

Your SOFTWARE DEVELOPMENT JOURNEY

A clear guide to planning, buying, building, and caring for your software.

LARGE-PRINT EDITION

You do not need to know how to code. You need to know what you want to do, what you are buying, and when to make each choice.



Follow one client from an idea to launch. See the work behind the screen. Compare services, delivery levels, and sample costs.

START HERE

Use this guide to make a choice

You can buy help for the whole journey or for a clear part of it. The written scope must say where our work starts and ends.

Read

Start with the story and the journey map. Use the glossary when you meet a new word.

Choose

Compare the delivery levels. Mark each service as buy now, do yourself, use another provider, or save for later.

Agree

Before work starts, agree on the result, price, due dates, checks, and who does each task.



All prices are sample USD amounts. Final company rates and project terms must be approved in a written quote.

START HERE

Find the answer you need

Read from start to finish, or go straight to the decision in front of you.

Understand the work | Pages 4 to 8

Meet Maya, see the full route, explore the iceberg, and learn who does what.

Walk the journey | Pages 9 to 35

From your first idea through design, building, checks, launch, training, and care.

Choose and budget | Pages 36 to 44

Compare delivery levels, read the service menu, and see a complete sample budget.

Control change | Pages 45 to 50

Set boundaries, price new ideas, and make informed choices about work you decline.

Make your decision | Pages 51 to 58

Use the decision path, worksheets, plain-word glossary, and final review.



MEET YOUR GUIDE

Maya wants fewer missed jobs

Maya runs a small home repair team. Job requests arrive by phone, text, and email. Some get lost.



Her first idea

“I want one place to see each request and who will handle it.” She does not start with a long list of screens.

Her first release

Staff add a job, assign it, change its status, and find it later. The owner can see all jobs. A worker sees only assigned jobs.

Her later list

Customer payments, text alerts, and a phone app sound useful. Maya saves them for a later choice so the first build stays small.

Maya and her company are fictional. Her numbers are teaching examples, not a client result or promise.

Your journey at a glance

Each stage gives you something to review. A review may send us back to improve earlier work.



Plan | Idea through requirements

Name the problem, learn how the work runs, and agree on what the first version must do.

Shape and build | Design through development

See the screens, try the flow, choose the structure, and build in small parts.

Prove and open | Testing through launch

Check the work, try real tasks, review safety, and choose a launch date.

Care and improve | Handoff through support

Keep records, help users, watch running costs, and plan the next release.

Security, access, and accessibility start early and continue through the journey. They are not only final checks.

What you see

A clean screen is the visible tip. A small button may depend on a large amount of work below it.

The visible part

A page opens. A form saves. A list sorts. A message appears. The screen looks simple because many choices have already been made.



Visible: screens and buttons
Hidden: data, rules, tests,
access, recovery, and care.

Maya sees one button

She clicks “Assign job.” The worker’s name appears. That is the moment she can see.

Ask before you price

What must happen after that click?
Who can click it? What if it fails?
What must the system remember?

The iceberg is a teaching picture. Its size does not represent a fixed share of time or cost.

What we build underneath

The hidden work makes the visible action useful. It also helps the team find and fix problems later.



Visible: screens and buttons
Hidden: data, rules, tests, access, recovery, and care.

Behind “Assign job”

The system checks access, finds the right job, saves the worker, records the change, and shows a clear result.

When something goes wrong

What if two people edit at once? What if the worker is no longer active? What if the connection drops before the save ends?

Work you can ask to see

Request the data map, access rules, test results, known issue list, and handoff notes for the services you buy.

Visible features and hidden work both belong in the scope. A picture of a screen does not define them all.

Who makes each decision

One named client lead gathers feedback and gives clear approval. This helps the team avoid mixed instructions.



You own business choices

Explain your work. Rank needs. Provide content and approved access. Choose who may see data. Try agreed tasks and approve the result.

We own the agreed delivery work

Explain options. Build the chosen scope. Show progress. Record issues and changes. Supply the evidence and handoff items listed in the quote.

We decide together

Agree on success checks, dates, risks, and changes. Name the person who can approve new cost. Write down each major choice.

You can delegate tasks. Keep a clear owner for each one. A task with no owner can stop the project.

Start with the problem

A strong idea says who needs help, what is hard today, and what should become easier.

Say it in one line

“Our staff need one job list so fewer requests get lost.” This gives the project a clear purpose.

Bring a real example

Show a sample request, today’s steps, and a common problem. Remove private details from examples.

Choose a useful result

Maya will compare missed requests before and after launch. She will also count how many jobs staff can find without asking her.

