** and making a detailed plan for the upcoming sprint. Usually, we would also update a general action plan for the following 2-3 weeks.

The meeting typically gathers the **project manager**, **lead developer** and **product owner** (person responsible for coordinating the project from client’s side). The first point on the agenda is to summarize the progress made in the past week. This allows you to become updated on the recent accomplishments, as well as technical problems. You also get to give us your feedback and report tickets or bugs that you or your users encountered.

Weekly Sprint Meetings are a chance to not only review the technical aspects of the project, but also the workflow and communication between you and us. If something needs fixing, this is the time to address it. Remember that we are open to making any adjustments, as long as they improve our work!

READ ONE OF OUR CASE STUDIES & SEE HOW AGILE WORKS IN PRACTICE



Nielsen Case Study

PART III: Our Workflow Management

As an agile software house, we put a lot of attention to optimizing and tracking our workflow. Because we charge you per hour of our work, we want to be as efficient as possible during that time. And we can account for our productivity.

Throughout the Development Phase we provide you with detailed documentation of all the project work. This allows you to see how much time we spend on which tasks, as well as monitor our progress in real time. Additionally, you have access to each of the developers in case you need to talk to any of them directly.

To get your head around everything, you have a Project Manager assigned to you and your business. You can think of him as your personal guide who keeps you informed, gathers your feedback and ensures you follow the communication happening around the project. He will also assist you with monitoring the current state of your product by giving you access to our task- and time-management tools.

Managing Tasks, Time and Budget in JIRA

JIRA is the software used by world-class developers, allowing to organize, plan and track all the work required in any type of project. Each of our developers uses it to keep record of their tasks and to view the overall progress of the rest of the team. JIRA ensures that every action necessary to build your product has a person assigned to and responsible for it.

Task & progress monitoring

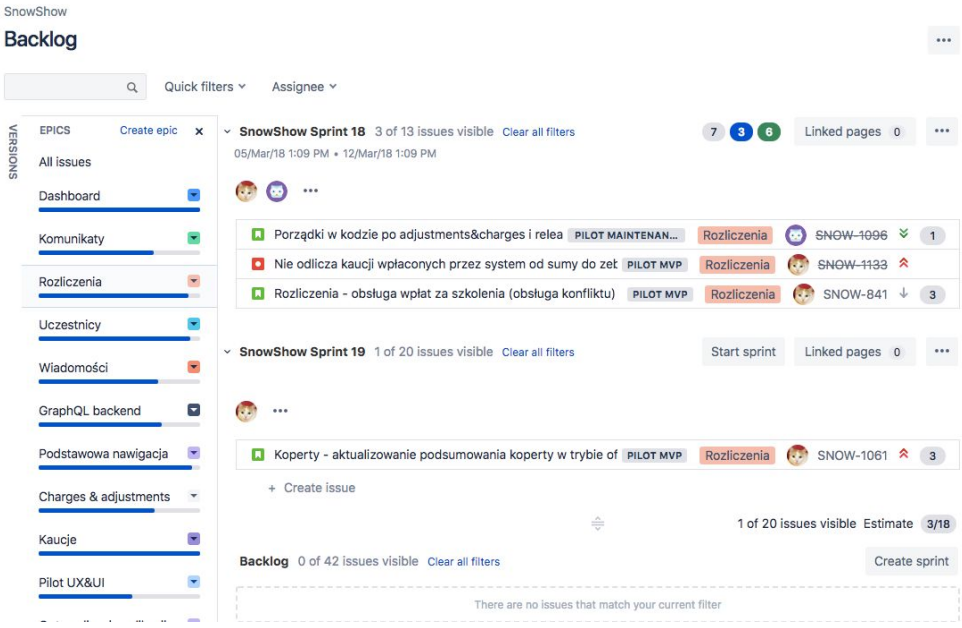


Image: JIRA view, tasks assigned to one developer

Our starting point to work management is defining **milestones**, which are synonymous to consecutive versions (or releases) of your project. Later on, we break them down into **epics**, which are shippable pieces of work, usually focused around developing a specific feature of your product. An example of an epic can be **creating user dashboard** or **implementing new payment method**.

Finally, each of the epics is divided into specific **tasks** assigned to particular developers in JIRA.

