



OUTSOURCED SOFTWARE DEVELOPMENT GUIDE FOR MANAGERS

By Nisos Health

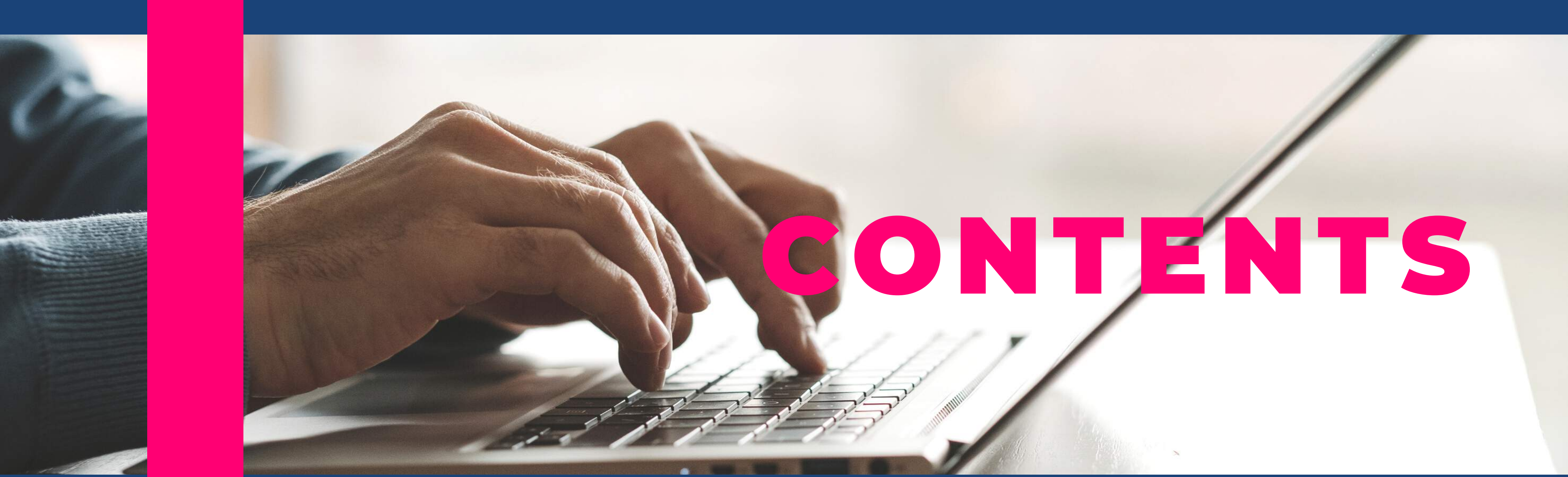
VOLUME I

Outsourced software development – Guide for managers

**2021
GUIDE**

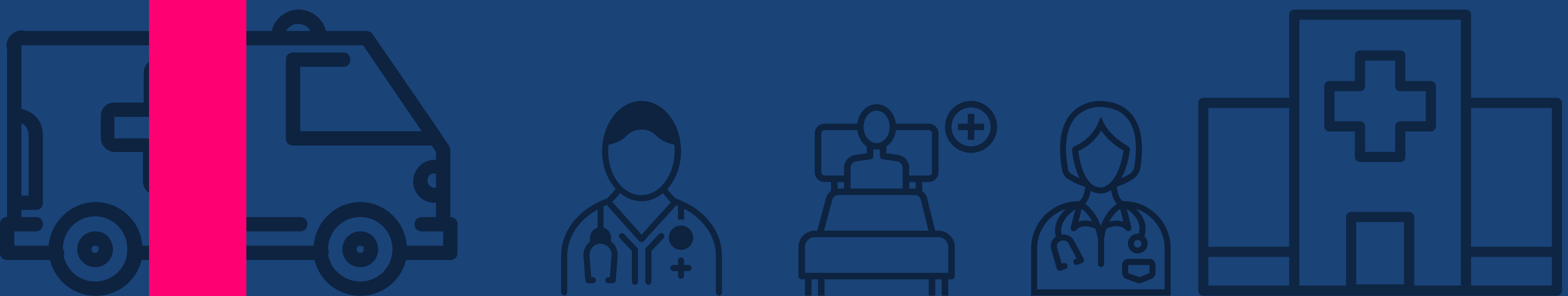
Written by Nisos Health (nisos.health)





CONTENTS

- 04** **Summary**
- 04** **Reasons to consider outsourcing software development**
- 08** **Which outsourced product development model to use**
- 10** **Should you really choose a global delivery model For Outsourced Software Development?**
- 13** **How to set up necessary processes for Outsourced Product Development?**
- 17** **How to set up a communication plan between the offshore and onshore stakeholders**
- 20** **What engineering processes should you expect in an outsourced software development partnership?**
- 23** **What project methodology to choose for Outsourced Product Development success?**
- 26** **How to plan & schedule an outsourced agile software development project**
- 31** **How to handle project management during Outsourced Product Development**
- 40** **How to set up quality assurance for your outsourced software development project(s)**
- 45** **Maintaining Quality in an outsourced product development model**



Here's a guide for managers thinking about hiring an outsourced software development company to bring their software products to market at acceptable costs / budget.



Summary

Here's a guide for managers thinking about hiring an outsourced software development company to bring their software products to market at acceptable costs / budget. WARNING – Outsourced software development or outsourced product engineering relationships are successful only when you are fully prepared for the changes in communications protocols, employee engagement protocols.

You will fail miserably if your organization is not ready to treat your outsourced software development partner as a true extension of your team. What are some pros and cons of outsourced software development / product engineering

Reasons to consider outsourcing software development

Most managers consider outsourcing their software development for one or more of the following reasons:

1. Budget – hiring on-shore staff is expensive. Offshore talent allows you to leverage the labor arbitrage.
2. Attrition – retaining on-shore staff is expensive and with 2 week notice periods, hiring replacement staff, training them, making them productive is EVEN MORE expensive.
3. Specific skill sets – building a team for a specific skill set (e.g. mobile or web or Dynamics CRM or Salesforce etc) requires you to, yourself, know enough to be able to build a team or a center of excellence around it. This is particularly hard.
4. Speeding up product life cycle – shorter term needs to speed up product development lifecycles and deliver software products to market without having to commit to hiring and retaining a team on-shore.
5. Product support – maintaining on-shore staff to support a software product that is not actively being developed does not make sense.
6. More...

Issues with outsourced software development

Outsourcing your software development essentially requires you to give up some level of control on the same. It requires you to understand that you no longer have development staff reporting directly to you, hence their behavior is also not exactly what you would expect from a development team that reports directly to you.

When you outsource software development, you do depend on the development processes, the quality control, the development methodologies that your development partner uses. This can sometimes lead to issues.

You can no longer have quick team meetings in your office to hash through discussion points. You depend on the availability of the offshore team and the communication protocols of the development partner. This forces you to change the way you work.

Business domain and specific product knowledge is something that we take for granted. This leads to the biggest issues and conflicts when you outsource your software development life cycle. A development partner might have the requisite business domain knowledge that your industry needs, but they cannot possibly have the same product knowledge that your current development team has. Most issues occur when managers expect the development partner to hit the ground running from the moment the contract starts.

Expected ROI on software development outsourcing typically leads to even bigger issues. Most managers take on outsourcing as a way to leverage labor arbitrage. However, they tend to forget that these development partners are humans as well and need just as much time as a local, on-shore developer hire would take to come up to speed on the specific product. In the first year of outsourcing, the ROI almost never matches the expectations.

Commitment to your own company – you can expect your own development team to be fully committed to your own firm (although, even that is questionable in many cases). You can expect your own team to drink the kool aid of your company, branding etc. But if you are expecting your development partner to do the same and have the same commitment towards your firm, you are in for a rude awakening.

Cultural differences – generally speaking, you know how to work within the socio-economic and cultural confines of people from your own country. When you outsource your software development to another country (off-shore), you do need to be cognizant of the cultural differences and the style of working. The colloquial language that you might be used to, might not make sense to your development partner's team at all. In the USA, you might be used to your development team speaking their minds all the time, but in India you will almost never hear a NO from your development team. These cultural differences do end up being problematic if you are not used to working with offshore software development teams.

How about some benefits of outsourcing your software development?

Labor arbitrage – yes, it truly does allow you to leverage the same development talent (and the same headaches) at a much cheaper cost. In many cases, you actually get a much larger talent pool to choose from. Read more about how much it costs to develop custom software [here](#).

Reducing human capital management overheads – In most cases, the entire hiring, training, mentoring overhead is completely outsourced to someone else. This effectively allows you to focus on deliverables to bring to the market rather than concentrating on managing manpower.

Increasing your bus factor – if you have given a large piece of your business to an outsourced development partner, they will care enough about your business to provide you with shadow resources / bench strength. We all know that developers (employees in general) can quit at any time – this is employment at will. As a development manager, bus factor is something that keeps you up at nights – especially if you do not have enough of a budget to maintain a strong bench. Your development partner should be able to provide you with the bench strength and therefore increase your bus factor without you having to pay anything extra for it.

Specific skill sets – we forget how expensive it is to hire development talent with very focused, specific skill sets.

When development managers hire folks without those skill sets, they then need to train their development team on those specific skill sets. However, this usually leads to your own development team learning on your own dime.. Being trained on the specific skill to be effective in the future. While investing in your employees is a MUST HAVE, this also delays your ability to bring products to market on time, on budget and on value. This is where a development partner can add tremendous value. It is their job to keep development talent with these specific skill sets on staff. You pretty much end up reaping the benefits of being able to go to market on time, budget and value PLUS you don't spend your budget on getting the requisite certifications (these are typically not very cheap to obtain).

Going to market faster – Hiring on-shore talent is not cheap. While your CEO might be pushing to take a product to market faster and even gives you the budget to do so.. It's not always easy to hire, train, onboard on-shore talent WHILE you are developing, supporting existing software products. This is where outsourced software development partners can help – they offload the entire norming and forming phase of an outsourced development project.. Something that you simply do not have time to do.



Which outsourced product development model to use

There are various models to consider while benefiting from outsourced product development.

Full fledged product development

– wherein your outsourced software development partner develops your product lines from soup to nuts. This is beneficial because you cut down on the coordination overheads a lot. However, be careful with this as you effectively give up all control over your software development. This model works well when you do not have a strong dev team in house. This model could also work very well when you have at least 1 very senior person (e.g. a CTO) that can oversee overall software development life cycle, the quality of software development, quality assurance processes and set the direction as needed.



Co-product development

– wherein your outsourced software development partner works with your product engineering team to develop faster, across time zones. This works very well but comes with the overhead of communications and coordination. A very good way to achieve success in this model is to separate responsibilities distinctly.

– e.g. backend vs frontend development or mobile vs web development. This kind of separation of duties allows you to go to market a LOT faster than you traditionally could, with your own development team in a single time zone.

Modular product development

– wherein your outsourced software development partner develops specific parts/modules of your core product set. This works well but definitely has the overheads of communications and coordinations. However, just undertaking this initiative in itself matures your own software development organization's processes and methodologies. If you have a mature development team, this could work very well. Good examples of such development are wherein your team develops the core product while the outsourced development partner develops the integration module (e.g. EMR integration). EMR integration is a core part of most medical software but the skill set needed to develop EMR integrations are not very easy to find. This, typically, is a great candidate for modular product development.