Your turn: Who needs help? What is hard today? What one result would make this project worth doing?



Learn before you commit

Discovery means asking questions and checking how your work happens today.



S01 / Discovery

\$600 sample USD

What we do

Talk with the people who do the work. Review sample forms, current tools, hard cases, and the budget. Check whether a simpler tool could meet the need.

What you receive

A problem summary, a first feature list, key unknowns, and a suggested next step. The quote defines how many talks and reviews are included.

Maya's useful finding

Workers often lose signal at a job site. Offline work could change the build. She chooses an online first version and a clear fallback process.

If you skip discovery, unresolved guesses still exist. Supply a clear brief or budget for questions and possible rework.

Turn wishes into clear rules

A requirement is a written statement of what the software must do.

Estimated completion: 6-9 business days



S02 / Requirements and scope

\$800 sample USD

Too broad

“Make job tracking easy” leaves room for many different builds. It does not tell us when the work is complete.

Clear enough to check

“An owner can assign an open job to an active worker. The job keeps that worker after the page is closed and opened again.”

Include the limits

List roles, devices, fields, reports, normal steps, and error cases. Add a check for each important rule. Approve a numbered version of the list.

Maya approves R01: assign a job. Test T01 and acceptance task A01 will check that same rule.

CHOOSE YOUR FIRST VERSION

A small useful release

The first release should solve one clear problem from start to finish. It does not need every idea.

Now

Add jobs, assign workers, update status, and find past jobs. These steps solve Maya's lost-request problem.

Next

Add automatic text alerts after staff show that the main workflow works well.

Later

Consider customer payments and a separate phone app when the need and budget are clear.



Cut a feature before cutting the checks needed for the result you plan to use.

See how people will use it

Design shows where things go, what words people see, and how they move through a task.



S03 / Screen design

\$1,200 sample USD

What you receive

Screen layouts for named pages and states. States include empty lists, saved work, errors, and waiting for a response.

What you review

Can a new worker find the next step? Are labels clear? Does the phone view work? Does the page say what happened after a click?

Maya changes a label

Her team understands “New job” better than “Create record.” She fixes the wording before the page is built.

Approve the named screens and the number of design review rounds. New screens may add scope.

Make the key tasks easy to use

People may use large text, a keyboard, a screen reader, or a small phone. Plan for them early.



\$13 / Accessibility review

\$800 sample USD

Start with the basics

Use clear labels, readable contrast, useful error messages, and a visible keyboard focus. Do not use color alone to show status.

Check with people

Try key tasks with users, including people who use assistive tools when in scope. A scan alone cannot prove that a site is accessible.

Set the target

Name the pages, devices, standard target, and manual checks in the quote. A full review costs more than a small sample check.

W3C WAI recommends checking accessibility early and throughout the work. See the reading notes on page 58.

Try the flow before the build

A prototype is a model you can click. It helps you judge the steps before the full system exists.



S04 / Clickable prototype

\$600 sample USD

Try one real task

Maya opens a pretend job, assigns a worker, and returns to the list. She notices that the return step is hard to find.

Give useful feedback

Say what you tried, where you got stuck, and what you expected. Gather your team's notes into one list.

Approve the flow

A good review confirms the steps and screens. It does not prove that real data saves or that user access is safe.

If you decline a prototype, you may first discover flow problems after paid development has begun.

A demo is not a finished app

Ask what is real each time you see a demonstration. Different kinds of demos prove different things.



Screen design

Shows how the app may look. It may have no working buttons or real data.

Clickable prototype

Shows how a person moves through a task. Buttons may lead to pretend results.

Working build

Runs some or all real features. Ask which features were tested, which use sample data, and what remains unfinished.

Use these labels on every review: pictured, clickable, working, tested, accepted, or launched.

Choose how the parts fit

Architecture is the plan for the app's main parts and how they connect.

Estimated completion: 6-9 business days

S06 / Architecture workshop

\$600 sample USD

The main parts

The screen takes an action. The app applies rules. The data store keeps records. Outside services may send messages or process payments.

Choices that affect cost

Many user roles, many customer companies, offline work, and large data volumes can add work. Tell us about growth needs early.

What you receive

A simple system map, key choices, limits, and named outside services when this planning service is included.

The cheapest starting choice may cost more to change later. Ask which choices are easy to replace.

Give your data a clear home

Data means the information your app keeps, such as names, dates, and job details.



S07 / Data migration

\$800 sample USD

Define each field

For each item, say what it means, who enters it, whether it is required, and who may view or change it.

Moving old records

Data migration means moving records from an old tool into the new one. It may include mapping fields, cleaning rows, a trial import, and a final count.

