

net solutions

50-Point Checklist to Select a Development Partner



This tool will help you take a well-informed decision while selecting a development partner. You can refer to this checklist to ask relevant questions that will help you understand and evaluate the potential partners. Not every company will have all the answers or provide a satisfactory response. By comparing their responses and grading them to your priorities, you can make the final decision with confidence.

Overview

1 Tell me about your Organization.

Tip: Identify the overall profile of the organization, ask specific questions if they do not cover these in the description:

a How long have you been in business?

Tip: Generally longevity means stability, but fewer years qualified by a reasonable explanation of history also gives you insight into the experience of a new organization. Older organizations have proven processes; newer ones have a hunger to prove themselves. Balance it out with your risk appetite.

b How many employees do you have? How many are developers?

Tip: Just get an idea; deeper into the details later.

c How many clients have you worked with? How many projects has your company successfully completed with them? Have you done business with any companies in <your country/your general area>?

Tip: This gives you an idea of their exposure and experience, and having experience with people in your general area is a great advantage. The number of clients and projects should give you an idea of their exposure.

d How many clients are you currently working with?

Tip: Use this with other information to verify if the PMs/developers are overloaded and the scale of projects they work on. For example, 200 developers working on 100 projects means a bevy of small projects.

e How long have you worked with your oldest client?

Tip: Use this to get an idea of longevity of their focus, ability to deliver quality, etc.

2 Where are you based?

a Do you have any other offices?

b Does the sales team and the development team sit at the same location?

Tip: Having the team together ensures nothing is lost in translation and the sales team proposes a viable solution.

3 **What are your company's values?**

Tip: They should align with yours.

4 **What are your core services offering?**

Tip: Are you going to need these services in your engagement?

5 **What are your areas of expertise?**

Tip: Does this align with what you are seeking to build?

6 **How does your company differentiate itself from others? What is special about you?**

Tip: See how they define the difference and evaluate whether it is a beneficial difference.

Experience

7 **What is the average size/duration of your projects?**

Tip: Together with #1, this will give you an idea of their quality/focus.

8 **Have you worked on projects similar to ours? What is your experience in my vertical?**

Tip: Those who have worked on similar projects and have experience in your vertical will always have an edge over the others in being able to understand quickly and design a better product. Beware of how they qualify 'similar' though — most will pick upon a familiar feature than the overall project or vertical.

9 **Do you subcontract/outsource any work?**

Tip: Dig deeper if they do and identify why, how frequently, and how they control quality/timelines/etc. especially if they do not have regular partners. Generally, you should avoid companies who sub-contract.

Processes

10 **Tell me more about your process/approach to executing a software development process.**

Tip: Here, focus on the big picture first.

11 **How do you collaborate with clients at different stages?**

Tip: The core process should be followed by how which should change slightly for all different stages like requirement gathering, design & planning, production, and deployment/post-deployment.

12 Describe your Software Development Life Cycle (SDLC) in detail.

Tip: Ideally they should describe some iterative development strategy like Agile.

a Why have you chosen this model?

Tip: Are they able to enunciate the benefits of their chosen model?

b Are your Project Manager, Business Analyst and Account Manager different people?

Tip: They should be.

c How do you capture & manage requirements?

(i) How do you ensure sign-offs for changing requirements?

(ii) How do you ensure the latest requirements are visible and available to the team and the team is working on them?

Tip: Dig deeper into their SDLC and see how they capture & track requirements.

d What is your design process?

(i) Do you build personas, write user stories? What tool do you use?

(ii) Do you create wireframes/prototypes? What tool do you use?

Tip: A design-thinking oriented approach has user stories & personas that are essential for a well-thought-out platform today.

e What communication tools do you use internally?

Tip: An effective internal communication tool ensures better communication between the various team members.

f What project management tools do you use?

Tip: They should use industry-standard, reliable tools to manage their project and be open to using your preferred tool if necessary.

13 Do you use DevOps?

a How is your DevOps structured and implemented?

b What tools do you use for DevOps?

Tip: Prefer those who use DevOps since it provides many benefits. If they do, explore how they define DevOps and whether that aligns with the SDLC they described before. There should not be any discrepancy between the two.

14 How does your team keep clients informed of the progress? How would my team track progress without asking for it?

Tip: Focus on where and how they record their progress and whether their clients have access to this information. This is different from collaboration or how the PM or developers inform you. Here focus on how they make their work visible so that your team is aware of what is going on. For example, a good answer is "The team posts daily status reports of work accomplished."