Peripheral product development

– wherein your outsourced software development partner only develops assistive products that support your main product line(s). Many outsourced software development agreements start this way, where the development managers first test the working relationship with a vendor by handing over development of assistive products only. Since the roadmap of assistive products do not coincide nor clash with the product life cycle of the main product set(s) of your firm, this is a low risk way of testing out the working relationship with an outsourced vendor.

A photograph of three people, two men and one woman, looking at a computer screen in a dimly lit office. The man in the foreground is wearing a striped sweater and has a beard. The woman behind him is wearing a yellow top. The man to the left is wearing glasses and a dark shirt. They are all looking at a computer monitor which is partially visible on the left. The background is dark with some blurred lights.

Should you really choose a global delivery model For Outsourced Software Development?

When you choose to leverage the benefits of outsourced software development, your vendor or technology partner typically would offer a few global delivery models. Before you choose to outsource your software development processes, here are a few tips to understand the various global delivery models and to figure out what works best for you and your organization.

Outsourcing Delivery Models typically follow any of the approaches suggested below:

1. Time and Material Outsourcing
2. Project-Based Outsourcing
3. Dedicated Development Team
4. Managed Services Model

Time and Material Outsourcing (T&M)

The time and material (T&M) outsourcing approach is suited to a scenario where you want to initiate the development or work right away, but you are not clear about the entire scope of work.

The T&M model facilitates agile development process as

- It is a flexible model with negotiable budget and low risk for both the parties.
- Opportunity to pay in parts only for the performed work, perfect for long projects for which the timelines cannot be predicted during the initial stages.
- You have control over the project and tasks.
- Enhancements, changes and negotiations can be done during any project stage.

Project-based Outsourcing

The Project-based outsourcing approach involves complete outsourcing by you to a technology partner/vendor. The Project Management for the entire project is taken care by your technology partner, while you, the customer, are involved in sharing the requirements, and acceptance of the project deliverables. The project based outsourcing approach has the following benefits for you, the customer:

- Availability of skilled resources, which may not be easily available with the customer. This allows the customer to focus on the core mission in hand, and also save on employee hiring, training, and associated costs.
- Improve efficiencies, focusing on core processes and eliminating inefficiencies by outsourcing non-core processes.
- Better provision of risk management as the customer can outsource the associated risks thereby, spreading and reducing the risks.
- Offshoring to a different country, which is in a different time zone is useful as it helps to utilize the 24 hours available more effectively. As the partner's day is the customer's night, partner can take over from where the customer left working during the day, and continue working during the customer's night.

Outsourcing enables staffing flexibility and helps in meeting uncertainties in demand.



Dedicated Development Team

In this model, a dedicated development team is allocated to work with you full time, exclusively on the projects on a long term basis. Your technology partner is responsible for setting up the software development team mapping to your project/product's skill requirements, workflows, methodologies, practices, management style, and business culture.

The dedicated team is assigned to projects by your team and is managed by your representatives. You are charged a fixed monthly fee for each employee and the workload for each employee is maintained by your team.

A few benefits of the dedicated development team model are:

- Initial investments are not required on the long-term, pricing is usually flexible and therefore, return on investments are quick.
- Full Operational control with client and therefore, greater flexibility in operations.
- Readily available infrastructure and facilities.
- Flexible and consistent team.
- Seamless extension to your workflow and processes – ease in team augmentation without hiring on your side.
- Strong service delivery expertise from your technology partner.

Managed Services Model

In this model, your technology partner, as a service provider, becomes responsible for the end to end delivery of a service and operations which was earlier being maintained by you, the customer. While you are responsible for only managing, or monitoring the engagement with the service provider, and evaluating its performance, you are not required to engage in planning, designing, managing and delivering the services. These activities are the responsibility of the service provider/technology partner.



The different types of Managed Services Engagements are:

- Team Augmentation
- Tasks and services with specific service levels
- Complete IT operations and end-to-end service level management
- Hosting and IT operations with end-to-end service level management
- Complete outsourcing, including all assets, planning and design with end-to-end service level management

Few key benefits of a managed services model are:

- Risks, rewards and responsibilities are shared between the trusted partners.
- Predictable costing
- SLA based deliveries – Service Level Agreements based SLAs set.
- Freeing up of your resources for taking up other important activities.
- Clearly stated, measurable outcomes using metrics.
- Consistent high quality delivery.
- Continuous improvement, using and improvising upon shared knowledge of best process practices, standardizations, and application life cycle improvements.

Which model you end up choosing depends on your particular situation. While global delivery models for outsourced software development bring in a lot of benefits, they also introduce risks, communication and coordination overheads. Think through these challenges carefully before choosing a global delivery model.

How to set up necessary processes for Outsourced Product Development?

Aligning processes with your company's strategic goals is key to achieving operational success. Process Architectures are designed and implemented considering organizational goals in mind. When you are handling a large scale outsourced product development effort, ensure that your process owners or managers are educated to appreciate the relevance of these processes and manage them effectively.

The process areas include inter-project activities with respect to definition, planning, implementation, deployment, and monitoring processes, as well as measuring, controlling, appraising and improving processes.

The five relevant process areas are:

1. Organizational Process Focus (OPF)
2. Organizational Process Definition (OPD)
3. Organizational Training (OT)
4. Organizational Performance Management (OPM)
5. Organizational Process Performance (OPP)

Basic Process Management Process Areas

The first three process areas namely Organizational Process Focus (OPF), Organizational Process Definition (OPD) and Organizational Training (OT) are basic process management process areas as they provide the ability to document and disseminate best practices, process assets related to the organization and organizational learning.

Organization Process Focus

This is focused on helping to plan, implement and deploy improvements, based on an analysis of current weaknesses and strengths of the processes or process assets. The processes can be improved based on feedback from other sources, such as Project Management, Support and Engineering groups. Process improvement proposals based on process appraisal, process measurement results, lessons learned and product evaluations help in improvement of processes.

Any effort for improvement of processes should be carefully planned, and a Process Improvement Plan is to be drafted before any such plan is updated. A process improvement plan, which addresses relevant areas of appraisal planning for processes, process action planning to create the action plan, pilot planning for validation of process implementation and deployment planning is to be envisaged.

The results of such planning are process action plans, which document how the implementations for improvements would be taken care.

Organization Process Definition

This process area sets and defines a set of standard processes including work environment standards, and assets based on objectives and process needs. The other assets include lifecycle descriptions, tailoring guidelines, other data and documentation related to processes.

Ensure that your project tailors these set of standards so as to create the projects' defined processes, while the other assets support the tailoring as well as the defined processes.



Organization Training Needs

This, on the other hand support the strategic training needs, as well as tactical training needs relevant to project teams and support groups. The training plan typically includes a managed training development program, documented plans & staff with the knowledge and skills to make the trainings effective.

Advanced Process Management Areas

These process areas help your organization to achieve improved capability to achieve its quantitative goals. Each of the advanced process areas, such as Organization Process Performance (OPP) and Organization Performance Management (OPM) have the ability to develop and deploy processes and supporting assets, and this ability is provided by the Basic process management areas.

OPP derives quantitative objectives from the organization, whereas the organization shares baselines, models related to process performance and common measures.

In OPM, process performance models and baselines are analyzed to understand ability of the organization to meet business, quality and process performance objectives, which in turn enables to have innovative and incremental improvements enhancing overall performance.

The selection of process deployments is to be purely based on the benefits and predicted costs of improvements. Causal Analysis is a supporting process which is executed in order to find the root causes of weaknesses and suggest process improvements.

Process Reviews / Meetings

A few reviews or meetings that are to be performed to improve the processes are:

Monthly Project Quality Audit:

On a Monthly basis, each project, which is under execution undergoes an internal audit by a Quality Representative, who reviews whether the project follows the necessary standards as desired by project owners. The Project Leader or Project Manager from each project represents the project. A report is prepared and shared with the Business Leader and Quality Head. This report has details about non-conformances of the projects, performance improvements of the projects for the last quarter. Also, it has observations from the project teams regarding any process improvements that could be useful.

Monthly Quality Improvement Meeting:

The Business Leader conducts a monthly Quality Improvement Meeting with the Quality Head, Quality Representatives, representatives from project team (project manager / project leader), support or engineering team representatives to review the non-conformances, issues if any, dependencies and suggestions for improvements. Suggestions accepted by Business Leader are considered for process improvements and once approved, become part of the Process Improvement Plan.

Process Improvement Meetings:

These are meetings in which the teams responsible for process improvements meet on a periodic basis. These meetings are conducted by the Process Team Leader for Process Improvements. Depending on the urgency of the improvements required, the meetings are organized as daily scrums and/ or weekly review meetings. The progress of the process improvements are reviewed here, dependencies like hardware, software, resources required, etc. are discussed and usually resolved upfront. If there are some escalations required, such escalations are shared by the Process Team Leader with the Business Leader. These meetings could be attended by the Quality Representatives, Business Leaders, Support / Project representatives, if they are specifically required for the same.

Quarterly Process Review Meeting:

This meeting is conducted by the Business Leader to review progress of process management initiatives, note the improvements in processes, take actions to remove impediments and take corrective actions wherever needed. The action items are formalized for the next quarter and are considered for review for the next Quarterly Process Review Meeting.

How to set up a communication plan between the offshore and onshore stakeholders

Communication and coordination between your offshore development partner and your existing business users + development team is usually the biggest hurdle to cross.

It is crucial to have an offshore coordination communication plan. This plan should detail the interactions that the offshore software development team members have with the onsite team associated with the software product.

At a bare minimum, the following are requirements for the plan to work effectively:

1. A list of important and relevant contacts, and client contacts with their phone numbers, email IDs.
2. The planned communication methods and techniques
3. The templates to be used for planned communications.

Various kinds of communications

The Software Offshore Development Communication Plan should list out the different types of communication:

- Weekly status reports shared by the Project Manager, in a Weekly Status Reporting format
- A Status Meeting on a weekly basis with your team. This could be face to face meetings with predefined agenda if you have any onsite talent from your offshore development team. If onsite meetings are not possible, remote meeting tools like webex, or skype help in the process
- A daily touchbase meeting / call through teleconference / skype / internet messenger, usually early in the day, which includes the offshore team as well as the onsite team / your team members. The offshore team members usually share the work done the last day, the work planned for the day and the immediate future. Any issues or queries are discussed upfront so that your project manager can respond to such queries. Also, your development or project manager can share feedback and any comments so that they can be addressed right away.

- Ongoing planned meetings conducted by project managers or business analysts on either side to communicate about application development issues. These can be conducted by teleconference, or webex or on skype on an as needed basis and used to obtain more information on software development issues on an ongoing basis
- Working sessions conducted by Lead Designers using skype/ teleconference / Internet Messenger Meeting/ video conference on an as needed basis.
- Project announcements from time to time by the Business Analyst or the Project Manager and used to inform general information about the project.
- Monthly Progress Reports shared by Project Manager to the stakeholders by the end of month. This can be from project managers from either side of the pond / either teams.
- Monthly Progress review initiated by either team. The review should be done once a month.
- Quarterly account review meetings – this should happen every quarter to ensure that your in-house development team, your project sponsors and the development partner are happy and moving in the right direction.

Channels of communication can be both formal and informal. While the formal modes discussed above are effective, and ensure that only necessary information is filtered up and top management does not get bogged down with unnecessary details, they are also time consuming and may affect decision making if the vital information does not reach the concerned.

Therefore, informal channels of communication like chat services are used, which when used judiciously can be useful to pass vital information to the stakeholders, and at the same time, reduce time taken to share the information.