Maya checks a sample

She reviews ten imported jobs before the full move. She compares totals and checks that dates and worker names still mean the same thing.

Keep a copy of the original data. Decide how to handle duplicates and missing values before the final move.

Give each person the right keys

A role is a set of allowed actions. Access rules decide which records a person can use.



\$12 / Scoped security review

\$1,000 sample USD

Make a small access table

Owner: see all jobs and assign work. Worker: see assigned jobs and update status. Guest: no job access.

Check the blocked actions

A worker should not see another worker's private job by changing a link. Access must be checked behind the screen too.

Plan staff changes

Choose who adds users, resets access, and removes former staff. Keep a second authorized recovery contact where the service allows it.

Hiding a button does not by itself protect the data behind it.

CONNECT OUTSIDE TOOLS

Make each connection clear

An integration lets your app work with another service, such as email, payments, or a calendar.



S08 / One integration

\$1,000 sample USD

Ask what moves

Which records move? In which direction? How often? What happens if the service is down or sends the same message twice?

Plan the accounts

Name who owns each account and pays each bill. Setup work and the outside provider's charges are separate costs.

Maya's choice

She keeps text alerts out of the first release. If added later, the quote must cover delivery errors, user choices, and service fees.

A connection can change when the outside provider changes its service. Ongoing repair is a support choice.

Build in small reviewable parts

Development turns the approved plan into working software. Code is the set of instructions that runs it.

S05 / Development

\$6,000 sample USD

Work in short batches

A batch may add job entry, then assignment, then status changes. Each batch should have a clear result and a review point.

Separate the spaces

Build and try changes in a test space. Keep live customer work in a separate production space. Use made-up data while practicing.

Your part

Give one clear feedback list by the agreed date. If a new idea changes the approved rules, send it through the change process.

A progress demo shows current work. It does not replace the final tests or your acceptance review.

SEE REAL PROGRESS

Ask for evidence you can use

Progress should connect a promised task to a result you can see and try.

Estimated completion: 5-8 business days



S20 / Monitoring setup

\$400 sample USD

A useful update

Done: job entry saves. Next: assign workers. Blocked: client must confirm job status names. Decision due: Friday.

A useful record

R01 is the rule. Build B03 contains the change. T01 records the check. A01 records your acceptance. The names link the story.

Watch the right signal

“Most of the code is done” is less useful than a list of finished tasks, open issues, remaining work, and budget used.

Agree how often you receive updates, where they live, and who can approve a change.

Check that the parts work

Testing tries the software and compares what happens with what should happen.



S09 / Feature testing

\$2,000 sample USD

Check a small part

Does a required field reject a blank value? Does the job status save? These checks focus on one rule at a time.

Check the full task

Add a job, assign a worker, close the page, open it again, and confirm the record. Also try invalid or missing information.

Keep the results

A useful test record says what was tried, the expected result, the actual result, and whether it passed. Fixes should be checked again.

Testing can find problems. It cannot promise that no unknown problem exists.

Check the delivery as a whole

Quality assurance, or QA, helps the team follow a sound process and review the whole release.



\$10 / Release QA review

\$800 sample USD

Testing asks

“Does this action work?” QA also asks whether the right checks were planned, recorded, and completed.

Look across the product

Review the agreed browsers, roles, key tasks, issue list, and release notes. Confirm that a fix did not break a task that worked before.

Know the team

Ask whether another person reviews the builder’s work. A separate reviewer adds a fresh view; it is included only when stated in the quote.

QA is not a certificate or a promise of perfection. Ask what was reviewed and what was outside the review.

CHOOSE YOUR CHECKS

Different checks answer different questions

A single “tested” label is too broad. Name the checks your intended use needs.



S14 / Load testing

\$800 sample USD

Function and repeat checks

Function checks test the promised behavior. Repeat checks, called regression tests, look for damage caused by later changes.

People and device checks

Usability checks ask if people can complete tasks. Accessibility checks look for barriers. Browser and device checks test named setups.

Load and security checks

Load checks try an agreed level of activity. Security checks look for ways to misuse access or data. Deeper reviews need their own scope.

If checks are declined, the result is unverified in those areas. Do not describe a raw handoff as ready for public use.

Try the work you are buying

Client acceptance means you try agreed tasks and decide whether the delivered work meets the approved scope.



S11 / Acceptance facilitation

\$400 sample USD

Use a task list

Use the agreed acceptance steps in the test space. Record pass, fail, or blocked. A blocked task is not a pass.

Report a clear issue

Include the task, steps, expected result, actual result, and a screenshot without private details. Name the build you tried.

