

## When to Hire on Project-Based-Agile Hiring Method

We recommend project-based agile hiring when

1. You have a new software application to develop and you have the high level functional and non-functional requirements almost fixed. It may change but not too much.
2. You have a fair idea of how it will work but you still would like to explore the secondary requirements, such as screens, UX, and many hidden aspects those crop up during the development only.
3. You would like to have some sort of control over the budget and timeline
4. You can be available with the team on a daily basis during the entire period of the development

## Why hire an Offshore Developer?

1. **Reduced Cost** - If you are from a country where the cost of labor is significantly higher than in India, with us, you are likely to get better developers at 1/5th (or even less) price.
2. **Reduced Headache** - Running an in-house team does not only add 5 times but a lot of headaches too. Here you will not have to bother about the recruitment, HR activities, operational processes, office management, hardware, and software, etc. You can simply concentrate on your project only.
3. **Expert Overnight** - It takes years to build an organizational knowledge base over the software development process for an organization. We are in the industry for over 20 years. The moment you choose us as your development partner you can claim that your team is backed up with 20 years of experience.
4. **Ready Platform** - We will give you free access to our end-to-end software development cum agile team management platform. There you can see the full picture of the team activity or project development progress or software documentation required for future maintenance.

## Why not Hire a Freelancer?

Freelancer hiring often looks lucrative. But you may have to spend a lot of time getting reliability from a freelancer because of some inherent problems. Here are those.

1. **Uncertain** - Most freelancer works fulltime somewhere and you do not know when he will be suddenly off to save his permanent job. You do not know when he/she will leave. If you hire him/her even dedicated, there is every chance he will have to continue to support other clients. You do not know when he will get a better deal and leave you.
2. **Unsafe** - Your codes, data privacy, and IP rights may not be in a safe hand. There is a possibility that your code / IP rights are being re-used in other projects, particularly when a freelancer leaves. You can never catch a freelancer. An ethical company never does it who thinks long term. Employees working for a company do not dare to do it because of legal action.
3. **Limited Talent** - Software development is teamwork. Building a professional application that

will provide value to its customers requires many experts. An architect, analyst, server admin, designer, developer, test analyst, etc. It is never possible to get someone with the skills required to play all these roles. So other than developer skill you will have to compromise. With a company like us, you can hire them on an SOS basis and pay only for limited hours. In fact, for the smaller issue, the help comes absolutely free.

4. **Dead End** - With a freelancer, your growth option is a dead end. Initially, you may start with something small a developer might be apparently sufficient for you, but soon your application may become bigger and more complex. You may need help from experts or new resources. A freelancer will pull you back.
5. **No Organisational Tracking** - We have an organizational tracking system. We will do the activity tracking, work tracking on your behalf, and rectify in the background to yield greater output. If you hire a freelancer you might have to hire another manager (or spend a lot of time of yours) to track him.
6. **No Organisational Knowledgebase** - We are in web application development since 1999. We have several people who are in the organization for more than 14 years. We know more than a freelancer what works and what does not work. You will lack it when you hire a freelancer.
7. **Knowledge Transfer Crisis** - A developer may leave. Whether you hire from us or hire a freelancer. And the worst situation normally comes when he/she leaves. When a freelancer leaves you to have to manage the knowledge transfer yourself. It may not be possible for you. But we will easily manage it. The scrum master will have a high-level knowledge of the system. Plus we have a standard documentation process to transfer it to a new developer.

## Project-Based-Agile Hiring - How does it work?

1. **Initial Project Discussion** - We will discuss with you the primary product concept, business objectives, similar businesses, opportunities, and threats. We will also discuss the possible solutions, possibilities, technologies, resource type required - and try to give a rough projection on time and cost.
2. **High-level requirement document** - The next job is to create a high-level requirement document such that we can give you a high-level estimate to make the primary deal. This requirement document will consist of the functional requirements, major nonfunctional requirements, a rough sketch of the proposed system, and associated documentation. We will help you with this.
3. **MVP (Minimum Value Product)** - It may take years to develop the system if every possible feature is developed. As per lean product management dynamics approx 80% of values lie in only 20% features. So developing the lesser valued features in the early stages is simply a waste. We have to choose the top 20% features, develop it, and launch early with 80% values.
4. **Prepare a ballpark estimate and development plan** - We will give you a ballpark estimate for the MVP along with a development plan based on the initial requirement provided.
5. **Finalize the deal and sign the deal** - The next job is to finalize a deal and agree on it. The deal will likely consider the final set of requirements, the change management processes, IP rights, cost aspects, and timeline. It will also consider billings, payments, development process, involvements on both sides to avoid project delay and quality.
6. **Pay the initial deposit** - Just pay the first installment to get started with the development process. It can be as minimum as 15 days approx bill of the teamwork.