The planning for channels of communication could also vary based on multiple factors – project management methodology used like SDLC or Agile, size of the engagement, organization maturity for both the client and the offshore partner, complexity of technology or domain, and time zone differences, etc. The communication plan includes important constituents like type of communication, frequency, initiated by / run by, description of the agenda, invitees to participate in the same and the duration in case of a meeting.



Communication best practices For Outsourced Agile Projects

Communication is as vital as other processes, as it can deteriorate unless enforced and controlled. Ad-hoc changes or cancellations in meetings due to personal inconvenience is a great example of this. Such changes not only affect the culture of the company, but may also adversely affect the timeline of the project when accrued altogether. Make sure your technology partner understands the importance of communication in offshore coordination and diligently works on best practices to improve such communication.

Some of the best practices that help in Offshore Communication and Coordination are:

- A dedicated offshore project manager with experience, skills and expertise in PM / PMO activities, processes such as SDLC, domain knowledge / understanding of the offshore team and processes is on-boarded to the project during the inception stage.
- A well-defined Steering Committee / PMO structure to govern the project.
- Audit groups / third party auditing, for monitoring that processes are being followed.
- Project dashboards and collaboration software for ensuring good communication within the team.
- Time tracking and monitoring for resource loading, and taking action when some team members are insufficiently loaded.
- Use of IM tools with VOIP facilities such as Skype, which help in collaboration and instant access to resources.
- Open group chats using Skype or other IM tools.

What engineering processes should you expect in an outsourced software development partnership?

The Engineering process areas cover the development of any product or service in the development domain (e.g. software products, services, processes, etc.).

It caters to five process areas of work:

- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation

Requirements Development

In the Requirements Development process area, your requirements are identified, and the needs are translated into relevant product requirements. A high level conceptual solution is worked out from the set of requirements. This set of product requirements is further divided into requirements for different product components.

Other requirements, which define the product are determined and assigned to the product components.



This set of product component and product requirements, often referred to as non-functional requirements, consist of product's performance, design features, quality attributes, performance requirements, verification or validation requirements, etc.

The requirements developed are shared with the Technical Solution and Product Integration teams.

Technical Solution

The Requirements development team provides documented requirements to the Technical solution team.

The requirements are further converted into the product architecture, product component designs and product components by the Technical Architect, supported by the designer and the developer.

The Technical Architect uses different ways, like architecture & design documentation, design tools (DB and code, mock UI), prototyping and code components or structures to explain the Technical solution.

The technical data packages are developed as part of this process for product components, which can be used by the Product Integration team or by third party vendors, who help with the Integration requirements.

A few different or alternate solutions are reviewed to select the optimum design based on established criteria.

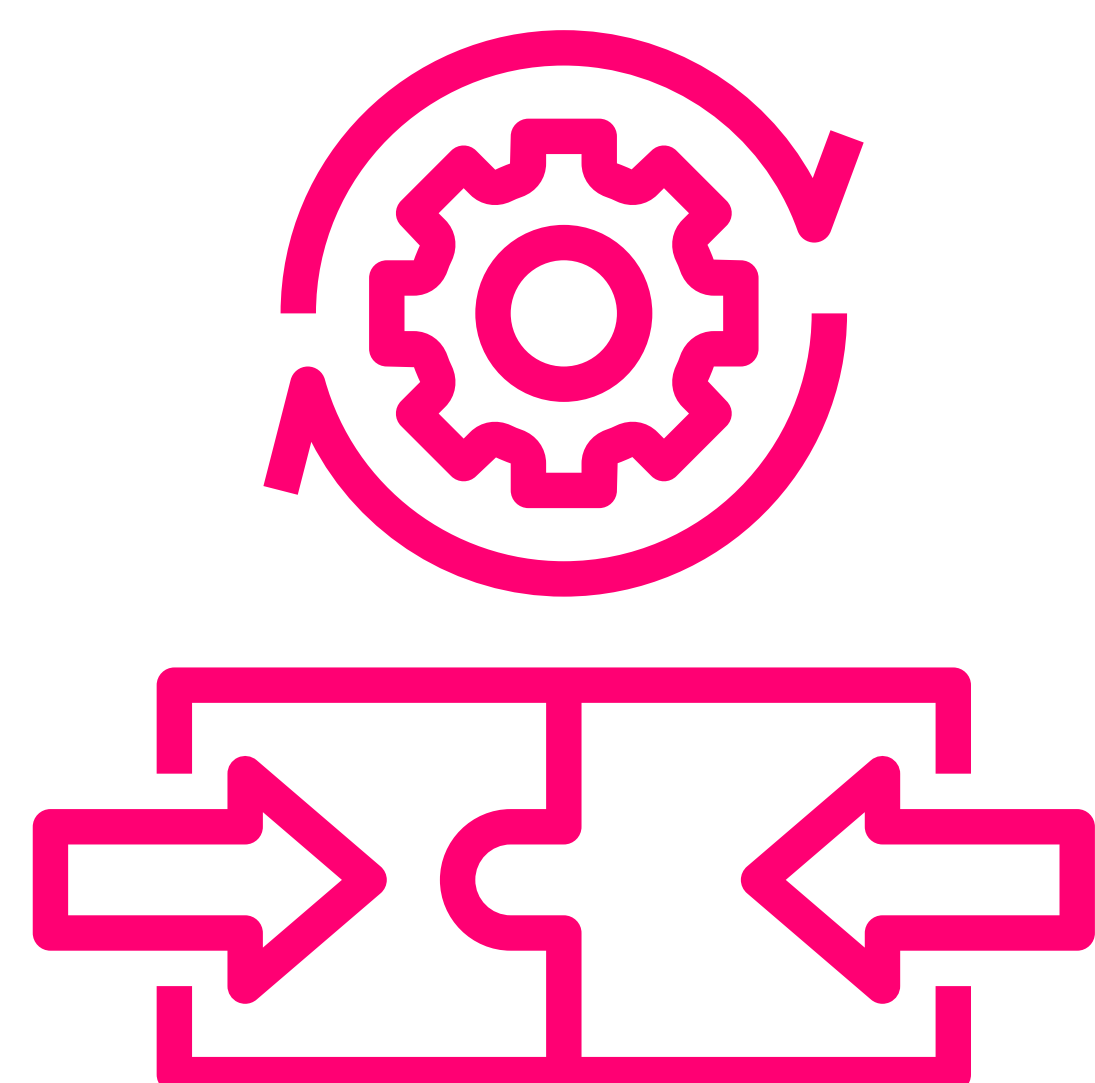
The criteria could vary based on the type of product, and therefore, there can be significant differences across products, depending on such criteria like operational environment, performance requirements, support requirements and cost or delivery schedules.

Product Integration

The Product Integration team is responsible for integrating the different components of the product.

The artifacts related to Requirements developed act as inputs for this process. The team verifies the different interfaces so that they are developed as planned in the Interface Requirements document.

The Product Integration area contains practices associated with generating an integration strategy, integrating the components of the product, and releasing the product to the customers.





Product Verification

The technical verification acts throughout the design and development in an incremental fashion, starting from product component verification and usually concluding with the verification of completely assembled products.

This process area includes preparing verification plans, performing verification, and identifying corrective actions.

The Technical Solution team relies on specific practices in the verification process to perform design verification and peer reviews during design until the final build. The appropriate verification mechanisms for the work products are determined at this stage, so that the work products meet the specified requirements.

Conducting Peer Reviews is a verification mechanism often used. Peer reviews are found to remove defects early, and provide valuable insights into the work products or work product components which are being developed.

Code reviews or unit testing can be important peer reviews at this stage.

Product interfaces verification is an important activity which is carried out prior to product integration.

Product Validation

Product Validation includes validation of products, selected intermediate work products, product components and processes.

The elements validated may undergo re-verification iteratively. Therefore, incremental validation of products against customers' needs is carried out.

Issues arising out of validation are addressed during Requirements Development or Technical Solution design.

Product validation can be done on both Operational Environment or on a simulated Operational Environment. Your team(s) coordination on the validation of requirements is an important aspect of Validation.

What project methodology to choose for Outsourced Product Development success?

Product methodology for outsourced software development typically varies across various technology firms. Firms usually adopt different project methodology frameworks based on their customers' needs. The way you work and succeed will largely depend on how your current team works.

Usually, these methodologies are iteration based – either in Agile Product Development Mode, or in Iterative Incremental Prototype Mode based on Rational Unified Process.

In an Agile model, the scope section of the proposal is based on inputs from Product Manager, which gets updated with each iteration, while in the Iterative Incremental approach, sufficient detailing to scope is done based on a prototyping approach.

The steps in the Agile model are incremental right from scoping, design, and development and divided into number of iterations, and scrums.

Just because you've heard great things about agile software development doesn't mean that it is right for your organization or your engagement with your development partner. Take a deeper look within your own organization to figure out what works best for you. You could do very well even with a waterfall model.

Success is not defined by the project methodology chosen but rather by your commitment to coordination and communication over a product's development lifecycle. It is governed by your working relationship with your outsourced development partner.

Get this right and you will reap the benefits of outsourcing your software development in a big way.



How to do agile development within a fixed budget scope?

We all know that agile development methodologies have become an alternative.. wait no.. THE alternative to traditional project management. This methodology focuses on developing software in an incremental and iterative fashion where design, implementation, requirements and testing continue throughout the project life cycle.

Meanwhile, fixed bid projects are a part of the culture of many companies. Many hold the opinion that agile development cannot be employed in fixed bid projects. Yup, we had scratched our heads with this one as well..But the reality is that you can implement agile development in fixed bid projects provided there is a variable scope. (Fixed Cost, Variable Scope).

Here's how you can still do agile development in fixed bid projects with your development partner

Defining Product backlog size before starting the project

It is necessary to analyze enough user story (some call them features) details before estimating the project. It is necessary the programming teams collaborate with the customer to get better understanding of the user stories. Both parties should reach a mutual agreement on "Definition of Done" for individual user stories, sprints and releases. You would typically nominate one person as the product owner from your development partner organization – that interacts with the product owner from your side. The entire development team considers the development partner's product owner as the "client".

Jot down the user stories with your development partner team AND agree on success criteria before even beginning a development sprint. Chasing a moving target isn't good for either team and only leads to frustrations and unmet expectations. So, while a sprint is going on, hold steady on the stories.

Use MoSCoW principle to prioritize

The team uses/should use MoSCoW principle to prioritize features that need to be included. In general the “must have” features should not exceed 60%. It is a good idea to include healthy number of “Should Have” and “Could Have” features to ensure scope can be varied. It is necessary to have clarity on the minimum scope for the product to be useful.

Move from Fixed Scope to Variable Scope

Agile development, of course, calls for variable scope in fixed bid projects. For example, if the project has 2000 story points, the variable scope will allow the team to replace existing user stories with new user stories that are somewhat similar. While doing so, the accepted change will always be within predefined boundaries that is 2000 story points.

Use Exchange model to accommodate changes

Agile development is feedback driven. You come up with new requirements due to changing business concepts which leads to change. The new requirements are translated into user stories. In agile development, you can adopt an exchange model for user stories in the product backlog. In this methodology, your team can introduce new user stories of high business value and remove user stories of less business value provided the team has not started working on the user story. Adopting the exchange model ensures the product backlog remains constant despite introducing new features at any stage of the development cycle.

Setting right expectations at the beginning

In agile development for fixed bid projects, you need to set expectations around satisfaction right from the beginning. The Agile Product Owner should understand their responsibility towards prioritizing the backlog. As an owner, he/she needs to have a vision for the product. The Agile PO should be available to answer any questions from the team and also actively participate in sprint review. The product owner needs to be business savvy as he/she is the person who is taking decisions on what features the product will have.