15 **How do you work with external teams? For example, what if we wanted somebody else to do UI or testing?**

Tip: They should be open towards working with outside help and have processes in place to involve their representatives.

16 **What metrics do you track? Which tools do you use and how do you collect them?**

Tip: At the very least they should be tracking work that is done and an internal bug listing.

Quality

17 **Is your Quality Control independent of your development?**

Tip: Ideally it should be or at least reporting to the PM & CTO.

18 **How do you ensure software quality? Could you give me a few examples?**

Tip: Some of this might have been covered in general processes but dig deeper through examples.

19 **What industry standards do you follow for quality assurance?**

Tip: Do they follow any international quality standards like ISO certification?

20 **Do you use Continuous Testing?**

- a **What tools do you use?**
- b **How does the CT process get triggered?**
- c **How are the issues reported?**
- d **How can we see reports generated by CT stack?**

Tip: Ideally they should be using it and the process should provide for trigger and reporting mechanisms. They should be open towards showing a sample report.

21 **Do you use any automation in software testing?**

- a **What tools do you use?**
- b **If you use CT, are these tools used too?**

Tip: Ideally they should be using some level of automation during testing. A standard toolset should be used with/without Continuous Testing integration.

22 **Do you do any manual testing?**

Tip: Automation in testing should not count out manual testing. Seek details of their processes to see if they align with the automated testing process.

23 **Do you do security testing? Please describe the process and guidelines followed.**

Tip: Security testing is a must today and they should be doing it as well. It should be done with a security testing tool that checks both the code and the database, with any integrations with third parties as well.

24 **Do you do load testing? How do you do it?**

Tip: Explore if they do, and if so, what metrics do they choose to monitor and how.

Team Size & Credentials

25 **What technologies do you support?**

Tip: A range of technologies is better than a few since that shows they have a range of experience available.

26 **What is the size of the teams for each of the technologies you support?**

Tip: Too small a team means a lack of available talent and peer-assisted skill enhancement.

27 **How are your technology teams structured? What are the experience levels of developers in these technologies?**

Tip: Identify if they have formalised a proper pyramid structure. A flat structure is fine if the experience level of developers involved is high. Ideally there should be a good spread from freshers to tech/team leads and not lopsided, in which case dig deeper into the reasoning.

28 **How does your team keep abreast of the latest? How does that help your clients and existing projects?**

Tip: Are there any organizational supported programs that assist the developers?

29 **Do you have any internal training programs for your developers?**

Tip: Ideally there should be a training/upskilling program hosted by senior developers to ensure there is constant upskilling.

30 **Do your developers have any certifications?**

a **How many certifications did they get in the last 1 year?**

b **How do you support your developers in getting certified?**

Tip: They should have certified developers, especially in technologies of your interest. Depending on their annual figure, see if they run a program for employees to get additional certifications.

31 **How do you hire new developers? What do you look for when hiring new developers?**

Tip: Identify how they choose developers. It should be a balanced approach between technical competence & a go-getter attitude.

32 **Do you have coding practices written down? Could you share a copy (even partially) of the relevant technology with me?**

a **How do you ensure the development is being done according to these practices?**

b **Does your team use any static code analysis tools?**

c **How do you document code?**

Tip: The team should follow coding standards and practices that have been documented for their reference within the organization. There should be both automated (preferably through a static code analysis tool) and manual review of written code.

33 **What infrastructure capabilities does your team have?**

Tip: Identify whether they support the cloud provider you prefer and seek reasoning why if they support/prefer a different cloud provider.

Flexibility

34 **How many projects does your project manager handle simultaneously?**

Tip: Ideally 1, but not more than 3. Multiple projects should only be allowed if they are small enough to not justify a full-time PM.

35 **Do your developers work on multiple projects at the same time?**

Tip: The answer should be an emphatic no.

36 **What is your process of scaling the team up or down?**

a **How long would you need to increase developers in the team by 1, 3 or 5 developers?**

Tip: 1-2 developers should be less than a month; more in 45-60 days unless it is a niche technology where it may take longer. Seek explanation to understand their reasoning.

b **What is the process for reducing the number of developers on the team?**

Tip: It should be a formal process, written in the contract and ideally range from 30-60 days based on skill-set rarity.