Make a written choice

Accept, request a fix, or accept with named minor issues and due dates. Silence is not approval in this sample process.

Agree the review window in the quote. Missed client review dates may move the launch date.

TRY MAYA'S ACCEPTANCE TASK

Can a worker see the right job

This task checks the approved assignment rule and a related access rule. Use fictional test accounts.

Steps 1 and 2

Sign in as the owner. Create job J100 and assign it to Worker A. Save, close the page, and reopen the job.

Steps 3 and 4

Sign in as Worker A and find J100. Then sign in as Worker B and try to open J100 using its link.



Pass means

The assignment stays saved. Worker A can open the job. Worker B is blocked. If any part fails, record the issue and try again after the fix.

Acceptance checks agreed business tasks. It does not replace a full security review.

Build protection into each stage

Security helps protect the app and its information from misuse. Start with the risks, then choose suitable controls.



S22 / Independent security assessment

\$2,000 sample USD

Plan and build

Decide who may do what. Keep passwords and service keys out of shared files and code. Use protected connections and supported parts.

Review before launch

Check access rules, account recovery, known software issues, and secret settings. Record unresolved findings and who must resolve them.

Keep caring for it

Name who installs updates, reviews alerts, and responds to a suspected problem. A one-time review cannot cover every future change.

NIST SSDF treats secure development as work across the life cycle. A deeper independent review is a separate service. See page 58.

Plan for a bad day

Privacy concerns how information is used. Backups help recover information after loss or damage.

Estimated completion: 6-9 business days

S21 / Recovery exercise

\$600 sample USD

Collect what you need

Choose the purpose of each data field, who can use it, and when it should be removed. Decide who handles user data requests.

Define the backup plan

Agree what is copied, how often, where it is kept, who can restore it, and how much lost work is tolerable.

Try a restore

A saved copy is useful only if it can be restored. Test the agreed recovery steps in a safe space and record the result.

Business-specific legal notices and compliance reviews need a separate agreed scope and qualified advice where needed.



Open the agreed release

Launch means placing the accepted release in the live space for its intended users.

Estimated completion: 5-8 business days

Before opening

Review acceptance, unresolved issues, access, data move results, backups, account ownership, and support contacts. Name the go or hold decision maker.

During opening

Move the approved version, check settings, run key live checks, and watch for errors. Keep the agreed route back if launch goes wrong.

After opening

Watch the first real tasks. Record issues in one place. A launch service covers the agreed launch window, not unlimited future care.

A launch date is a target until required decisions and checks are complete. Record the exact release version.



\$15 / Launch service

\$600 sample USD

Know what you will receive

A handoff lets you or the next provider understand, run, and maintain the work you bought.



S16 / Operations handoff notes

\$600 sample USD

Delivery inventory

List the source files, version, setup steps, data structure, service list, open issues, and agreed design or test files.

Access and ownership

Record who owns the hosting account, domain, data, and code rights under the agreement. Third-party parts may have separate license terms.

Confirm the transfer

Check that the named owner can open the files and use the accounts. Transfer secrets through an approved secure route. Remove unneeded builder access.

Source code alone may not include hosted services, data, paid licenses, design files, or a working deployment. List each item.

Give your team a written guide

Documentation is the set of instructions and records that helps people use and care for the app.



\$17 / User guide

\$800 sample USD

User guide

Shows common tasks in plain words, with pictures when included. Example: add a job, assign it, and correct a mistake.

Admin and technical notes

Explain account setup, access, backups, settings, and how the next developer can run the code. Each type has its own scope.

If you decline

You must create the guide, hire someone else, or rely on your own knowledge. New staff may need more help. Later writing is new work.

A raw code handoff still includes a file inventory and known limits in this sample model. Full training manuals are optional.

Practice with a guide

Training helps people perform their own tasks. It can be live, recorded, or self-paced.

Estimated completion: 7-12 business days

\$18 / Live training

\$600 sample USD

Choose who learns

An owner may need account and report training. A worker may only need job updates. Avoid paying to teach every role every feature.

Make it hands-on

Use fictional records. Let each learner complete the steps, make a safe mistake, and correct it. Give them the help route.

If you decline

Assign someone to teach the team. The software delivery price does not automatically include lessons, recordings, or unlimited questions.

Agree the number of sessions, people, roles, recordings, and practice guides before training starts.



Choose who keeps it running

Support helps users with problems. Maintenance keeps the app and its parts in working order.



\$19 / Monthly support

\$500/month sample USD

Support

Agree the contact route, support hours, included time, urgent issue rules, and first-response target. Response time is not a repair promise.