7. **Create an Agile Project Team** - To build a product that we need an agile team. To have a working agile team we need a product owner, scrum master, and a minimum of one developer. The development will be done by the developer but the requirements will be decided by the product owner. The scrum master's job is to clear any roadblocks through direct or indirect involvement. The product owner's job is likely to be played from your side. He must be available to clarify requirements points and do the acceptance testing.
8. **Setup Project Management Software** - We need to set up a team in a standard project management software like JIRA or in our own in-house agile product cum project management software.
9. **Software Environmental Setup** - The scrum master with the developer will setup software environments/technology stacks required for your development. It will consist of the local environment, DEV server at the cloud, deployment setup (GIT), etc.
10. **The Team Operation**- Both the offshore and onshore teams will work for hand in hand. However, the onshore team will concentrate more on gathering projects/requirements and development teams in the execution and defectless delivery. Anyone can be contacted anytime. The development process will be 100% transparent. Requirements, test cases, templates, system architectures, tasks, projects, work reports, retro reports, plans, estimates, progress reports, delivery reports - everything can be accessed anytime.
11. **Incremental Development** - In agile we will develop on an incremental basis. The requirement will be defined, analyzed, developed, tested, reviewed, and delivered one by one. While one feature is being developed, the product owner can define the next feature and review the features which have been submitted for UAT. The technical debt can be minimized by introducing changes based on feedback.
12. **Launch** - Launch the MVP after proper review and testings.
13. **Maintenance** - Start doing business. Change the system as per users' feedback.
14. **Next Phase development** - Try to find out the next 20% of features that cover 80% of the rest values. Then develop it and launch it. If everything is fine in two phases we will be able to deliver 96% values by developing only 40% features.
15. **Billing & Payment** - The billing can be monthly or periodically subject to milestones achieved.

## The Agile Team Roles

1. **Product owner** - Says the final word on any requirement issue
2. **Scrum Master** - Responsible to solve any problem that slows down the development.
3. **Team Members (Development Team)** - They actually executes the tasks. Contact product owner on any requirement point and contact scrum master if faces any roadblock. Developers, designers, testers, etc.
4. **Stakeholders** - They actually do not belong to the team but somehow will be benefited from the system. Example: final clients, shareholders, etc.

## The Product Development Life Cycle

1. **Concept building** - Define the value and value stream. Value means what unique benefits will cheer users and value stream refers to the processes associated with it. The product owner will define mainly on three points -
  1. what to build
  2. how will it work
  3. how it will benefit the stakeholders
2. **Prioritize** - As per product development dynamics 20% of features adds 80% values. So it is important to choose the top 20% features for phase-1 development. In phase-2 choose 20% of the rest 80% features and so on.
3. **Create the Task Backlog** - Once the top 20% features are decided the next step is to create a backlog of the tasks. The phase 1 product will be developed once tasks will be completed. All features may not require fresh development.
4. **Establish the development process & feedback loop**- Just plan how the will project is completed. How much to develop, how much to buy. Decide the technologies and platforms. Decide how the system will work and architect accordingly. Decide the tools and techniques for requirement development, modeling, testing, project management, and so on. Develop a closed feedback loop with the stakeholders/team members.
5. **Develop each in the agile process** - Simply pick a feature, develop/customize it, test it, deliver it for feedback, and improve the feature as per stakeholders' feedback. Mark them as done once accepted. Repeat the cycle until all features are done.
6. **Launch the PHASE-X** - Once all done launch it and monitor in production- preferably through telemetry. Fix/implement any reported error or improvement. A continuous improvement process.
7. **Next Phase Development** - Choose the next top 20% features for development. And repeat the cycle.

## Team Management Workflow

We have an automated agile project management system (Agile24) to manage the team management processes. We will not mind using 3rd party software like Jira too if you prefer.