In a fixed bid project, the schedule and the cost can be regulated but the scope greatly depends on the customer influence and hence it need not be fixed. The best practices mentioned above sheds some light on how to do agile development in fixed bid projects. Agile fixed bid projects can be successful and feasible if the development team defines product backlog, leverages MoSCoW technique and implements user story exchange model.

How to plan & schedule an outsourced agile software development project

Large Outsourcing Product Development engagements usually involve multiple teams at onsite at client location, and one or more offshore units. Applying agile methodologies to such engagements can be challenging if not applied with an appropriate planning framework in mind.

Mature software development firms use an appropriate planning framework consisting of 5 levels of planning. The components of the planning framework consist of:

- Product Visioning
- Product Roadmap
- Release Plan
- Sprint Plan
- Daily Commitment

Product Visioning

This is a broad picture about the product as envisaged by the Product Owner. The Product owner explains how the product would look like, the broad features, the priorities and goals of the product.



Product Roadmap

With customers demanding more frequent changes and releases, the time to market is shrinking considerably. The Product owner therefore requires to think the entire development in terms of shorter steps leading to a road towards the final product. A Product Roadmap is created by the Product owner and conveyed to the development/ delivery team.



The goals of the product roadmap are:

- Communicate the whole
- Determine and convey when releases are required
- Determine that the release includes appropriate functionality, each of these can be a set of features which can be clubbed together to form epics.
- Focus on business value planned from the releases

The delivery team will on the other hand realize benefits as below:

- Understand the product vision
- Learn the steps to realize the vision
- Learn the priorities in business
- Provide technical inputs for the roadmap
- Provide estimates for the projected features

The Product Owner would define the Product roadmap, using a series of meetings with the development / delivery teams. A list of desired features are also built by the Product owner which is called Product Backlog. The list needs to be prioritized in discussion with the delivery team by the Product Owner, so that features can be combined into a release. Besides functional features, the backlog would also include the non-functional features, i.e. the technology features.

Release Planning

A release consists of a set of product increments or features that is released to a customer. A release is done at a System or Program Level and consists of a number of iterations. Releases are shown in the Product Roadmap with a date, time and planned feature set. Prioritized Product Backlog serves as an input for Release Planning. A fixed release dates are assigned for all teams of the program with a typical interval of two to four months.

The iteration on the other hand is defined at the feature level. The delivery and product owner agree on the number of iterations as part of product release.

Release planning is highly collaborative or interactive in nature and could use tools like sticky notes, flipcharts and whiteboards. This involves the Product Owner for defining user stories, setting priorities and mapping stories to iterations; and Developers, who estimate stories, point out significant technical risks and establish velocity, which is a measure of team development capacity.

A typical Release Planning meeting would have the following agenda:

- Introduction, agenda, goals and updates
- Explanation of the Product Vision by the Product Owner
- Time scheduling / time boxes of the releases and iterations
- Capacity analysis by delivery team
- Agreement on completion of deliverables
- Movement Plan of features from backlog to iterations
- Determination of dependencies and resolution of dependencies by alternative iterations.
- Final estimation of workload per iteration and comparison with the available capacity.
- Review of discovered risks and issues
- Retrospective of the session.

Iteration/ Sprint Planning

A release consists of multiple iterations or sprints, and every iteration is therefore more granular than a release. The responsibility for Iteration Planning rests with the Scrum Master or Project Manager and the Product development Team. During the session, or before the session, details is added to each feature by dividing each of them into tasks. The core activities in Iteration Planning include:

- Determination of actual capacity, or amount of work that can be done within each iteration.
- Decomposition of features into tasks and fitment of tasks to iterations.
- Every task is separately estimated, a typical task could be anywhere between half a day to two days in effort.
- A feature is not done unless it is fully tested and confirmed by the product owner. Therefore, frequent demos to customer / Product Owner is done to ensure that product development is as per expectations. Any deviations are corrected and sprint plan is altered after confirming the change from customer or Product Manager.
- Each Sprint would necessarily consist of Analysis, Design, Build, Testing, Sprint Retrospective and Sprint Review activities.

It is coordinated by the Scrum Master / Team Lead. All the stakeholders of the development like Product Owner, Team Lead, Team Members, Senior Manager supporting the project, and representative groups from all the groups like Tools, Technology, IT, Specialist Teams such as DBAs, Tech writers or UI designers, QA/Testing team, any additional supporting personnel (like Finance, HR, etc.) whose potential support is required need to participate in the meeting.

Team Composition & Engagement

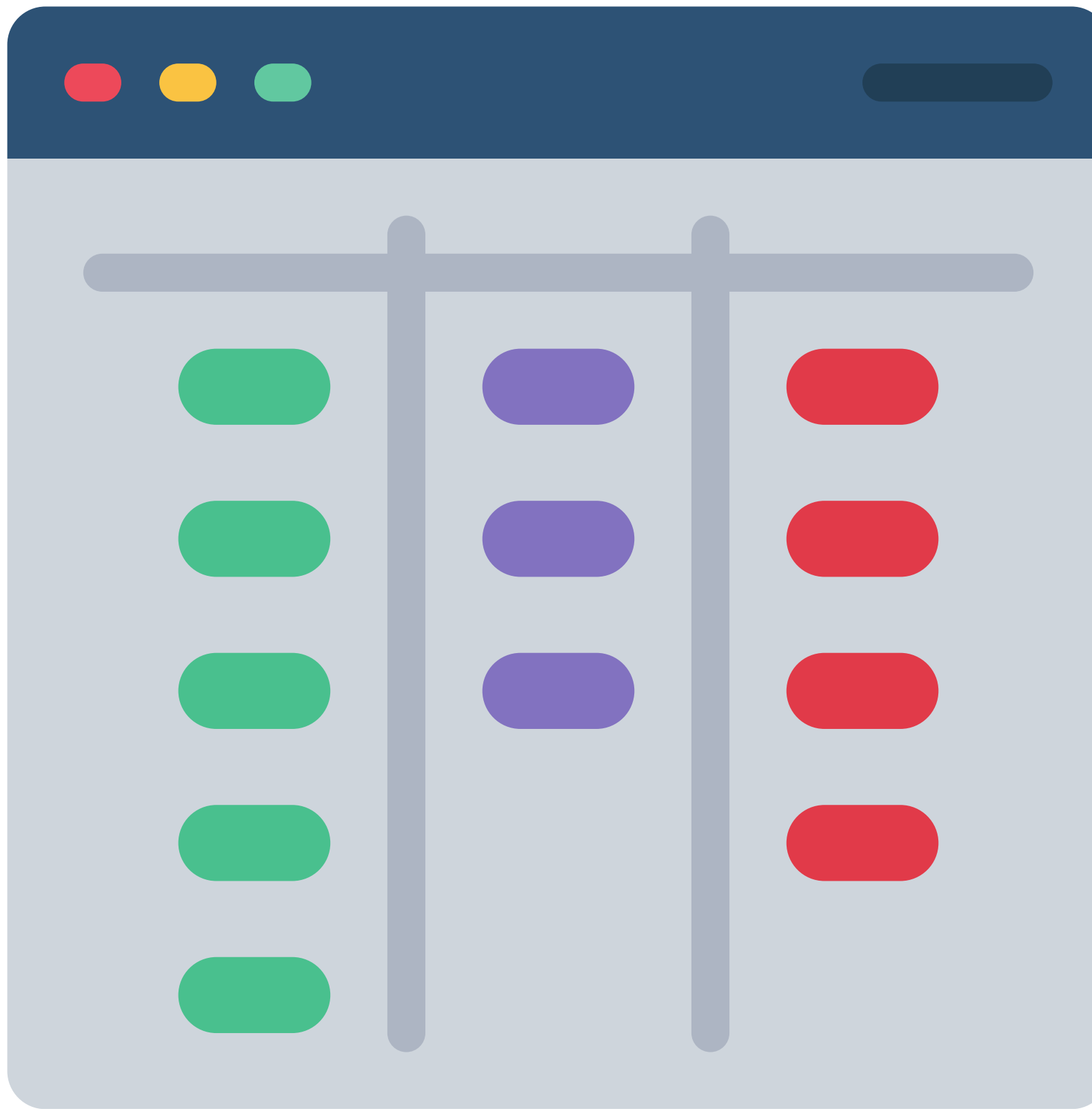
Do not discount the importance of having a product owner on both teams. This truly does help ease the outsourced software development engagement.

The Outsourced Partner at your outsourced software development firm would be at a remote location having the following members ideally:

- A Senior Manager representing the overall team and though not part of actual execution would be indirectly supporting the project and help in streamlining the work or removing obstacles in the process. He would also be involved in the project commercials and engagement with the customer.
- The offshore engagement would be led by a Team Lead / Scrum Lead, and supported by team members consisting of developers and testing team.
- In addition, the team would be supported by a Tools & Technology Group / IT Support, Specialists like DBAs, Tech writers and User Interface designers and additional developers & testers as a backup.

Project Schedule

An overall Project Schedule is developed based on the sprints planned. This creates better visibility and overview of the features associated with each sprint, the priority level, the estimate in story points, the dates planned, duration in days and the state of each task / feature whether completed. This can be used for planning and tracking the project. Any Project Management / Scheduling tool can be used for this activity.





How to handle project management during Outsourced Product Development

Project management during outsourced software development / product engineering projects is not easy. Depending on the size of the project you work on, there are several aspects you need to ensure you and your development partner handle.

This is a general guide based on our experience handling formal, large scale outsourced software development projects.

We can pretty much guarantee you that your success depends on how well prepared YOU and YOUR development partner (vendor) are.

Both you and your development partner team need to understand that projects are designed to create unique products, services or results and are temporary in nature. The requirements for projects also undergo progressive elaboration, and all of these create a level of uncertainty in projects. The strategic relevance of projects cannot be undermined as they often address organizational or customer goals, which may not be possible within the normal working of the organization.

A broad Project Management Framework is therefore the philosophy guiding projects through the stages of initiation, planning, execution, control & closure. This framework is designed to encompass different methodologies and life cycles of project execution such as iterative, agile, prototyping, spiral or waterfall models. The system based on the framework consists of a set of techniques, methodologies, procedures, tools and resources to manage a project.

Project Initiation

Initiation of a project involves a formal authorization to start a new project. Usually, this is preceded by other activities outside the boundary of the project, such as initial studies of feasibility, pre-sales and sales processes, a proposal is submitted with initial assumptions, dependencies and constraints, and an approved Purchase Order or Contract with Statement of Work (SOW) is received.

Project Initiation involves a high level definition of project, including capturing the business need / justification, initial identification of resources used including the project manager, the project methodology and system to be used. The information is captured by the Project Charter, and on its approval, the project gets official approval. Wherever needed, expert judgement or feedback from consultants, stakeholders including customers or sponsors, etc. are factored into the Charter. Approval and funding of project is usually done outside the project. A preliminary definition of initial scope is recorded here to initiate the project.

Project Planning

An Integrated Project Plan is developed to plan different processes that a typical project undergoes. During Planning, detailed steps for a project management methodology is confirmed, such as agile development, or a waterfall model, etc. Planning typically requires a project management system or tool for collaboration and feedback as the project plan is developed. A few tools which help the process is a Configuration management tool for version controlling and a Change Control System for managing changes.

The typical process areas in a Project Plan are:

Requirements management plan

- This plan indicates the scope definition, verification and control mechanisms, and depicts how the work breakdown structure would be created.