Communication

37 **How do we communicate during the project? What tools do you use with your clients?**

Tip: There should be at least a central communication hub where things are recorded for the project, an instant messaging app, and a video-conferencing app. Are they open to using apps that you prefer?

38 What is the English competency of the team that we shall be working with?

Tip: Beware of a blanket yes. Seek details on how they are evaluating the competency.

39 What is the escalation process?

Tip: For one, there should be one and it should be a formal, documented process. Inquire about the job profiles of the senior staff. The hierarchy should make sense and you should be clear what each level brings to the table.

40 If anything goes wrong, how do you ensure we know? Could you describe a similar situation that happened recently?

Tip: Check if they have proper processes for documenting/sharing bad news. In the example, check for how quickly they took ownership of both the problem and the solution.

Pricing

41 What is your pricing model?

Tip: Understand how they charge - by the day, by the hour, etc. What is included in those charges and what is not included? Do they have different pricing models aligned with different engagement models?

42 What is your payment schedule?

Tip: It should be clearly stated both during the discussion and listed in the contract. See if it is different for different engagement/pricing models and what flexibility they allow.

Knowledge Management

43 What is your attrition rate?

Tip: Ask for a number. The industry average is around 20% and a lower rate is better. If it is more than 15% different, ask for why theirs is so away from the average. This will give you an idea of their internal employee happiness/satisfaction.

44 If a developer or PM on the team wants to leave your organization in the middle of the project, how do you retain them?

a What happens if they cannot be retained?

(i) How do you retain knowledge?

(ii) How do you ensure confidentiality?

Tip: Flag for a reluctant answer or a broadly generic one; seek specific information to get an idea of their internal values on treating their employees. Since PMs are more valuable than developers, see if they have a different set of rules for senior employees like a PM. Happy employees mean better code written faster with fewer errors. Look for their knowledge management practices in this answer. There should be a formal process that ensures knowledge & information does not leave with an employee. Look for employee offboarding processes like notice period where a longer period allows for better handover to the new PM taking over. See if they have an internal NDA with their employees to ensure confidentiality of projects post their leaving the organization.

Legalities

45 Will you sign an NDA? Is the NDA perpetual or time-limited?

Tip: There should not be any reluctance around NDA & the NDA should not be time-limited. Ensure that the NDA has a no-marketing without permission clause as well. If they sub-contract, then dig deeper into how they will ensure confidentiality.

46 What are the main points of the contract we shall sign with you?

- a Does it explicitly say who owns the design & code?
- b Does it mention rates clearly in unambiguous terms?
- c What are the warranties & indemnities you provide?

Tip: The contract should cover not only the terms of engagement & payment, but it should clearly state ownership of code, pricing, warranties, termination process, etc. Look for warranties that code produced would be original, bug-free (leading to a certain amount of free maintenance), free from any potential IP infringement risks, etc.

References

47 Where can we see reviews about you?

Tip: Testimonials on their marketing website aside, look for links to independent review sites like Clutch, Goodfirms, etc. where clients have given reviews.

48 Please provide at least 3 references with contact information.

Tip: See if they can provide at least one reference to a client in your time zone so that you can confirm how working with them was like in your context.

Additional

49 What would you expect from us during the engagement?

Tip: A lot should already be covered before but ask them to describe it collectively. It will give you an insight into how they structure their engagement and the collaboration points they use. See if there is any discrepancy from what they have said before.

50 How do you use your experience to help your clients deliver a better product? How do you provide advice and assistance?

Tip: Try to identify something unique that others do not provide — it could be a process innovation or advice or introduction to VCs their previous clients have worked with, etc.

Net Solutions is a strategic design & build consultancy that unites creative design thinking with agile software development under one expert roof. Founded in 2000, we create award-winning transformative digital products & platforms for startups and enterprises worldwide. To discuss our services, call us on any of these numbers, or email us on info@netsolutions.com



LOS ANGELES

11601 Wilshire Blvd, West
Los Angeles, CA 90025,
USA



NEW YORK

101 6th Ave, 8th floor,
New York, NY 10013,
USA



TORONTO

111 Queen St E #450,
Toronto, ON M5C 1S2,
Canada



LONDON

Aldgate Tower,
2 Leaman Street,
London E1 8FA,
UK



CHANDIGARH

Site No. 15, Phase 1,
Chandigarh Technology Park,
Chandigarh, U.T. 160101,
India



PUNE

Pride Purple Square,
B 315-316, Wakad, Pune,
Maharashtra 411057,
India