Maintenance

Plan updates, backup checks, small repairs, and monitoring. Say whether unused hours roll over and how extra work is approved.

Bug care and new features

The signed agreement defines any defect repair period. New features and outside provider changes are not automatically free fixes.

If care is declined, name your own provider or owner before live use. Software still needs attention after handoff.

The build is one part of the cost

Your ongoing bills may include hosting, storage, email, text messages, paid services, and support.

Sample monthly plan

Hosting allowance \$60; email allowance \$15; support allowance \$500 for 5 hours. Sample total: \$575 per month. These are budget placeholders, not provider quotes.

Twelve-month view

The sample Launch Ready build is \$14,000. Add $12 \times \$575 = \$6,900$. First-year sample total: \$20,900, before taxes and excluded work.

Control the bill

Name the payer, set spending alerts where available, review usage, and agree what happens when limits are reached.

Usage fees and provider prices can change. Get current provider quotes before signing. No support hours are counted twice here.



What a sample price means

Sample pricing shows how choices affect a budget. It is not a quote, market average, or final company rate.



Illustrative service prices

Menu amounts are sample service prices. Hours are example work allowances. The signed quote sets actual pricing.

A shared example

Maya's app has one organization, two roles, four main screens, and no payments, offline mode, or imported history. Larger scope changes the estimate.

What to check

Every line needs a result, a limit, a price, and a payment trigger. Identify taxes, outside fees, client tasks, and any optional work separately.

SAMPLE / ILLUSTRATIVE PRICING — FINAL COMPANY RATES TO BE APPROVED

Choose the finish line you need

Each level ends at a different point. Compare the actual included services, not just the package name.

Raw Code Handoff | \$6,000 sample

For a client with an approved specification and another team to finish the work. No formal test package, launch, training, or ongoing care.

Working Pilot | \$11,000 sample

A small working build in a test space, with design, feature tests, and a guided acceptance session. No public launch.

Launch Ready | \$14,000 sample

Adds a whole-release QA review, scoped security review, launch, and operations handoff notes to the pilot.

Guided Launch | \$15,400 sample

Adds a user guide and two training sessions to Launch Ready. Optional support: \$500 per month for five hours.

All four figures use the same limited app example. See the Services Menu for exact service IDs and exclusions.



LEVEL ONE

Raw Code Handoff

You receive the agreed code files and a clear list of known limits. This is a development handoff.

Estimated completion: 14-25 business days

You supply

A complete approved specification, design decisions, access rules, and answers to build questions. Planning gaps may require a new quote.

You receive

S05 development: 60 sample hours, \$6,000. File inventory and known limits are included. Basic coding checks are not a formal test or QA service.

Work still to assign

Testing, acceptance management, security review, setup instructions, deployment, user guides, training, and ongoing care are excluded unless added.

Intended use: transfer to a technical owner. It is not represented as tested, secure, accepted, or ready for live users.



LEVEL TWO

Working Pilot

A pilot lets a small group try the main workflow with sample data in a test space.

Estimated completion: 24-43 business days

You receive

Discovery, requirements, design, development, feature testing, and acceptance facilitation. Sample service total: \$11,000.

You review

Can users complete the core jobs? What should change before wider use? Record limits and keep the pilot group small.

Pilot feedback may reveal new scope. The pilot price does not include every idea raised during the trial.



Not included

A separate clickable prototype, broad QA, dedicated security review, public launch, old-data migration, and ongoing care. No sensitive live records in this pilot example.

LEVEL THREE

Launch Ready

This level prepares the agreed small app for a controlled live opening after all listed checks pass.

Estimated completion: 30-53 business days

You receive

Working Pilot plus S10 QA (\$800), S12 security review (\$1,000), S15 launch (\$600), and S16 operations notes (\$600). Total: \$14,000.

You still decide

Confirm the real users, accounts, launch time, known minor issues, and who will provide care after launch.



Boundaries

The security review is limited to the named app and checks. Independent penetration testing, compliance certification, migration, and full training remain separate.

The package name is a goal. If launch checks fail, the release stays on hold until the blockers are resolved.

LEVEL FOUR

Guided Launch

This level adds help for the people who will use the finished release.

Estimated completion: 33-59 business days

You receive

Launch Ready (\$14,000), S17 user guide (\$800), and S18 training (\$600). Total: \$15,400 before outside fees and tax.

Optional care

Add S19 support at \$500 per month for five hours. In this sample, unused time expires monthly and extra hours need approval.

What it does not mean

There is no promise of unlimited edits, round-the-clock support, or a fixed repair time. The quote sets the actual service terms.