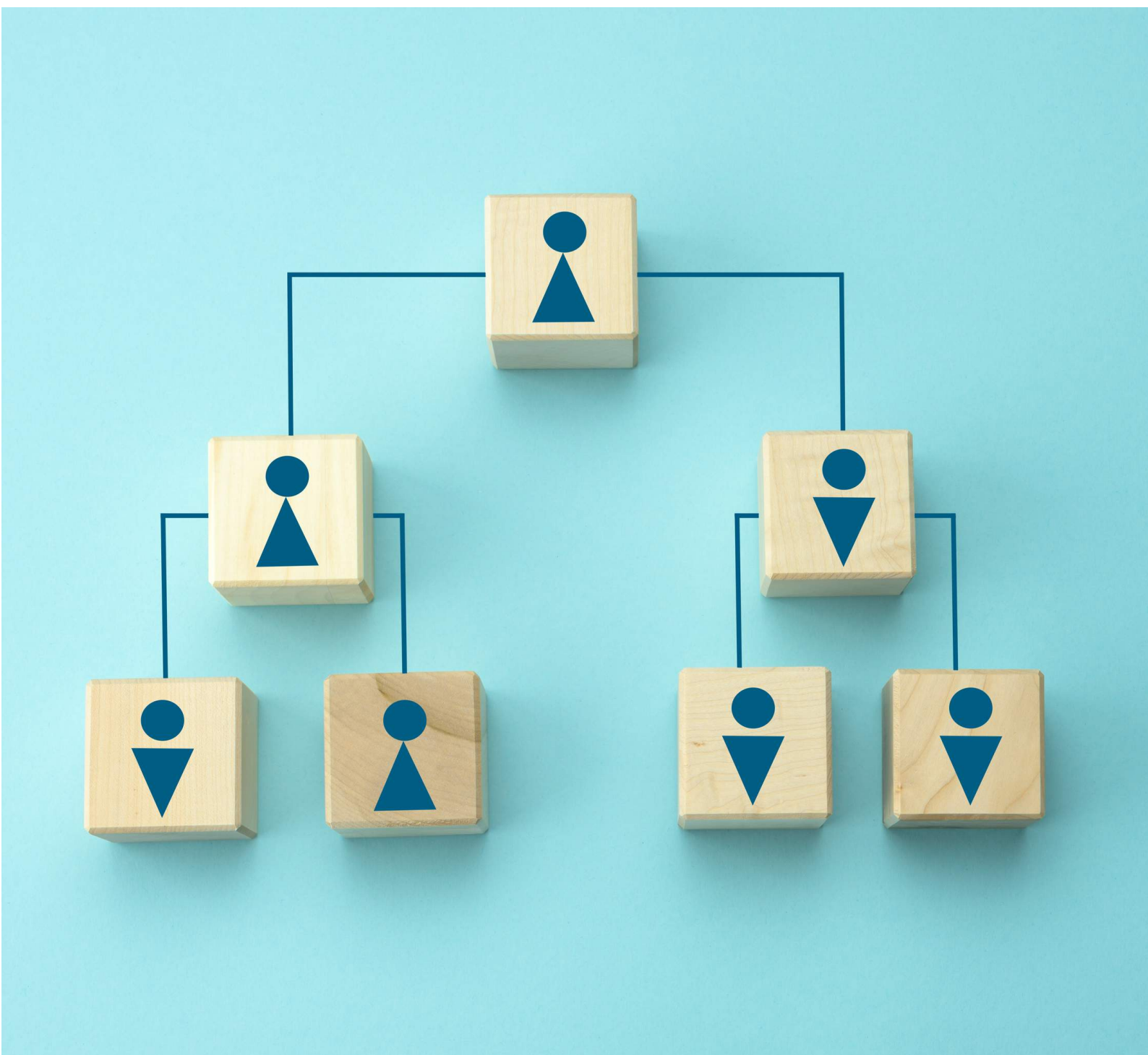
Requirements Definition

- This involves creation of a detailed scope statement with the major objectives, description, project boundaries, acceptance criteria, assumptions, constraints and deliverables, which help in executing the project in the future.

Detailed analysis of product with techniques such as product breakdown, value engineering, functional and value analysis, systems analysis are used for Requirements definition. Alternatives are identified by techniques such as brainstorming, expert opinion and lateral thinking. Moreover, needs and expectations of stakeholders are analyzed to arrive at the exact definition of requirements.

Work Breakdown Structure (WBS)

– The major project deliverables are decomposed hierarchically into smaller components till work package levels at the lowest level for better control and assigning to the team members. Each work package level represents a service, product or result that can be independently verified. WBS templates from previous projects can be useful for working out the breakdown structure.

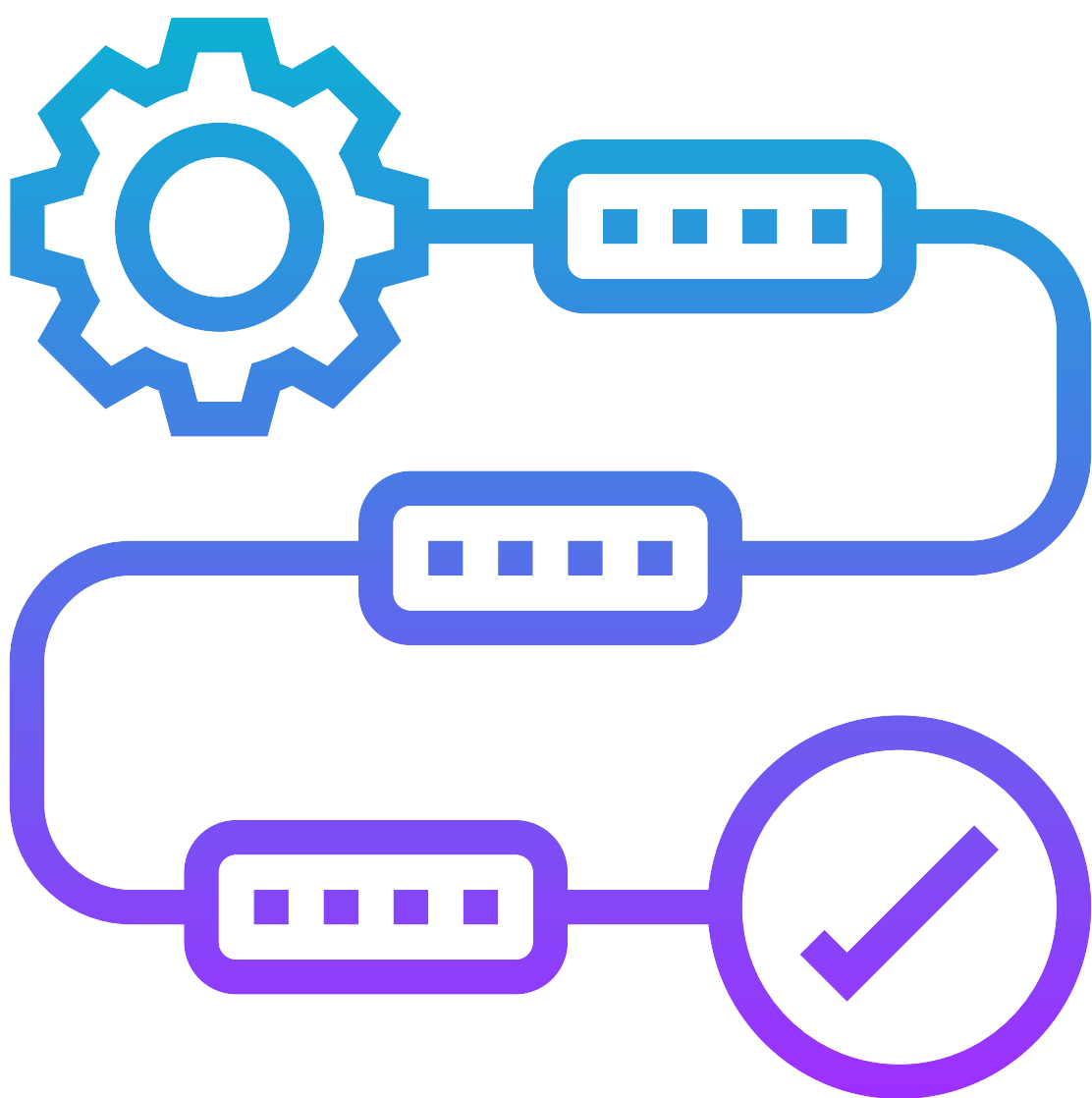


Task Definition

– Each component in a WBS is further elaborated into number of specific activities or tasks that can be individually estimated, scheduled, monitored and controlled. Task lists are generated with their attributes such as task identifiers, their codes, descriptions, predecessor or successor tasks, leads and lags.

Task Sequencing

– In order to understand the schedule for completion of the tasks, the tasks are to be sequenced by identifying and documenting the dependencies or logical relationships among them. A project management software is used for developing a scheduling or a network diagram with appropriate leads and lags.



Resource Estimation

– This exercise aims to estimate the type and numbers or quantities of resources required to perform each task. Effort is estimated per task with any of the estimation methods such as by expert judgement, available estimate data, organizational resource capability baselines, published data, estimation methodologies and using project management software.

Task Duration

– Based on the resource capabilities or expert judgement or past data, a task duration is obtained. Parametric estimates based on productivity rates of resources or three point estimates (most likely, optimistic and pessimistic) methodologies are couple of additional methods, which can be used. Usually a contingency period is added to the duration as a buffer for scheduled risk.



Schedule

– A project schedule or an iteration schedule prepared based on an analysis of task sequences, durations, schedule constraints and the resource plan. It can change over the project timeline and is used as a baseline against which a project can be tracked. Project Management Software is used, and analytical techniques such as Critical Path Method, Critical Chain Method, What-if analysis, Resource leveling and Schedule compression techniques are used to develop an optimized project schedule.

Cost



– An estimate of the costs is developed based on the resources planned to be deployed per task, and the task duration. Analogous cost estimation techniques based on past projects, determination of unit resource cost rates, parametric techniques using statistical relationships between historical costs and other variables (e.g. lines of code) are some methodologies to determine costs of tasks.

Budgeting

– The project budget is prepared by aggregating all the costs associated for the tasks or work packages. An overall budget is prepared and this is useful for funding requirements for the project.



Quality

– In order to determine the relevance of quality standards to be used for the project, and meet the quality requirements, a quality plan is designed. Quality planning uses techniques like benchmarking against other projects, determining the cost of investments in quality and computing the cost-benefit analysis. The details related to metrics to be tracked, quality related checklists, improvements planned and quality baseline are delineated in the quality management plan.

Staffing

– Project roles, responsibilities, reporting relationships, the organizational hierarchy, are drafted as part of the HR plan. The project staffing plan is derived based on the HR plan, and includes plan for staff acquisition, time table, training needs, release criteria for resources, recognition of rewards, etc.

Project Communications

– A plan for communication, frequency of communication, escalation process, and the communication infrastructure or technology is to be set for the project. The communication plan is also developed keeping in mind the information and communication needs of stakeholders.



Risk Management

– This section of the plan determines the process, by which a risk is identified and the risk management activities that are performed.

- **Risk Identification** – Risks that affect the project are identified and documented in a Risk Register.
- **Qualitative Risk Analysis** – Risks are prioritized for further analysis, after analyzing their probability, occurrence of their impact, and details are updated in the Risk Register.
- **Quantitative Risk Analysis** – Based on a numerical analysis of effect on overall objectives of identified risks, the risks are prioritized.
- **Risk Response** – This elaborates the strategies or responses to be taken to avoid or mitigate the risks or their effects if they have occurred.