JIRA allows us to not only decide on who is doing what, but also monitor the progress and plan out our weekly sprints. Thanks to this, each developer remains focused on the tasks they are responsible for. At the same time, they can keep an eye on the overall progress of particular epics and milestones.

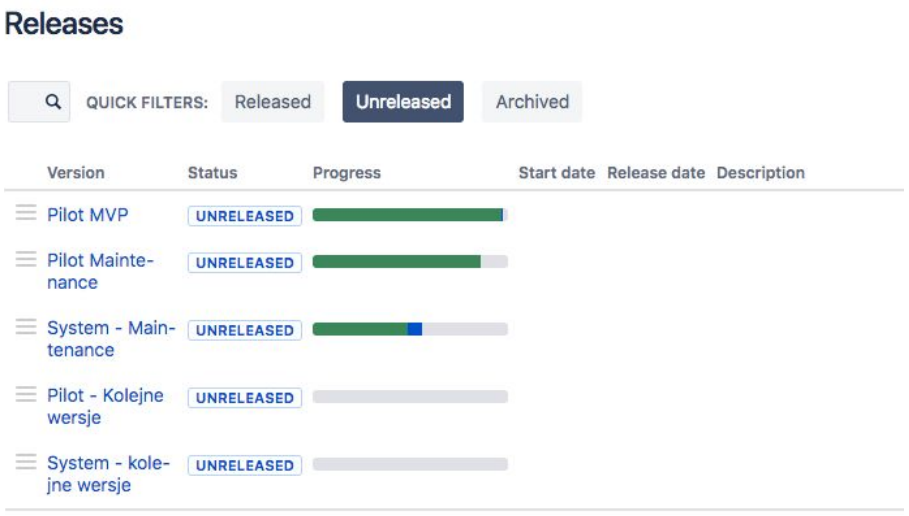


Image: JIRA view, progress of milestones

Time tracking

JIRA also allows us to track the time spent on each task. Every developer is responsible for tracking their own working time. Thanks to this, when we invoice you, we are able to account for all the man-hours we charge you for, telling you what exactly we did in that time.

Throughout the project you have complete access to our JIRA dashboard. This allows you to monitor the following information (in real time):

- What task is assigned to which developer;
- What is the current progress of any given task/epic/milestone;
- How much time we have already spent on particular tasks.

You also have the possibility to add comments under each task.

IDEA\MOTIVE

Timesheet Report

2017-12-01 - 2017-12-31

Summary

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Overall sum
Marcin	8.0			8.0	8.0	8.0	8.0	8.0			8.0	8.0	8.0	8.0	8.0			8.0	8.0	4.0	6.5	3.0					8.0	8.0	8.0			**141.5 h**
Witek	8.0			8.0	8.0	8.0	8.0	8.0			8.0	8.0	8.0	8.0	8.0			8.0	8.0	7.5	8.0	4.5					8.0	8.0	8.0			**148.0 h**
Adam																		1.5	1.25		2.75											**5.5 h**

Overall sum: **295.0** h

Detailed list of work

Date	Task & Description	Developer	Time spent
2017-12-01	Merge translation files from locales/views directory (JRP)	Witek Muszyński	2.0 h
2017-12-01	Merge translation files from locales/views directory (JRP)	Witek Muszyński	6.0 h
2017-12-01	String extraction to yml (JRP-)	Marcin Olszewski	8.0 h
2017-12-01			Sum: **16.0** h
2017-12-04	String extraction to yml (JRP-)	Marcin Olszewski	7.5 h
2017-12-04	General work (JRP)	Marcin Olszewski	0.5 h
2017-12-04	String extraction to yml (JRP-)	Witek Muszyński	8.0 h
2017-12-04			Sum: **16.0** h
2017-12-05	String extraction to yml (JRP-)	Witek Muszyński	7.0 h

1/6

Image: Timesheet Report with detailed list of work

Managing budget

We don't charge you a fixed price for your product for one simple reason. We want to leave you the flexibility to change the requirements, specification and features of your product as we develop it. We know from experience that this is natural – both you and us are only able to learn and realize certain things in the process of development, and not before.

