SUMATOSOFT

How to Choose The Offshore Development Company

5-Minute Guide

Part 1: Find the vendor

Use a structured checklist and compare vendors using the same criteria.

1) Validate relevant experience

Look for proven delivery in your domain and product type.

- Similar industries or workflows
- Similar complexity: integrations, performance, security, data volume
- Case studies with outcomes and scope, not only technology lists

2) Match the service model to your goal

Different needs require different engagement models.

- Full-cycle delivery for end-to-end responsibility
- Dedicated team for long-term product development
- Staff augmentation when you already have strong internal leadership
- Discovery + MVP when requirements are unclear

Check that the vendor's listed services match your engagement plan.

3) Evaluate engineering depth and delivery process

A strong offshore company has predictable delivery routines.

- Clear roles: PM, BA, tech lead, QA
- Regular demos and progress reporting
- Transparent backlog and change control
- Mature QA practices and CI/CD discipline

Ask how they handle scope changes, estimates, and release planning.

4) Check communication and collaboration fit

Offshore success depends on day-to-day execution.

- Overlap hours with your timezone
- Communication channels: Slack, email, calls
- Meeting cadence and escalation path
- English level for key roles

Ask who will be your main point of contact and how risks are surfaced.

5) Compare pricing with total delivery value

Hourly rate matters less than predictable outcomes.

- Compare price ranges with team seniority and delivery scope
- Verify what is included: QA, PM, DevOps, support
- Prefer clear billing rules for T&M and change requests
- Low rates often shift cost into rework and delays.

6) Use independent reviews to validate claims

- Cross-check vendor statements with external feedback.
- Review volume and rating stability
- Recent feedback and delivery consistency
- Mentions of communication, PM quality, and reliability
- Use platforms that verify reviews and show detailed project context.

7) Confirm ramp-up speed and scalability

- Ask how fast they can staff your exact roles
- Check replacement policy if someone leaves
- Confirm scalability plan for future phases

8) Reduce risk by discussing the pilot project

- If the scope is large, start small.
- Discovery or technical audit
- MVP or a single feature slice
- Clear success criteria: quality, speed, communication, predictability
- A short pilot reveals delivery maturity better than a sales call.

Part 2: Questions and Answers that help to identify the professional vendor

Below are high-impact questions to ask an offshore software development company. For each question, you get a 2-column table: what a strong answer looks like vs what a risky answer looks like.

1) What delivery model do you recommend for our case, and why?

✅Good answers	⛔Bad answers
Explains options (full-cycle, dedicated team, augmentation) and maps them to your goals, risks, and timeline	Pushes one model for every Client
Mentions roles, governance, and how decisions will be made	Focuses only on speed or price
Proposes a short discovery/pilot when scope is unclear	“We can start tomorrow, no need for discovery”

2) Who will be on the team, and what is the seniority mix?

✅Good answers	⛔Bad answers
Names core roles (PM/BA/Tech Lead/QA/DevOps) and provides seniority split	Avoids specifics or says “we’ll assign later”
Shares anonymized CVs or profile summaries and interview availability	Refuses to show who will do the work
Explains how staffing changes are handled	“People rotate, it’s normal”

3) How fast can you start, and what does onboarding include?

✅Good answers	⛔Bad answers
Gives a realistic timeline with steps: access, environments, repo, documentation, kickoff	“Next week” without any plan
Lists what they need from you (SMEs, stakeholders, access)	Assumes they need nothing

Defines onboarding outputs (setup, first sprint plan, risk log)	“We’ll figure it out as we go”

4) How do you estimate scope and manage changes?

✅Good answers	⛔Bad answers
Uses a clear estimation approach (story points, ranges, assumptions)	Gives a fixed number immediately
Defines change control and how scope changes affect time/cost	Treats changes as free
Proposes backlog grooming cadence and acceptance criteria	No process for requirements changes

5) What is your delivery cadence and reporting format?

✅Good answers	⛔Bad answers
Defines sprint rhythm, demos, weekly status, KPIs (velocity, defects, burndown)	“We’ll send updates sometimes”
Offers transparency in Jira/Confluence and regular demos	Keeps progress in private docs
Includes risk/issue tracking and clear escalation	No risk reporting

6) How do you ensure quality (QA), testing, and code review?

✅Good answers	⛔Bad answers
Clear QA strategy: unit/integration/e2e, review gates, definition of done	“Developers test their own work”
CI checks, code review rules, coverage targets, release criteria	No mention of automation
Bug triage process and severity SLAs	“Bugs happen, we’ll fix later”

7) What does your CI/CD and release process look like?

✅Good answers	Bad answers
Describes environments (dev/stage/prod), approvals, rollback, tagging, release notes	"We just deploy when ready"
Mentions IaC, secret management, monitoring and alerts	No monitoring or rollback plan
Offers to align with your existing pipeline (GitLab/GitHub/etc.)	Insists on ad-hoc manual deployments

8) How do you handle security, access, and compliance?

✅Good answers	Bad answers
Least privilege, MFA, SSO, audited access, secure secret handling	Shares credentials via chat/email
Secure SDLC practices, vulnerability scanning, patching routine	No security practices described
Can sign NDA/DPA, supports SOC 2/ISO 27001 practices when needed	"We don't do compliance"

9) Where will the code live, and who owns IP?

✅Good answers	Bad answers
Your org owns IP; code resides in your repos; clear handover terms	Vendor keeps code in their repo
Defines ownership of deliverables, libraries, documentation	Vague ownership language
Includes exit plan and knowledge transfer	Avoids discussion of transition

10) How do you handle time zones and communication?

✅Good answers	Bad answers
Confirms overlap hours, response times, meeting schedule, escalation path	"We'll respond when we can"

Clear channels (Slack, email, calls) and agreed etiquette	No defined communication rules
Names accountable PM/Tech Lead and backup	No accountable owner

11) What happens if a key person leaves the team?

✅Good answers	⛔Bad answers
Replacement SLA, shadowing, handover docs, minimal disruption plan	“It rarely happens” only
Knowledge base maintained continuously	Knowledge is in people’s heads
Contract covers continuity and transition	No continuity commitments

12) How do you measure success in the first 30–60 days?

✅Good answers	⛔Bad answers
Defines measurable outcomes: delivered scope, quality metrics, cycle time, stability	Talks only about “working hard”
Proposes a pilot with clear acceptance criteria	No success criteria
Aligns metrics with your business goals	Measures only hours billed

13) What are the payment terms, and what is included in the rate?

✅Good answers	⛔Bad answers
Transparent billing: roles included, meeting time policy, overtime rules	Hidden extras appear later
Clear contract type: T&M vs fixed price, invoicing cadence	“We’ll decide later”
Defines what triggers a change order	No change order process

14) Can you share 2–3 relevant references, and what did you deliver?

✅Good answers	⛔Bad answers
Provides relevant references, similar scope, and outcomes achieved	Only generic testimonials
Explains what went wrong on a past project and how they fixed it	Claims everything was perfect
Shows evidence: case studies, metrics, or deliverable examples	Refuses to share anything

15) What is your approach to documentation and handover?

✅Good answers	⛔Bad answers
Defines docs: architecture, runbooks, setup, ADRs, API docs	“Docs after launch”
Docs updated during development, part of Definition of Done	Docs are optional
Includes final handover checklist and training	No handover plan

SUMATOSOFT

Thank you for your time!

Any questions? Drop us a line!

Headquarters

One Boston Place, Suite 2602
Boston, MA 02108, United States

Other ways to get in touch

info@sumatosoft.com
sumatosoft.com