Purchase & Acquisitions

– The plan regarding purchase requirements for execution of projects, e.g. software or hardware requirements and their schedule or expected timeline is covered here.

Contract Plan

– If there are external parties needed to work on the project, contracts are necessary for documenting service requirements, product expectations or results, and for identification of potential sellers.

Project Execution

The project execution plan consists of activities or processes required to complete the project as planned. The people and resources available are to be coordinated and directed to achieve the project objectives. Additional approved change requests if any, are also executed as part of the project execution. Variances in project which require activity duration changes, unanticipated risks or requirements, etc. may affect the project management plan and therefore, a change request can be triggered and approved by customer. Such a change request potentially changes the project management plan and possibly can impact schedule.

Following are the activities performed during project execution:

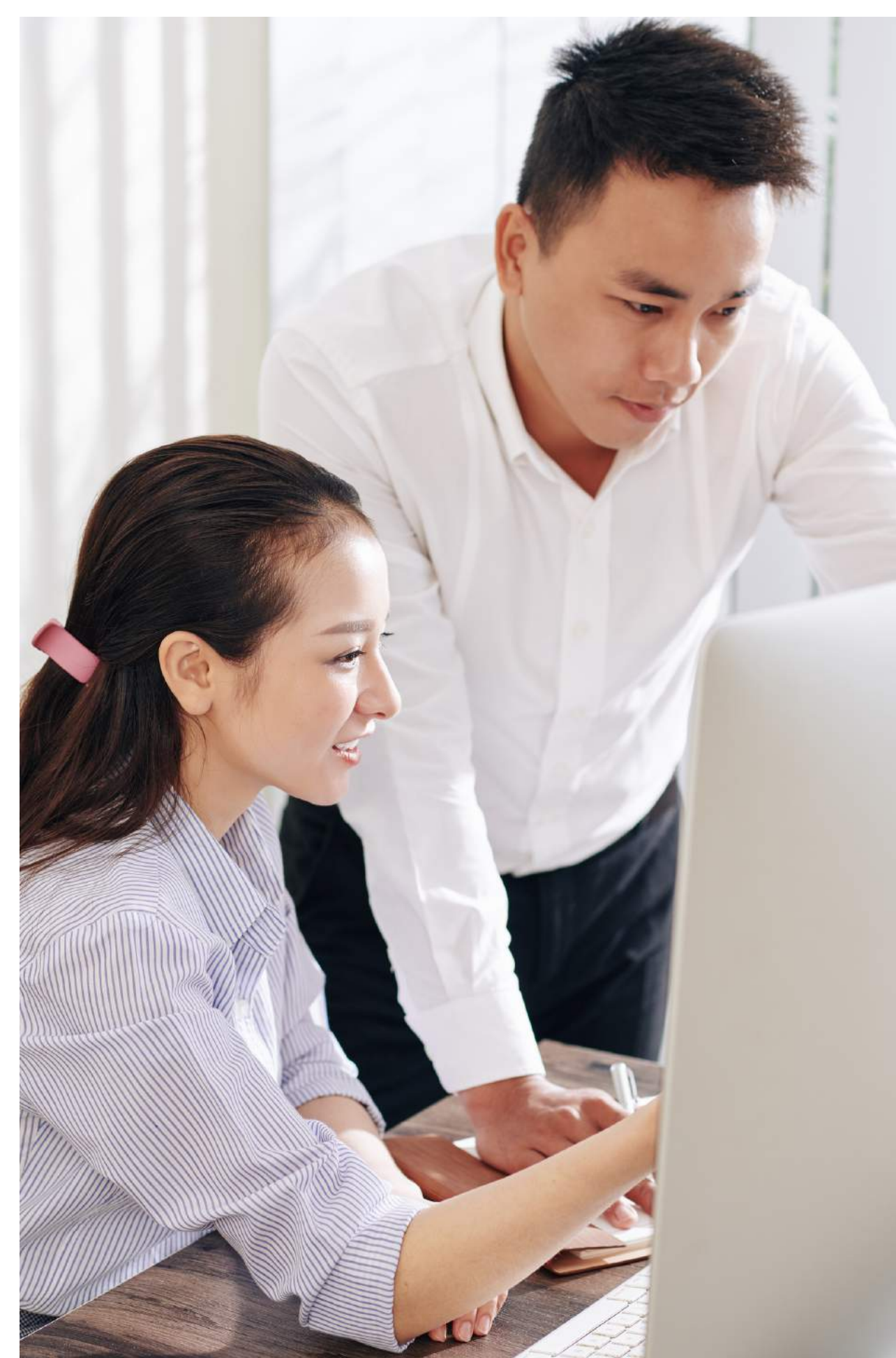
- **Project Execution Management** – This requires direction and management of the planned tasks to completion. The Project Manager is responsible for executing and controlling the project resources and team. Status of completion of deliverables and work accomplished is shared as part of Performance Reporting Process.
- **Quality Assurance** – During the process, the planned quality activities are performed to ensure that the project uses desired processes to achieve the desired quality goals. A few tools used for quality assurance include the tools used for quality planning,, quality audits, process analysis and other quality control tools.

- **Project Team Acquisition** – During this process, the project augments the number of team members working on it based on the Staffing Management Plan. This ensures that the resources are available when required.
- **Project Team development** – Not only the team is acquired, the competencies of the team members are nurtured during this process, by formal and informal methods. Knowledge sessions, sharing documentation, classroom trainings, online, computer-based, on-the-job training, mentoring sessions are some of these methods. General management skills and interpersonal or “soft skills” are stressed upon. Indicators of team performance assessment include improvements of skills in assigned tasks, improvement in competencies and sentiments of the team that enable better performance and reduced staff turnover rate.
- **Communication of Information** – Project information updated on Project Management Software, and status reports are conveyed to all stakeholders in a timely manner, using collaboration tools, web-conferencing, emails, etc. Lessons learned are shared using a Knowledge Repository.
- **Request for vendor proposals** – In case of outsourcing part of the work to an external vendor, a separate process for requesting proposals from vendors is initiated.
- **Selection of vendors** – This process is required for reviewing proposals for outsourcing work planned, and select the right vendors after negotiating contracts with them.

Project Monitoring & Control

An ongoing monitoring of the project activities against the project plan provides an insight to the project team about the health of the project. At the same time, it highlights areas which require additional attention. Variances in schedule, effort and other parameters overall could affect the project objectives, which could result in a revisit of the planning processes, and in turn require updates in the planning process. The processes include:

- **Monitor & Control Activities** – This includes collection, measurement and dissemination of project status. Status reports in terms of scope, duration, budget, resources, issues, quality and risks are indicators of project performance.



- **Scope Verification** – Monitoring the development processes for meeting scope requirements and Change Requests requires an ongoing verification of scope for deviations if any. The verification involves customer's review of deliverables using product reviews, audits and walkthroughs. The deliverables are accepted in the process, or customer may raise requests for change. Alternatively, the customer might reject the deliverables citing issues in meeting scope requirements.
- **Scope Control** – This process ensures that the factors that could cause scope changes are controlled, and that uncontrolled changes do not affect the system causing scope creep. The fundamental tools to control the scope are a Change Control System, variance analysis to determine the magnitude and cause of variance, determine if corrective action or a re-planning is required, thereby, adjusting the scope statement / baseline, the WBS, and the project management plan.
- **Control of Project Duration** – The duration of project is controlled based on project plan. A project management software is used. Tracking scheduling variance, progress reporting, monitoring change requests, monitoring critical path effectively are a few ways to control the schedule. A Change Control System with the requisite tracking mechanisms and authorizations is required for changing project schedule, if there is a need to do so.
- **Cost Control** – This involves influencing factors which could potentially affect the project budget by controlling changes. Cost control is achieved by monitoring the variance with the cost baseline, earned value analysis and using a cost change control system.
- **Performing Quality control** – Quality control is the task of evaluating the deliverables to confirm whether they comply with quality standards or acceptance criteria

Team Management

– This involves monitoring of task assignments, providing feedbacks, resolving bugs, and monitoring the implementation of change requirements. Project Performance Appraisals are useful tools for tracking team members' performance on projects.



Performance Reporting

– Project performance is to be collected and disseminated and therefore, progress measurement, status reporting and forecasting form part of the reports.

Stakeholders Management

– Communication with stakeholders is essential for stakeholder satisfaction. Product demonstrations, stakeholders' feedback, face to face meetings, web-conferencing, emails and telephone calls are important mechanisms for stakeholder management.

Risk Control

– Communication with stakeholders is essential for stakeholder satisfaction. Product demonstrations, stakeholders' feedback, face to face meetings, web-conferencing, emails and telephone calls are important mechanisms for stakeholder management.

Vendor Administration

– The key indicators of the vendors' performance are monitored as per commitments set in contract, and vendor performance reports are shared. Actions are taken based on the performance, such as additional follow-ups, motivations for the vendor, closure of contract, or additional contracts.

Project Closure

– At the end of the project, after the deliverables are completely handed over to your team, the activities of the project are terminated.

This involves a formal acceptance that you, the customer, has signed off the project, all deliverables are complete for receiving the final payment, the project retrospective meeting is conducted to review the best practices used during the project and a post-mortem of issues faced during the project is carried out. Details of Project Retrospective Meeting is shared with all stakeholders.

Contract Closure

– This determines the financial closure of the project. All the components of the Contract are reviewed here and pending issues, e.g. project outstanding are collected from customer to close the project.



How to set up quality assurance for your outsourced software development project(s)



Maintaining Quality in an outsourced product development model takes considerable effort. Be sure you are prepared well.. way before you even embark on this journey. Here are some tips. Your technology partner is responsible for ensuring that the commitments made to you/ their customers are delivered as per expectations, and that your development team or your business users are satisfied at all times.

Keep in mind that for most outsourced development partners, repeat business from you (their client) and payments as scheduled (from you, the development manager) are a couple of important indices for judging customer satisfaction. Leverage these.

A quality assurance framework woven around the key processes is an important ingredient to ensure consistency in quality of work, client satisfaction and value for money.

The Quality Assurance paradigm consists of a process framework to govern the areas which impact quality. This process framework typically consists (or should consist) of the following elements:

1. Requirements identification and management
2. Project scheduling, risk & tasks management
3. Design Validation
4. Development Assurance
5. Testing & Validations

Requirements Identification & Management

The Business Analyst along with the Project Manager from both parties (your firm and the software development partner firm) ensure that the business and development requirements are clearly understood and shared internally within the *entire* project team. This project team includes your staff and your development partners' staff as well.

A process of maintaining communication on a daily basis, documenting the requirements with you, confirmation and sign-offs is designed so that the project requirements are not misunderstood. This is crucial – and is recommended so that each party is fully aware of what is being developed, what the expectations are, what the definition of done is etc.. on BOTH teams (not just the development partner team's side).

Moreover, the requirements should be configured in a Requirements Management Tool, or a centralized repository with visibility to every member of the team. This could even be a spreadsheet if you are not using a requirements management tool. The most important thing is that this be actually maintained. Elaborate requirement management meetings are arranged to confirm that the requirements are well understood by the team. This cannot be stressed enough – variances in understanding of technical requirements and overarching business requirements/drivers are what lead to much bigger headaches in the future.