Taking that your product is likely to evolve over time, we prefer to be transparent and we don't charge you contingency buffers to insure ourselves against a potential change in the scope of work. Instead, we carefully track the time we spend designing and developing and we then **bill you per hour of our work**. On top of that, we might add costs of materials we needed to purchase in order to build your product.

This doesn't mean that you don't have control over your spendings. We always ask you what your budget is in the first place and, no matter whether the scope of work changes or not, we make sure to **remain within budget**. We may only exceed it if we receive **an explicit permission** from you.

Additionally, we update our time-and-budget estimations whenever you change requirements or modify features of the product. We always inform you about those updates, so that you are constantly aware of the most recent assessment of the scope of work, project's remaining duration and anticipated costs.

> As unpredictable as software development might be, we make it as predictable as possible for you.

Planning expenses

Before entering the Development Phase, we estimate the number of man-hours assigned to your project per month. As the work continues, we continuously update this number, based on how the project is progressing. Any such updates are made in advance and clearly communicated to you.

Additionally, we ask you to set a limit on the number of man-hours you are happy to pay for each month. This serves as a spending cap for us, in case any unforeseen work arises – for example, we need to consult the UX designer. Having such limit set by you ensures that we fit all the required work within your budget.

In rare cases, we may encounter unpredictable issues, calling for increased number of man-hours. Such situations are exceptional and usually the excess is minor. If an issue like that occurs, we always contact you about it in

advance and make sure we find a solution that suits you and doesn't harm your finances.

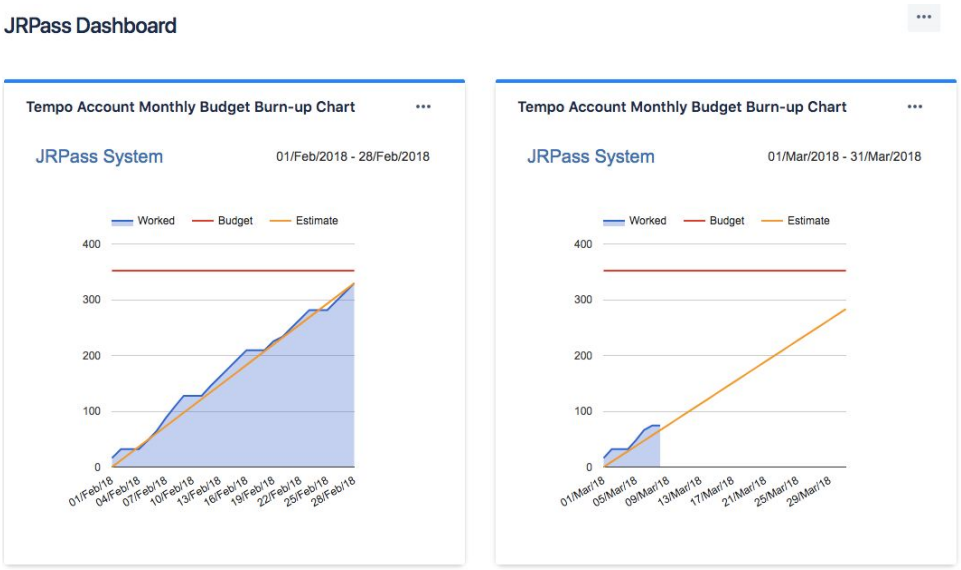


Image: Timesheet Report with detailed list of work

Additional materials & billing cycle

Purchasing additional materials and assets necessary to complete the project happens only **after we received your permission**. Examples of such assets may include: a graphic, particular font or server fee. We will always attach an invoice for the purchased materials together with the billing for our services.

Our billing cycle is monthly and this is how often you will receive an invoice from us. In the attachments, you will always see relevant documents accounting for the hours worked and any additional spendings.

REMEMBER:
We will never exceed your budget without consulting it with you first. You are always in control of your spendings.

Project Documentation in Confluence

Managing ongoing tasks to ensure that the process is flowing smoothly is one thing. The other is **storing all the information related to each task, epic and sprint**, as well as keeping **records of the meetings** we hold. To ensure you and us have all the information handy whenever we need it, we use Confluence for project documentation.