Choose this level when the team needs task instructions and practice as well as the software itself.



USE THE SERVICE MENU

Build a plan around your budget

The separate Services Menu shows the result, value, price, and decline effect for every service.

Keep one service ledger

Start with a package or individual lines. Mark services already included so you do not add their prices a second time.

Use four choices

Buy now. Client will do it. Another provider will do it. Defer or decline it. For each unfinished task, name an owner and a date.



Check dependencies

A build needs agreed rules. A public launch needs its planned checks and an operating owner. If you decline our service, the work may still need to happen.

A lower invoice can move work to you. It does not always remove the need for the work.

See how the total is built

Maya selects Launch Ready for the limited job tracker. All line items below come from the Services Menu.

Planning ranges: see the estimate tables at the end.

Plan and build | \$8,600

Discovery \$600 + requirements \$800 + design \$1,200 + development \$6,000.

Check and open | \$5,400

Feature testing \$2,000 + QA \$800 + acceptance \$400 + security \$1,000 + launch \$600 + operations notes \$600.

Total and reserve

Base build: \$14,000. Optional reserve for approved changes: \$1,400. Planning allowance: \$15,400. A reserve is not a fee or permission to spend.



Running costs are separate. A reserve is charged only through work you approve under the agreement.

Tie payments to clear events

A payment plan should say what is due, when it is due, and what work or approval triggers it.

Sample \$14,000 plan

30% at booking: \$4,200. 30% at approved design: \$4,200. 30% when the scoped build is ready for acceptance: \$4,200. 10% at accepted handoff: \$1,400.



Set a realistic calendar

List build time, review windows, and outside dependencies. Name the client dates for content, access, and decisions. Late inputs may move later dates.

Agree pause rules

The quote should define overdue invoices, paused work, restart timing, cancellation, and any restart fee. None of those fees is automatic in this example.

Payment does not by itself prove acceptance or authorize public launch. Record each decision separately.

Say what is in and what is out

Scope is the agreed list of work. A useful scope also says what the price does not cover.

Included

Maya's example: one organization, owner and worker roles, four main screens, job entry, assignment, status, and search.

Excluded

Payments, text alerts, offline use, a store-listed phone app, data migration, many client companies, and extra reports.

Assumptions and limits

The client supplies copy and approved access. Name supported browsers, review rounds, user volume target, record volume, and service allowances before signing.



“Small change” is a description of appearance, not a price. The team must inspect the work behind it.

CHANGE ORDERS

Approve the change before the work

A change order is a written update to the agreed scope, cost, or schedule.

Estimated completion: 5-7 business days



S23 / Change assessment

\$200 sample USD

Step 1 | Ask and assess

Describe the request and why it matters. We check the design, code, data, tests, documentation, and launch impact.

Step 2 | Choose

Receive a written estimate. Approve it, reject it, swap another feature out, or save it for later. Record any assessment fee before that work starts.

Step 3 | Confirm and deliver

The named approver accepts the revised total and dates. We update the scope, build the change, test it as agreed, and ask you to review it.

An idea in a call or message is a request. It is not permission to spend until the agreed approver says yes in writing.

The button is only one part

Maya asks for text alerts after approving the first build. This change touches more than a button.



Sample work and change price

Assess 1 hour; update design 1; build 5; test 2; update guide and deploy 1. Total: 10 hours. Sample change price: \$1,000.

Revised total

Launch Ready \$14,000 + approved change \$1,000 = \$15,000.

Message-provider setup or usage fees are excluded. Five added working days are a sample schedule change, not a promise.

Check for overlap

If S08 already covers this exact text connection, do not also add this full change estimate. Credit paid assessment time that is included in the final scope.

Only \$400 remains in a \$1,400 reserve after this \$1,000 change. Another idea needs another choice.

Name the kind of request

The right label helps decide whether the work is included or needs a change order.



Revision

A design adjustment within an included review round. Example: change the approved draft's button label before design signoff. New pages may exceed the round's scope.

Bug or defect

The build fails an approved requirement. Example: the assigned worker disappears after reload. Repair terms follow the signed scope and defect-care terms.

New feature

The client asks for behavior not in the approved scope. Example: send the worker a text after assignment. Assess and price it before starting.

If the label is disputed, compare the approved requirement, version, and evidence. Pause only the affected work while you resolve it.

Save ideas without losing control

New ideas are welcome. A simple idea list helps you choose when they should become paid work.



\$24 / Monthly improvement planning

\$300/month sample USD

Capture

Write the problem, who benefits, and why it matters. Mark it urgent, useful soon, or later. A wish list is not part of the signed build.