1. **Create a Team** - The administrator of the project management software will do it for you. A team must have a product owner, a scrum master, and members. Members can be added or removed when necessary.
2. **Create a Product** - The product owner creates a product that is to be built. The proposed system is considered to be a product. It will be there until the product is removed from the market. The product owner defines it in terms of features and stories (functionalities).
3. **Create Projects** - The scrum master creates one or more projects to build the proposed system when the project. One phased development can be considered a project too.
4. **Create Backlog** - Once the product concept is built and the features to be developed in a particular project are decided, the ScrumMaster and product owner jointly create the backlog of the tasks that project. All stories are converted as tasks. Apart from stories, there can be other tasks too - general tasks, test runs, test suites, test runs, defects, etc.

5. **Create Sprints** - The ScrumMaster does a rough estimation of the full project and full development time is divided into manageable chunks with a fixed duration (say, two weeks or 1 month). Tasks are moved to sprints as per priority. Many tasks may not be moved will lie in the unscheduled backlog.
6. **Assign Tasks** - Now the ScrumMaster assigns tasks to members considering the availability and capability of the developers / other resources.
7. **Sprint Meeting** - A meeting is done to convey the sprint objective and decide the gameplan to complete the tasks of the sprint.
8. **Daily Scrum** - There will be a fixed time when all members of the DEV team will meet together to discuss just 3 points. What was done yesterday? What is to be done today? What are the impediments that slow the development process? The work plan of the day is planned accordingly.
9. **Execute Tasks** - The members execute their duty as planned. The product owner must be available to remove the impediments related to the requirements. Everyday a scheduled meeting with the product owner is recommended to avoid any sort of technical debt.
10. **Day end work entry** - Everyone makes work entry - what has been done today along with time to track hours to complete each task.
11. **Test, Review, and done** - The initially done work is tested by the developer first and then reviewed by the product owners/scrum master. It may further undergo testing by testers as the situation demands. Once passed it is marked as passed.
12. **Sprint Retro Meeting**- Everything cant goes well. In the sprint end, a blameless meeting is done to review what went wrong and what right. Then ScrumMaster takes action to repeats that went right and avoid that went wrong.
13. **Analyze Reports** - An automated delivery report for a specific duration available for the team/user/project to assess the ground reality. Work reports are available for further analysis. Estimate change reports are available to determine when and how a project is delayed.

## How will it cost?

The cost will be calculated on the basis of the rates and estimates. Here are the base rates

## Rate Chart: Monthly Full Dedicated

Resource Type	Experience	Monthly (\$)
Web Developer (For E-Com & Generic Websites using PHP)	2-6 Yrs	1000 - 1500
Web Designer ( Photoshop, HTML & CSS)	4-8 Yrs	1000 - 1500
Backend API Developer (PHP/Node.js)	3-7 Yrs	1200-1500

Front End Web Developer (React/Angular/Vue)	3-7 Yrs	1200-1500
Mobile App Developer (React Native)	3-6 Yrs	1200-1500
Full-Stack Developer (React + PHP/Node.js)	5-10 Yrs	1200-1800

**Note \*\*** - Cost will vary with the technology, skill level, and order volume

## Rate Chart: Hourly Rates

Resource Type	Approx Experience	Hourly (\$)
Web Developer (For E-Com & Generic Websites using PHP)	2-6 Yrs	7-12
Web Designer ( Photoshop, HTML & CSS)	4-8 Yrs	9-12
Backend API Developer (PHP/Node.js)	3-7 Yrs	10-14
Front End Web Developer (React/Angular/Vue)	3-7 Yrs	10-14
Mobile App Developer (React Native)	3-6 Yrs	10-15
Full-Stack Developer (React + PHP/Node.js)	5-10 Yrs	12-15
Routine Tester	3-8 Yrs	10-14
Load & Security Tester	3-10 Yrs	10-14
Test Analyst	7-15 Yrs	12-15
System Architect	10-15 Yrs	15
Scrum Master	10-15 Yrs	15
Server Administrator	10-15 Yrs	15
Database Administrator	10-15 Yrs	15
Data Entry Operator	4-15 Yrs	4-7

**Note \*\*** - Cost will vary with the technology, skill level, and order volume