Confluence allows for complete transparency and good organization of information about the working process. Because all the requirements and meeting agreements are stored in one place, we can all easily refer to them if any doubts arise. Additionally, it eliminates redundant communication, because any member of the team has immediate access to complete project database.

On top of recording all the meetings and task-related information in Confluence, we **make sure our clients can see them**. You will get notified via email or Slack whenever we add any important documents.

Documenting epics and tasks

We store as much information as possible about the tasks and epics created in JIRA. In Confluence, we specify what exactly needs to get done: what are the requirements and particular actions to be taken. All the epics are well-defined and described based on our mutual agreements, meetings and workshops.

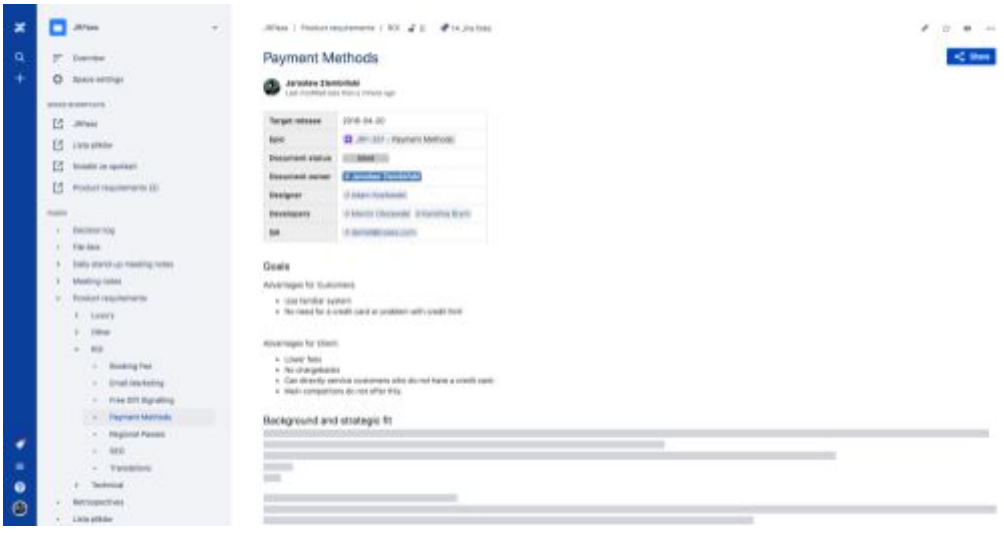


Image: Epic described in Confluence

Documenting meetings

Meeting documentation means taking minutes from all the sessions and gatherings we hold. We record each meeting with you (such as Estimation Workshops, Scoping Sessions or Roadmap Planning) and also our internal team gatherings – for example Daily Stand-Ups. The main purpose of meeting documentation is to track our communication and know exactly what the outcomes of each meeting were.

Daily Stand-Ups

Daily Stand-Up meetings are an important pillar of our work. They ensure that we communicate with each other on a regular basis and act as a team – rather than just as a bunch of individuals. This is the way we can offer you best possible value.

The main point of our Daily Stand-Ups is to **acknowledge where we are within the project** and to **exchange updates with the rest of the team**. For that purpose, each of our developers answers the following three questions daily:

- What have I done since the last stand-up?
- What am I going to do today?
- What challenges do I face?

This simple practice allows us to stay productive and aware of where we currently are in the project. Other benefits of Daily Stand-Ups include: **identifying challenges, maintaining supportive work atmosphere** and **transparency in our communication** with you. For this last reason, we give you access to all the records from our Daily Stand-Ups.

Image: Example of Daily stand-up questions

Communicating With You

We highly value our client's input in the development process. That's why we encourage **open communication** between you and the project team. We use different communication channels for different purposes.

- Through **Confluence** we give you access to the project records and updates from all kinds of meetings.
- In **JIRA** you can see what each of the developers is currently doing & you can comment on particular tasks directly.
- By holding **Weekly Sprint Meetings**, we provide space for communication about anything and everything currently going on in the project.
- During **Scoping Sessions** and **Roadmap Planning** we talk to you about specific issues and general scope of work.
- Ongoing communication via **Slack**, **emails** or **Google Hangouts Meet** ensures we stay in touch and send you most important updates and documents daily.