Review

At an agreed review point, compare value, effort, risk, and available funds. You can buy an assessment before deciding to buy the feature.

Release

Group related approved ideas into a later release. Keep a separate total, date, and acceptance list. Avoid changing the current release every day.

A fixed budget works best with flexible scope. Choose which work to remove or postpone when new work is added.

WHEN YOU DECLINE A SERVICE

Make the remaining work visible

Declining a service is a decision about responsibility as well as cost.



Record the choice

Name the service, the reason, the unfinished work, its owner, and the date it will be done or reviewed.

Name the intended use

Declined formal tests may be suitable for a raw code transfer to another team. They do not support a claim that the app is ready for customers.

Know the later cost

Later work may need fresh setup, a new review, or changes to finished screens and code. Get a new quote rather than assuming the old price still applies.

A decline record does not turn an unsafe result into a safe one or change duties that apply to the project.

CHOOSE YOUR PATH

What do you need next

Begin with your next useful result. You can move to another level later under a new or revised scope.

Still unsure what to build

Start with discovery and requirements. If the workflow is uncertain, add a design or clickable prototype.

Need a technical handoff

Choose Raw Code Handoff only if you have a complete specification and a technical owner for the unfinished work.

Need people to try it

Choose a Working Pilot with sample data and a defined review group. Use the results to plan live use.

Need to open and teach

Choose Launch Ready for the defined live release. Choose Guided Launch when your users also need a guide and hands-on help.

If your app handles high-risk or sensitive work, add a project-specific review before choosing the final scope.



YOUR SERVICE SELECTION

Fill in one shared plan

Use the fillable Decision Workbook to turn your choices into a quote request.



Choose the starting level

Write the level name, included service IDs, and its current quoted total. Use the sample price only for planning.

Add or remove services

List each change once. Mark buy, client, other provider, or defer. Name the owner and due date for work outside our scope.

Check the complete budget

Add one-time service fees and approved changes. List monthly fees separately. Keep any reserve separate from the contracted price.

Before signing, ask: “What will I receive, what will still be unfinished, and who will do that work?”

Tell us what success looks like

You can answer these questions in plain words. You do not need a technical plan to begin discovery.

People and problem

Who will use the app? What do they struggle with now? Show one normal task and one hard case.

Result and first version

What should improve? Which three tasks matter most? What can wait? What devices and outside tools are involved?

Bring examples without passwords or private customer details. Use the workbook brief page for your answers.



Budget and decisions

What can you spend now and each month? Is the date firm or preferred? Who approves scope, content, payment, and launch?

Go or hold with a reason

A launch decision should point to evidence and an owner. A green color alone is not enough.



Go

Required checks pass. The client accepts the named version. Accounts, data, support, and recovery steps have owners.

Hold

An important task fails, a review is blocked, access is wrong, or a required owner is missing. Record the fix and the next review date.

Minor issues

List each accepted minor issue, its effect, owner, and fix date. Major access, data-loss, or core-task failures stay launch blockers in this sample process.

Use the workbook to record the decision. Acceptance, payment, and launch approval are separate records.

PLAIN WORDS

Words you may hear

Ask us to explain any term that is unclear. These short meanings can help you start.

Plan words

Scope: agreed work. Requirement: a rule the app must meet.

Prototype: a model you can try.

Milestone: a named point in the project.

Build words

Source code: instructions that run the app. Database: a place that stores records. Integration: a link to another service. Production: the live app space.



Review words

QA: the process used to support quality. Acceptance: your review against agreed tasks. Defect: failure to meet an agreed rule.

Regression: a new change breaking old work.

Care words

Maintenance: work that keeps the app running. Backup: a saved copy. Restore: recover from that copy. Change order: an approved update to scope, cost, or dates.

Leave with clear answers

You should understand the work before you decide to buy it. Use this list during the proposal meeting.



The result

What problem will the first release solve? What is included, excluded, and still unknown?

The money and dates

What is due now, later, and monthly? What could change the price or the schedule?

The proof and handoff

What checks will show that it works? What will I receive? What do I or another provider still need to do?

The next step

Complete the brief and service choices. Ask for a written scope and quote. Review those documents before approving the start.

The lookbook teaches the choices. The signed scope and agreement define your actual purchase.

QUICK CHECK

Do these choices make sense

Use these four short questions to check your understanding before the proposal meeting.



1 | The demo looks good

Does that prove the app saves real data? No. Ask whether you saw a design, prototype, or working tested build.

2 | You want one more feature

Should the team build it right away? No. Record it, review the impact, and approve a change before spending.