Project Scheduling & Task Management

The Project Manager is responsible for governing the project and is responsible for estimation, scheduling, task management and monitoring project execution. Try to have a dedicated project manager on your side as well. The Project Managers inform BOTH teams about the risks involved in the project and updates the status of the risks throughout the project.

In addition, collaboration tools and automatic task schedulers ensure that planned activities are executed as per plan, thereby adhering to commitments made to you as a customer and to the team from your (the customer) team.

Design Validation

A Technical Architect is responsible for the design and the architecture of the system. If you can have a technical architect on your team – it truly goes a long way towards the success of the product being developed. In reality, no matter what you do, there's always going to be some lack of trust from your side with regards to your software development team.



From time to time, certain technical design decisions are made and technical debts are undertaken by your outsourced software development partner to meet the deadlines set forth by your team. When you have a technical architect on your team, the development partner has a much easier time communicating these technical aspects with your firm.

When you do not have any technical representation on your side, typically, the development partner has to take the blame (and heat) for anything that goes wrong. This is usually bad news for the partnership in general and needs to be avoided at all costs.

The design of the system is not only prepared with the inputs from the Business Analyst and the client, but the design is also approved by your team / technical project manager before it can be considered for implementation.

Development Assurance

Quality during software development is ensured by the following steps – whether it is within your firm or done by your outsourced software development partner:

- Baseline coding standards, code structures, rules or any policies that are to be followed.
- Code walkthroughs to review whether code meets coding conventions, rules set, and organizational policies if any.
- Code certification by the Technical Architect
- Unit testing by each software engineer.
- Version controlling the code using configuration management tool and labelling the releases.
- Collaboration using automated tools.
- Daily meetings to review progress and issues.

- Client validation during and at the end of coding of each module
- Testing & Validations

Many times, on-shore development managers themselves do not have such rigid processes set up. It is recommended to agree on such steps and processes before actually starting the work. You are not required to utilize and implement each and every one of these recommendations, but the more you do, the easier it is to slip into a smooth, functional, working relationship with the development partner.

Quality Assurance activities during test management involves a series of testing and validations. The primary activities in testing are:

Creation of a test plan & test cases

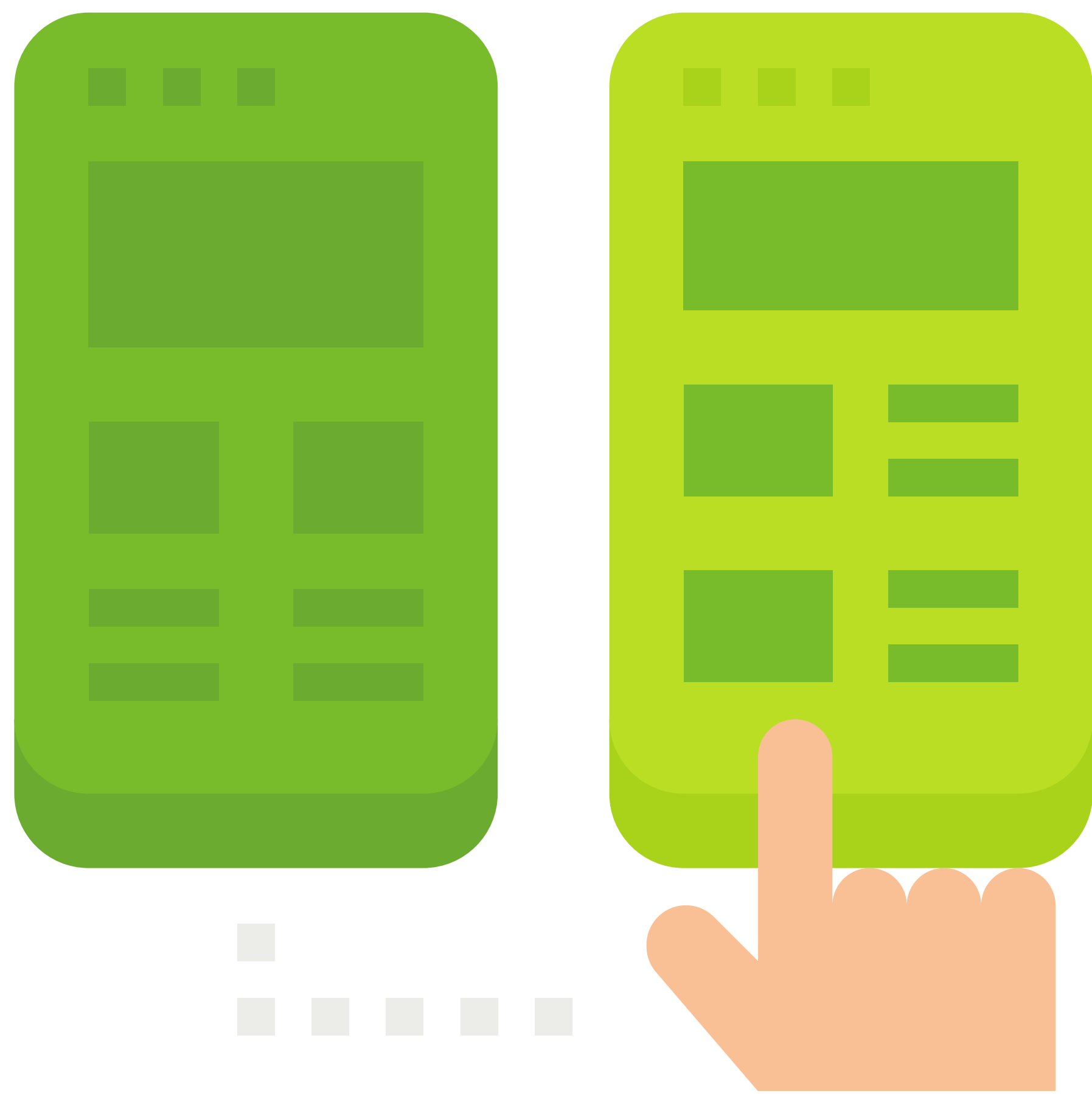
– A test plan elaborates the procedure by which testing would be done for the features of each module. Test cases on the other hand represent the testing specifications, which are run or executed to observe the outputs of the software for a set of inputs. The QA Analyst starts documenting the test cases and test plan early in the lifecycle of the project, so that the test cases are completely reviewed and accepted before the start of testing activities.

Automation Regression suites

– By automating tests which are frequently run, the regression test suites save a lot of manual effort, and are run frequently to catch regression errors, or conduct routine testing. Automation suites free up testing resources from mundane testing efforts, and bring forth predictability in testing, that could be lacking in manual forms of testing. As you already realize one benefit of investing in automated testing is that your total cost of testing software manually goes down as the software grows in features and functions. This does require you to commit to having an automated testing team – whether it is onshore or from your development partner.

Creation of test data

– Test data creation is an important aspect in software testing. A number of bugs often get replicated with adequate data, which are not to be seen with inadequate data. Data is created by the QA Analyst and team of considerable size for advanced functional testing and performance testing.



Black Box Testing

– After the Lead Software Engineer approves the module based on integration testing, the QA Analyst does a black box testing for the features, and logs issues in an Issue Log. The issue log needs to be addressed by the development team based on priority of the bugs.

Validation of software from your team

As soon as the module is approved by QA Analyst based on black-box testing, the module should be shared with your team for validation and confirmation. The issues reported by your team should be logged in the issue log which are to be fixed by the development team, validated by QA and released in subsequent releases. This might seem like an obvious step but you'd be surprised at the additional planning and preparation that is required from your side as well. Even if you have outsourced entire software development to a development partner, your development team or business users still need to commit to validating a software's features and functions at least once before the software is made available to the world (generally available).

- **System Testing** – QA Analyst is responsible for testing the system at the end of the project or at the end of each software release cycle. The issues / bugs are logged and tracked till they are closed. An issue management tool helps in this process. Also, the Business Analyst tests the software to see that the requirements are met, as desired. The System Testing should be focused on the following areas to ensure that all the areas of the testing are covered:
- **Functionality Testing** – This involves testing the features of the product, i.e. what the product does.

- **Usability Testing** – This includes capturing and stating requirements based on user interface issues, e.g. issues such as accessibility, interface aesthetics, consistency, browser compatibility, etc.
- **Performance** – Performance involves issues such as throughput of processing information, response time, start up time and recovery time.
- **Supportability** – This group of requirements address supporting the software, such as adaptability, maintainability, compatibility, configurability, scalability, localization and installation requirements.
- **Beta Testing** – This testing is done at the client's end by using a number of users or mock users. By doing testing which closely represents usage by actual users, this testing helps in streamlining the product for releasing to actual users.

Maintaining Quality in an outsourced product development model

With teams across continents, it is important to focus on quality and ensure that right processes are established throughout the product development and during the project lifecycle, in order to maintain quality uniformly across teams.

We have also witnessed the need for having releases quicker and better over time and this has led us to carve out a framework for outsourced development. The framework is based on the widely accepted Agile Methodology, and therefore, applicable globally to existing users of Agile Methodologies and other companies which want to transition their operations and adopt this methodology.

Quality is a key attribute in this framework and is based on the “Agility Manifesto” that values “Working software over comprehensive documentation”. This seems to be in tune with J M Juran, one of the TQM gurus who had coined the interesting phrase “fitness for use” to define product quality. We believe that by consistently developing products using this framework, one can achieve higher levels of product owner satisfaction.

While a traditional outsourced software development project usually has a detailed Quality Plan, against which the project quality is monitored, such a plan may not be required in an Agile framework. The quality assurance and control, which are significant efforts in traditional projects are already inbuilt within the framework, and ensured by the Agile process. Here are a few practices in this framework which are effective for maintaining quality norms:

Product Owner Integral to the Team

Since the Product Owner is a person responsible for the requirements, and the team follows these requirements, the first step to ensure that the product is developed as per requirements is to have the Product Owner as part of the team (yes, you can still reap the benefits of outsourced software development). The Product Owner therefore participates in the daily scrum meetings and all the other meetings, and this ensures that the development is very well aligned to the requirements.



Software for release in every iteration

Though meeting the functional requirements of the Product Owner could be a priority, it is not only the criterion met during each release. As there are multiple releases based on iterations planned, the releases are carried out with all the basic requirements for a release such as:

- meeting the Product Owner’s expectations for the specific Time box,
- having the best design for the features implemented already (by refactoring wherever there is a need),

- meeting the coding standards set (by code review and corrections),
- testing satisfactorily by the relevant stakeholders and the team, and arranging for easy maintenance of each release easily.

Product Reviews – scheduled and unscheduled

There are two kind of product reviews that you should be following (at a minimum):

Scheduled Product Reviews – The product is formally reviewed towards the end of each iteration. If required, more scheduled reviews can also be organized. The observations of the reviews are documented, assigned to team; the progress is discussed during the daily review meetings and changes required are incorporated in the plan.

Unscheduled Product Reviews – Since Product Owner is part of the team, there are opportunities to review the product informally as well as from time to time. The review comments are noted and the action items from these meetings are shared with the team for further action.

Moreover, the product also gets reviewed during the daily scrum meetings to determine whether the right approach is being followed.

Using a Test Driven Development Framework

It is recommended to have a Test Driven Development framework. These are the tests written to represent requirements or user stories. It is needless to say that the testing team (yours and your development partner's) is in the forefront of understanding the requirements from the Product Owner and writing the test specifications.

The different types of testing that the testing team is involved in are:

- **Acceptance Tests** – The requirements in agile development, also called User Stories represent the high level description of the business rules, or the external behaviors of the system. Each user story corresponds to at least one Acceptance test case. Acceptance test cases are usually reviewed by Product Owner to ensure that they follow the Product Owner's intent correctly.