Our Project Manager will assist you with all the communication and ongoing queries. However, we understand that sometimes the best way to go about an issue is to talk to a developer. For this reason, we make it easy for you to communicate with our developers through JIRA, Slack or Google Hangouts Meet.

CONTACT US TO REQUEST A FREE DISCOVERY CALL ABOUT YOUR PROJECT

FILL ESTIMATE FORM

Part IV: Bonuses from Ideamotive

Insight Into Our Development Process

The crucial part of delivering your digital end-product is our development process. Over the years we have established proven practices to ensure highest possible quality of the software we produce. We make it a point to have **maximum control over the code**, work according to the **most recent standards** and **use experience from the past to write better code today**.

In this section, we would like to share with you what exactly we do throughout the Development Phase to deliver high-quality products.

Best day-to-day practices

- We use best practices, tools, conventions and coding methods that are well-known across software industry. This is the easiest way to ensure that any code produced by Ideamotive is commonly understood by all programmers – not just the ones writing it.
- Our developers understand business and therefore they are able to look at the product from your point of view. During meetings they actively propose technical solutions that suit your business model.
- In many tasks and processes, our developers rely on checklists. This lowers the risk of overlooking an important element of the process – such as installing an SSL certificate during release to production.
- Continuous Integration and Delivery are intrinsic parts of our working system. We cannot imagine software development without CI/CD – it saves a lot of our time and automates mundane and repetitive activities, e.g. tasks connected to testing or implementing changes in the code.

- We use Version Control System to securely store source code. This allows to experiment with all kinds of coding solutions without running the risk of losing what we have already accomplished.
- We automate what we can to make the production process smoother. A good example is using static code tools like RuboCop or ESlint to verify whether all of the code is in accord with the style guide. Such tools prevent developers from adding changes in repository which are not in alignment with the style guidelines. They also help us detect any possible mistakes early on.

Refactoring and updates

One of our priorities is to **maintain the project in good condition** and **avoid technical debt**. Technical debt appears when developers use short-cut solutions to build a product faster, which often lowers the quality of source code. This usually results in additional re-works, slow-downs and increased costs in the future, and may discourage high-class developers from working on your project.

We prefer a more far-sighted approach when building software. To keep projects up to date with the most recent technical releases, we always use newest libraries, plugins and frameworks available. There are several benefits to it:

- Any new potential developer can step in to the project easily if it is well-maintained;
- We eliminate the problem of technical debt, which, if big enough, could force you to rewrite the whole project in the future;
- The product is more secure if regularly updated. This security is connected to the support of external vendors, which might no longer be offered for older versions of libraries.

Learning and knowledge sharing

Software development is such a rapidly growing industry that there is no other way to stay in the game other than **constant learning**. This is why we dedicate time and resources on a regular basis to make sure that our developers are... developing themselves.

We keep an **in-house knowledge base,** in which we collect our best practices, insights and checklists. Our developers often hold workshops for each other, to share their most interesting case studies or solve problems together. We have also implemented **retrospective evaluation sessions** which we hold after completing sprints, milestones and projects. This creates an opportunity for the team to reflect on what went well and what could have been done better.

Apart from sharing the knowledge internally among ourselves, we also publish it on our blog. For an example, check out our post ***Materialized views in Ruby on Rails with scenic***.

Additionally, Ideamotive's developers form internal "support groups" for each of the technologies we use. Those groups serve as a **supplementary resource base** for documenting experience and consulting particular issues related to specific technologies. Because of the multiple projects we have completed as a software house, this internal resource base is quite rich and usually allows us to find the right solution very quickly.

On top of that, we provide **online courses**, **relevant books** and **external trainings** for our developers on demand.

Backlog and Estimates

Keeping a backlog is a great way of **planning ahead**, **estimating** and **prioritizing** all the actions necessary to move the development process forward. We use JIRA to schedule our tasks, estimate their complexity in story points (more below) and assign them to particular developers. You can read **more about it in PART III**.

All tasks are grouped into weekly sprints which, in turn, build up milestones of the product. In order to be able to organise our work and prioritize it, we have a **proven system to estimate tasks.**

How do we make our estimations accurate?

No estimation is perfect and we have to keep in mind that it is what it is – **an estimate**, not fortune-telling. However, there are certain things we do to maximize the accuracy of our estimations.