3 | You skip the user guide

Does the team still receive training? Only if training is separately included. Name who will teach the users.

4 | You buy the raw code

Is it ready for customers? That level makes no such claim. A technical owner must handle the remaining work and checks.

If an answer is unclear, ask for a real example from your own project.

The basis for this guide

This guide explains a sample service model. It does not establish final company prices, warranties, or legal terms.

Sample model

All service amounts are illustrative prices for a fictional small job tracker. No market-rate claim is made. Quotes must define actual scope and current outside fees.

Security reading

NIST SP 800-218, Secure Software Development Framework 1.1, supports the life-cycle approach on page 28. It is background reading, not a certification of any product.

Accessibility reading

W3C WAI, Evaluating Web Accessibility Overview, supports early review and the need for human evaluation on page 14. A scan alone is not a full assessment.

Sources: csrc.nist.gov/pubs/sp/800/218/final | w3.org/WAI/test-evaluate/ • Accessed September 11, 2026



Service timing, at a glance

Services 1-12 of 24. Sample USD prices. Estimated business days after kickoff.

SERVICE	SAMPLE PRICE	ESTIMATED RANGE
S01 Discovery	\$600	6-9 days
S02 Requirements and scope	\$800	6-9 days
S03 Screen design	\$1,200	7-12 days
S04 Clickable prototype	\$600	5-8 days
S05 Development	\$6,000	14-25 days
S06 Architecture workshop	\$600	6-9 days
S07 Data migration	\$800	6-10 days
S08 One integration	\$1,000	7-12 days
S09 Feature testing	\$2,000	8-12 days
S10 Release QA review	\$800	6-9 days
S11 Acceptance facilitation	\$400	5-8 days
S12 Scoped security review	\$1,000	7-10 days

Illustrative estimates only - not a quote. Actual cost and dates are confirmed in writing. The four-to-five-day allowance is already included. Do not add buffered service ranges together for a delivery level.

Service timing, at a glance

Services 13-24 of 24. Sample USD prices. Estimated business days after kickoff.

SERVICE	SAMPLE PRICE	ESTIMATED RANGE
S13 Accessibility review	\$800	6-10 days
S14 Load testing	\$800	6-9 days
S15 Launch service	\$600	5-8 days
S16 Operations handoff notes	\$600	6-9 days
S17 User guide	\$800	6-10 days
S18 Live training	\$600	7-12 days
S19 Monthly support	\$500/mo	5-8 to activate*
S20 Monitoring setup	\$400	5-8 days
S21 Recovery exercise	\$600	6-9 days
S22 Independent security assessment	\$2,000	9-15 days
S23 Change assessment	\$200	5-7 days
S24 Monthly improvement planning	\$300/mo	5-8 to activate*

Illustrative estimates only - not a quote. Actual cost and dates are confirmed in writing. The four-to-five-day allowance is already included. *Monthly ranges cover activation; ongoing work follows the monthly agreement.

Choose the finish line

The same four delivery levels, with planning time beside each sample price.

Raw Code Handoff Estimated completion: 14-25 business days	\$6,000 sample USD
Working Pilot Estimated completion: 24-43 business days	\$11,000 sample USD
Launch Ready Estimated completion: 30-53 business days	\$14,000 sample USD
Guided Launch Estimated completion: 33-59 business days	\$15,400 sample USD

Illustrative estimates only - not a quote. Prices, work hours, and completion ranges are examples for planning purposes and may not reflect your project's actual cost or delivery time. Final pricing, scope, and schedule are determined after reviewing your requirements and are confirmed in a written project quote. Estimated completion ranges are not guaranteed delivery dates.

How to use these ranges

One consistent estimate model for the website, guides, and planning conversations.

Start with the same assumptions

Count business days after kickoff and after required materials, access, and prerequisite work are ready. Weekends and agreed holidays are excluded. The model uses the existing small-app scope and about 3-6 focused work hours per day. Client or provider waiting can extend the range.

Count the allowance once

Each displayed range already adds four business days to the low end and five to the high end. Delivery levels allow overlapping preparation and review. Extra services need a combined schedule. Monthly ranges are activation estimates; monthly hours, prices, and response terms stay separate.

Keep the decision in writing

Use the portal conversation category, document version, and Next steps owner. Request a meeting; business approval unlocks optional calendar booking. Times appear in your selected local zone. A message or recap does not approve new scope, cost, or dates.

Industry-informed inferences, not universal averages or delivery guarantees. Research: thoughtbot design sprints; Nielsen Norman Group usability studies; Atlassian estimation; GOV.UK discovery. Scope and source notes: successmarktrading.com/services.