

LEARNING GUIDE

The Software Development Process Explained for Non-Technical Founders

Six stages in plain English - discovery, design, sprints, QA, launch and maintenance - showing what a founder sees at each step, how to give feedback that speeds the build up, and the red flags that catch failing projects before they burn the budget.

6

stages in every healthy build

45%

avg budget overrun on large IT projects (McKinsey)

64%

of features rarely or never used (Standish)

10-100x

cost to fix a defect after launch (Boehm)

Sources: McKinsey large IT project research, Standish Group, PMI Pulse of the Profession, Digital.ai State of Agile, classic software engineering research (Barry Boehm), Emerson Technology case studies. Meadowbrook Dental figures are illustrative models with assumptions shown in the article. Full article:

For founders commissioning custom software for the first time - it distills the six stages, what you will see at each one and what your role is.

- **Short on time?** Skim the Key Highlights and the stage table for the ten-minute version; Stage 7 holds the red-flag list.
- **Planning?** Use the feedback table from Stage 3 in your next sprint demo, and the 90-day plan at the end to structure vendor conversations.

KEY TAKEAWAY

Custom software succeeds on process, not luck: a written scope from discovery, two-week demo visibility, real user acceptance testing, a rehearsed launch and a standing maintenance rhythm.

The scorecard

Metric	Baseline	Outlook
Metric (source)	The number	What it means for you
Budget / time / value overrun (McKinsey)	45% / 7% / 56%	Discovery discipline closes most of the gap
Rarely-used features (Standish Group)	About 64%	Fund the used 36%; cut the rest in discovery
Cost to fix a post-launch defect (Boehm)	10-100x vs design	QA is reputation insurance, not overhead
Project waste (PMI Pulse of the Profession)	About 10 cents per dollar	Process behaviors recover most of it
Agile adoption (Digital.ai State of Agile)	About 71% of organizations	Two-week demo visibility is the norm
Maintenance share of lifetime cost (literature)	60-80%	Budget 10-20% of the build per year
Meadowbrook Dental model (illustrative)	\$52K year one, 16 weeks	45% development, 15% QA, 12% maintenance

Sources: McKinsey large IT project research, Standish Group, PMI Pulse of the Profession, Digital.ai State of Agile, classic software engineering research (Barry Boehm), Emerson Technology case studies. Meadowbrook Dental figures are illustrative models with assumptions shown in the article.

Stages 1-3: Planning and building

- **1. Discovery turns wishes into scope.** McKinsey puts large IT projects 45% over budget and 56% short on predicted value - discovery exists to close that gap before code starts. The illustrative Meadowbrook build spent 2 of its 16 weeks here and cut the must-have list to shape the budget.
- **2. Design fixes the expensive decisions early.** Prototype testing killed 3 of 11 booking screens before development began, and architecture choices - like the indexed search that took a property portal from 9-second lookups to 80 milliseconds - decide how year two feels.
- **3. Sprints make progress visible.** 26 features across six two-week sprints, demoed on staging every time; agile adoption sits near 71% of organizations (Digital.ai) because two-week visibility beats one big reveal.

KEY TAKEAWAY

Plan in discovery, decide in design, verify in sprints - each stage catches one specific class of expensive mistake.

Stages 4-6 plus the red-flag list

- **4. QA protects the launch.** Structured QA catches about 93% of defects before launch versus roughly 68% without it (illustrative), while defects that reach production cost 10-100x more to fix (Boehm).
- **5. Launch is boring on purpose.** A written checklist, a rollback plan, named monitors and a soft-launch window; the logistics admin panel reached 98% SLA compliance only because onboarding was staged hub by hub.
- **6. Maintenance compounds the returns.** Maintenance runs 60-80% of lifetime cost (literature); a monthly iteration review turned the admin panel's post-launch work into a 57% workload reduction and turnover down from 31% to 12%.

KEY TAKEAWAY

Audit behaviors, not code: demos every two weeks, written scope changes, staging access, documentation and a real maintenance plan.

What you see at each stage of a healthy build

Stage	Typical duration	What you see	Your job
Discovery	1-3 weeks	Interviews, workflow maps, scoped feature list, written estimate	Answer honestly; define one success metric
Design	2-4 weeks	Wireframes, clickable prototype, tech choices in plain English	Approve flows; test the prototype with real users
Development sprints	4-12 weeks	A staging link and a demo every two weeks	Give specific feedback within 48 hours
QA and testing	2-4 weeks	Test reports, a device list, a shared bug tracker	Run user acceptance tests on real scenarios
Launch	1-2 weeks	A go-live checklist, app store review, monitoring	Be reachable; expect quiet, not fireworks
Maintenance	Ongoing	Monthly reports, updates, a change log	Review usage data; keep an improvement list

Your 90-day build-readiness plan

Fill in an owner and date for each item, then tick the box when done.

Days 1-30: Define the problem before the product

☐	Write the one-sentence problem statement and the single success metric for the build	Owner:	Due:
☐	List the top ten must-have features, then cut to six	Owner:	Due:
☐	Collect three quotes that each include a discovery phase with a written scope	Owner:	Due:

Days 31-60: Test the process, not just the pitch

☐	Ask shortlisted vendors for a demo cadence, a staging policy and a sample sprint report	Owner:	Due:
☐	Call two past clients and ask what launch week actually felt like	Owner:	Due:
☐	Agree the feedback rules: 48-hour responses, behaviors not adjectives, written scope changes	Owner:	Due:

Days 61-90: Start small and demand visibility

☐	Kick off a scoped first release with the success metric on the wall	Owner:	Due:
☐	Book UAT into your calendar and recruit three to five testers in advance	Owner:	Due:
☐	Set the maintenance plan at kickoff: monthly review, improvement budget, monitoring owners	Owner:	Due:

Your notes

Notes: what is the one success metric for your build, and which red flag will you watch for first?

TURN THE ARTICLE INTO YOUR BUILD PLAN

We run builds through these six stages every week - scoping, sprints, QA and launch included. Bring your idea and leave with a scoped plan, a realistic budget range and a process you can hold us to.

Book a free strategy call

emersontechnology.com/booking

Send us a message

emersontechnology.com/contact

Email us directly

support@emersontechnology.com
Read the full article with all data graphics:

emersontechnology.com/blog/software-development-process-expl