- We **involve client's relevant representative** (usually: the product owner) in the estimation process. By contributing their business knowledge and user stories, they usually give us a better understanding of the real scope of work connected to each task.
- Work estimation is a team sport. We try to do it as a group activity, to **bring different perspectives** to the table – those from developers, project managers and designers. This diversity is one of the most important factors for the accuracy of a project estimation.
- Dividing work items into **near-term and long-term** is crucial, as it helps our developers prioritize their tasks. This distinction also happens with an active involvement of a client's representative. Near-term tasks can be estimated very accurately, while the long-term ones are initially a rough assessment. As we get closer to them and learn more about the product, we are gradually able to make the estimations more precise.

- **We use story points** instead of hours to estimate our projects - which is a common practice in agile development teams. Estimating in story points solves the issue of non-development work creeping in and altering the forecasted time. It also removes the emotional attachment to deadlines and pushes the developers to make toughter decisions as to how complex a task is. In general, we found that story points allow us to estimate faster and more accurately.
- Last but not least, we **learn a lot from previous estimates**. Because we track story points assigned to certain tasks in the past, we can become more and more realistic with how we assess our work every day.

Tests and documentation

Testing is crucial to ensure software and applications' reliability. We use tests to provide smooth deploys, avoid unnecessary blocking bugs and perform change without hesitation about existing features.

The tests we run for our applications ensure that we can deploy every new and tested feature on demand, so it can be released to production. Our tests involve five simple steps:

1. The code is run against a testing application and rewritten until the test passes.
2. The developer adds the tested code into Version Control System.
3. The code is tested for compliance with the production environment – for that, it is uploaded on the Continuous Integration server.
4. Meanwhile, the developer and product owner run product's test in the browser.
5. Assuming that the code passed all the above tests, it is now ready for deployment to production. In practice, this means adding the newly-developed feature to your marketable product.

Along with testing, we put a lot of attention to gathering complete documentation of the development process. **We document...**

- **...all stages of the application setup.** This ensures that if new developers need to step in (for any reason), they can easily pick up on what is going on. Also, in case the project has to be put on hold (e.g. product owner needs to gather feedback), it is easier and faster to reactivate it after a break.
- **...all the key processes and business logic in Confluence.** This creates an accessible and comprehensible project database, which everyone involved can rely on. It is important for the developers to know the business context of your product, as it allows them to apply custom (and sometimes not so obvious!) solutions.
- **...the code itself.** This includes **documenting API** (Application Programming Interface), which facilitates further cooperation with third parties, as well as **documenting key mechanics** of the code, to decrease the risk of errors while implementing changes.

Cloud Infrastructure & Hosting

Good hosting used throughout the development process needs to ensure two things. First: that you don't need to make unnecessary investments even before your application is released. Second: that your product receives reliable support from the hosting staff, so that we can focus on the development.

The platform we use for hosting is Heroku. It is a cloud-based service with an excellent support team behind it. Thanks to the great service they provide, many potential problems are solved immediately – often without us even realizing that they existed in the first place. Additionally, Heroku is integrated with most workflow tools we use, and therefore it doesn't just host your

application – it also makes our work smoother (for example, thanks to **integration with the Review Apps** – see below).

If your product requires that, we can also offer you **custom deployment in the cloud**, supported by Amazon Web Services, Microsoft Azure or Google Cloud Platform.

Error Tracking

To track errors and stay on top of them, we use Rollbar. It is one of these tools that delivers exactly what it promises – helping us find errors even before our client does! Rollbar allows not to just identify errors, but also resolve them based on contextual metadata.

Rollbar supports all the platforms and technologies we use for building software. Additionally, it offers integrations with many of our work tools. A great example is the integration with Slack, through which Rollbar informs us about detected errors in real time.

Staging and Review Apps

Staging is the pre-production environment we deliver to clients, so that they can review features before the final release. To create this environment, we employ Review Apps.

The Review Apps can automatically create several review versions. This means that if we are developing a few features at the same time, all of those features can be evaluated independently, in their own dedicated environment. If a problem is found, the Review Apps make it way easier to find out which feature creates it.