- **Unit tests** – Both automated unit tests and manual unit tests are written to ensure that the issues particularly at unit level, are addressed. Automated unit tests help in regression testing.
- **Regression Tests** – Where possible, acceptance tests and unit tests are automated so that they can act as a regression suite. This suite can validate that changes in code do not affect the basic functionality or reintroduce issues which have been fixed in previous iterations. As writing regression test suites involve some effort, they are carefully selected, and only those test cases, which are run repeatedly are converted to regression suite.
- **Exploratory Testing**– This testing, as the name suggests is exploratory in nature and can be used in root cause analysis, identifying show stoppers like system crashing or unhandled errors, which are usually fixed immediately; and other less serious problems, which might be deferred. This testing could lead to new user stories that would get added to the backlog.
- **Specialist Testing**– The specialist testing includes extra testing activities, such as performance testing, etc., which require the help of specialists, or any other critical area where the regular testing team does not have the capability.

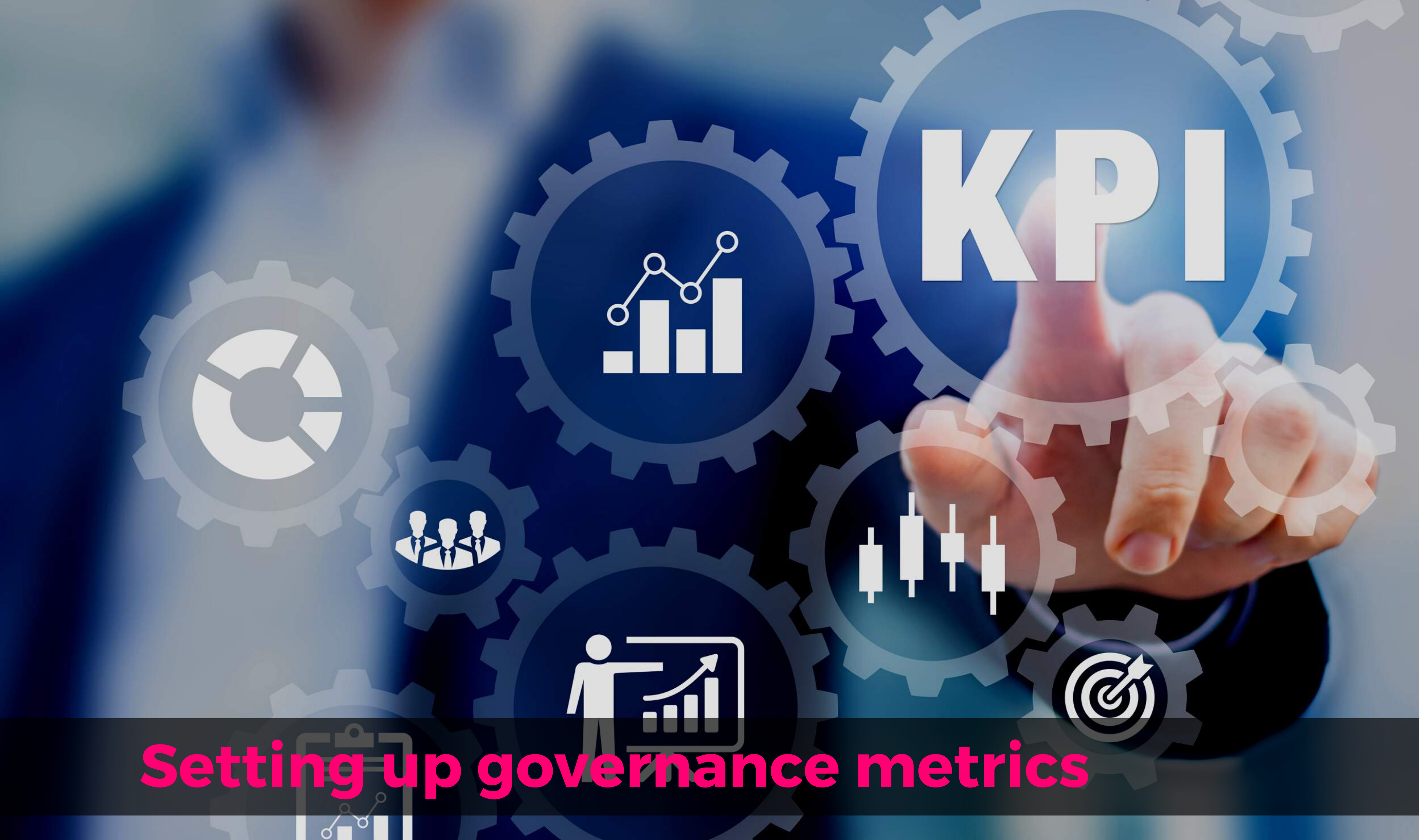
Development / Code Review



Do not skip development and code reviews. While you don't necessarily have to have this skill set on your team itself, you should request the same from your development partner. Do keep in mind that having all these processes in place do slow down development to some degree, but it is tremendously beneficial in the long run.

Code reviews should be done either by:

- **Traditional code reviewer method:** A code reviewer is assigned for reviewing all the developers' code.
- **Two Pairs of eyes approach:** In this approach, the code is developed by two developers. This ensures that while one is coding, the other is reviewing the code to ensure that coding standards are agreed to, design guidelines are met and are easily understood by the developer other than the author.



Setting up governance metrics

A set of metrics should be used to better govern the parameters of performance and improve the overall quality of work. These metrics include:

- Drag Factor – This indicates the effort in hours which do not contribute to iteration / sprint goals. Drag factors can be reduced by reducing the number of shared resources / non-contributing work. Estimates are refactored based on Drag Factor wherein the New Estimate = Old Estimate + Drag factor
- Velocity – This indicates the amount of backlog which can be converted to a shippable functionality of a sprint. This indicates the capacity of the team to complete requirements within a sprint, and can be used as an input for estimations.
- Number of unit tests or acceptance tests added
- Time taken to complete daily build
- Bugs detected per iteration
- Production defect leakage
- Test coverage
- Continuous Integration – Continuous Integration requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, usually done daily, allowing teams to detect problems early. This can be extended by automatically integrating and running a regression test every time a build is deployed. Running an automated regression test frequently means defects are highlighted soon after they are introduced (i.e. when the build goes Red, or it fails). The team's top priority is to get the build Green again.

Informative Workspace

Finally, project related dashboards and infographics are useful to provide a management overview of the project. Ideally, this has the Burn-down charts, the Sprint / Iteration Plan, current build status and additional metrics. This provides a better view for higher management, who can help if there is a need for the same.



Retrospective Meetings

Retrospective meetings are planned to reflect on the activities performed, find how the team performed during the last sprint, and assess the good practices or misadventures. The planning for the next sprint is done considering the learnings during this meeting. In the long run, these meetings help considerably in improving the quality of development (even more so, in outsourced software development development).



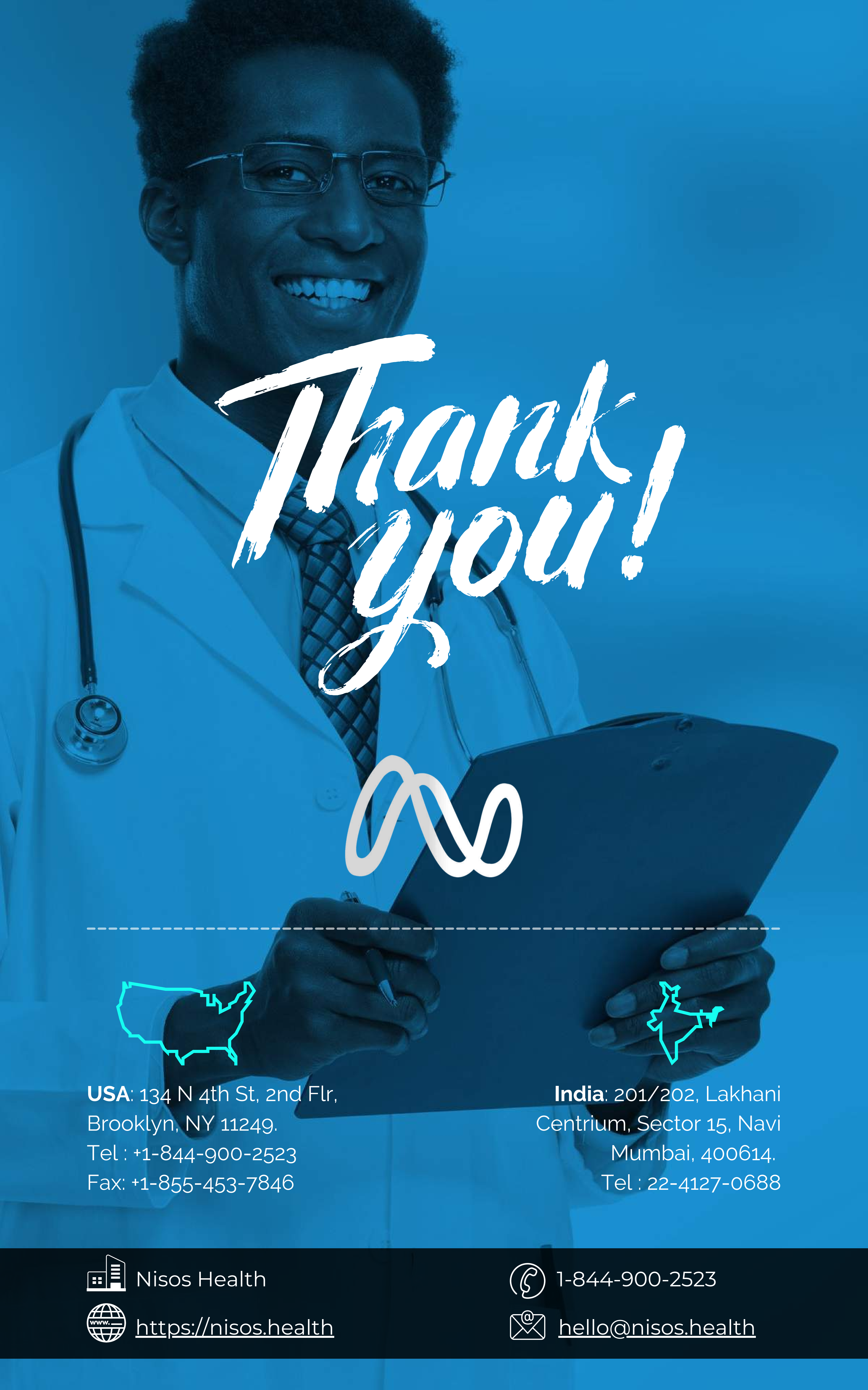


OUTSOURCED SOFTWARE DEVELOPMENT – GUIDE FOR MANAGERS

Written by Nisos Health (nisos.health)

Want to get started?

Our software and services help providers reduce operational expenses, increase collections, improve patient outreach and patient experience. Healthcare organizations rely on us for call center solutions, healthcare software services, healthcare BPO, medical billing, revenue cycle management solutions.



Thank you!



USA: 134 N 4th St, 2nd Flr,
Brooklyn, NY 11249.
Tel : +1-844-900-2523
Fax: +1-855-453-7846



India: 201/202, Lakhani
Centrium, Sector 15, Navi
Mumbai, 400614.
Tel : 22-4127-0688



Nisos Health



<https://nisos.health>



1-844-900-2523



hello@nisos.health