The staging environment serves you to see **updates of the code as frequently as possible,** along with their practical consequences. We use this form of **presenting the development progress to you during our Weekly Sprint Meetings**.

Code reviews

A very important part of our development process is **the culture of code review** which we have developed across our teams. This ensures that the developers feel collectively responsible for the quality of the code and they are motivated to perform at their best. Simultaneously, the knowledge about the project and particular features becomes spread across the team. This is the best way to increase the so-called “bus factor” (i.e. the number of developers that would have to be hit by a bus – or, put less drastically: disappear from the project – in order for the knowledge about the project to be lost).

Once a developer has finished programming a certain feature, they perform a **pull request** to let others know that a particular part of the project is ready for release. Before this part of the code is merged to the master branch, it gets to be reviewed by other members of the team.

Depending on the quality, the new feature either gets **accepted and merged** to the master branch, or it is **rejected and returned** to its creator for fixes. We repeat this process as many times as it takes, before we reach the excellent quality we aim for.

Insight Into Our UX/UI Design Process

In our UX/UI Design Process we follow five basic steps to ensure highest possible quality and usability of the end product. We believe that solid preparation before entering the designing phase is key to good UX/UI design.

Please note that some of the UX/UI design might happen before entering the Development Phase. We write more about it in PART I, where we enlist the **services we offer during Pre-Development Phase**.

Because we have UX/UI designers in-house, they are in constant touch with the developers – which makes the two groups speak the same language. Designers constantly update developers on what is happening on their end of the job, which means that developers are able to immediately assess how feasible the UX/UI ideas will be to implement.

As a result, the team can detect any potential difficulties in advance and make both the design and development processes smoother.

Step 1: Defining

We are big fans of the statement that **concept informs implementation**. This is why we put a lot of effort into defining your product as accurately as possible from the beginning.

Typically, we start by brainstorming with you to gather all the main **UX/UI functionalities** and **envision the context** of your product. We will also try to **extract the key value** we want to offer to the user. In order to do that, we need to answer some questions:

- What is the purpose of using the product?
- Who is likely to use it, how often and in what context?
- What is the added value?
- What is the main content and functionality of the application?
- What are the commercial and functional goals of the project?

The “Defining” step often concludes with concept sketching – even on a piece of paper – to visualize the idea.

Step 2: Research

After we have arrived at a clear definition of the product, we will need to carry out **user and market research**. This stage has two main purposes. The first is to better understand users’ needs. The second is to become aware of the state of competition and market in general.

Basic action steps during this stage may include:

- Individual, in-depth interviews with users;
- Competition research and looking out for similar, existing products;

- Finding opportunities for product implementation.

Step 3: Analysis

After having conducted the initial research, we are beginning to understand the context in which we will be creating the product. Now it is time to organize the data we collected and lay a solid foundation for the design process.

Analysing all the information we have allows us to identify the elements of UX/UI design that are still missing and plan our next actions.

- We identify the opportunities for innovation and improvement of the UX/UI of your product;
- We define personas as references for key audience groups;
- We name users' feelings and behaviours that are mostly likely to lead to actions;
- We identify priority behaviours and, based on them, priority objectives;
- We plan a roadmap that will lead us to reach those objectives.

In the end of the Analysis step, we have a wide understanding of what will make for a good UX in your product. We also have the complete roadmap to create it. Now it is time to start designing!

Step 4: Design – low to medium fidelity

This is probably the most rewarding part of the UX/UI design process. Finally, all the concepts we have been coming up with are taking shape in a visual form! Most of the times, this "low to medium" design stage involves four sub-steps.

Note that these sub-steps might not always be linear. For example, we might go back from “wireframing” to “sketching” if we discover that something needs improvement on a very basic level.

1. **Sketching** helps to quickly come up with visual representations of the most basic functions. This step often results in a variety of approaches to a single element (or function), and allows to pick the most optimal variant in the end.
2. **Creating user flows and maps** allows to determine what actions and content are available at specific stages of the application flow. This is when we establish connections between certain elements of user experience, discover possible interactions and tweaks to them. This sub-step allows for significant optimization of the product’s usability.
3. **Wireframing** is when we set up the information structure for key functionalities, their placement and scale. Wireframes created at this stage are filled with exemplary content, and their purpose is to make sure that every function of the product is represented in a proper way.
4. **Prototyping** is the final step which lets us verify the process, check screen-to-screen interactions, and tell whether the feel of the application flow seems right. After receiving feedback on the prototype, we often go back to the “wireframing” stage to make necessary adjustments.

Step 5: Design – high fidelity

When we are happy with the medium-fidelity prototype, it is time to design the final experience for your users. The last stage of our UX/UI design process results in giving your product the definitive feel and look, which build up the experience of the end-user.

Firstly, we need to establish the Design System (DS). This involves all the colours, fonts and other visual elements assigned to your product. Having a

set of DS files is a backbone for every UX/UI project, because it allows for easy adjustments to the look of your product.

Secondly, we finalise the visual design. This means replacing basic shapes of the wireframes with actual assets from the DS and with other external components – such as custom photographs and data. This step results in the final look of the product in the current milestone, which can still be altered after feedback.

Finally, we present the complete designs to you in realistic user scenarios and on various types of hardware. This allows you to understand how the final product will look, feel and respond like.

Provisions of Our Contract

The contract we sign with our clients is a legal way to ensure that the agile process is implemented the way we picture it in this whitepaper. We want to be completely transparent with you as to what you and us are agreeing upon when signing the contract.

Here are some of our key provisions – and the explanation why they are so important in agile software development. To request a full contract template, contact us directly:

Contact us

List of provisions:

1. **Provisions describing the development process.** Both Parties (Ordering Party and Ideamotive) declare to attend all the necessary meetings and agree on the format, purpose and participants of those meetings. Furthermore, they agree to comply with the general framework of the development process, i.e. how it begins, how it is maintained, and how it concludes.

   This is the part of the contract where both Parties explicitly agree to work in agile and time & materials system.

2. **Provisions stating that the development is done in iterations.** Both Parties acknowledge that the development will be broken down into stages (or iterations) – without defining how many of those stages are needed to complete the project. The exact number of iterations can only be established during the development process. However, before

signing the contract, Ideamotive provides you with an estimated number of iterations.

3. **Provisions covering the rights and obligations of Ideamotive.** These refer mainly to Ideamotive's obligation to keep the Ordering Party properly informed. We inform you about any anticipated changes or key events in the development process, and we provide detailed documentation of our work (including documenting time & materials spent on your project). We also reserve the right to employ third parties in the project, under the condition that we take full responsibility for their actions.

4. **Provisions covering the rights and obligations of the Ordering Party.** This part refers mainly to your obligations to:
    - provide information necessary for the development process,
    - respond to questions from Ideamotive,
    - appoint people responsible for cooperating with Ideamotive in the development process.

5. **Provisions related to the maintenance of the developed system.** The contract states that any work connected to technical maintenance of the ordered product is subject to a separate contract. Development and future maintenance are two different areas of work.

6. **Provisions related to remuneration.** We will charge you one and the same hourly rate for the work of all our employees. Additionally, the contract states that all the work done by the developers is subject to remuneration – including programming, research, bug fixes and meetings. Every billing amount is accounted for in the detailed work time report that we send you each month, together with the invoice.

7. **Provisions allowing us to use some of your product information in our portfolio and for marketing purposes.**

8. **Provisions regulating potential withdrawal from agreement.** You have the right to withdraw from the contract in case of any major negligence from Ideamotive's side.

9. **Provisions related to copyrights.** We transfer you copyrights to the month's code, as well as all the other assets we create (e.g. layouts, graphics), once you have paid the invoice.

10. **Provisions related to trade secrets and confidentiality.** Both Parties are obliged not to disclose sensitive information to third parties, unless it is demanded by legal authorities. Whenever there might be doubt about whether certain piece of information is confidential, the Party which is transferring the information is responsible for indicating this.

11. **Provisions regarding the settlement of disputes.** If any disputes arise, we will try to solve them amicably through negotiations.

12. **Provisions regarding our communication.** The contract appoints contact people representing both parties and indicates the channels through which they will communicate.

Does it sound like we could create something together?

Do you have any more questions about our workflow, contract or anything else?

Would you like to schedule a free Discovery Call with Ideamotive?

Contact us now

and tell us about your idea.
We will be excited to hear from you!