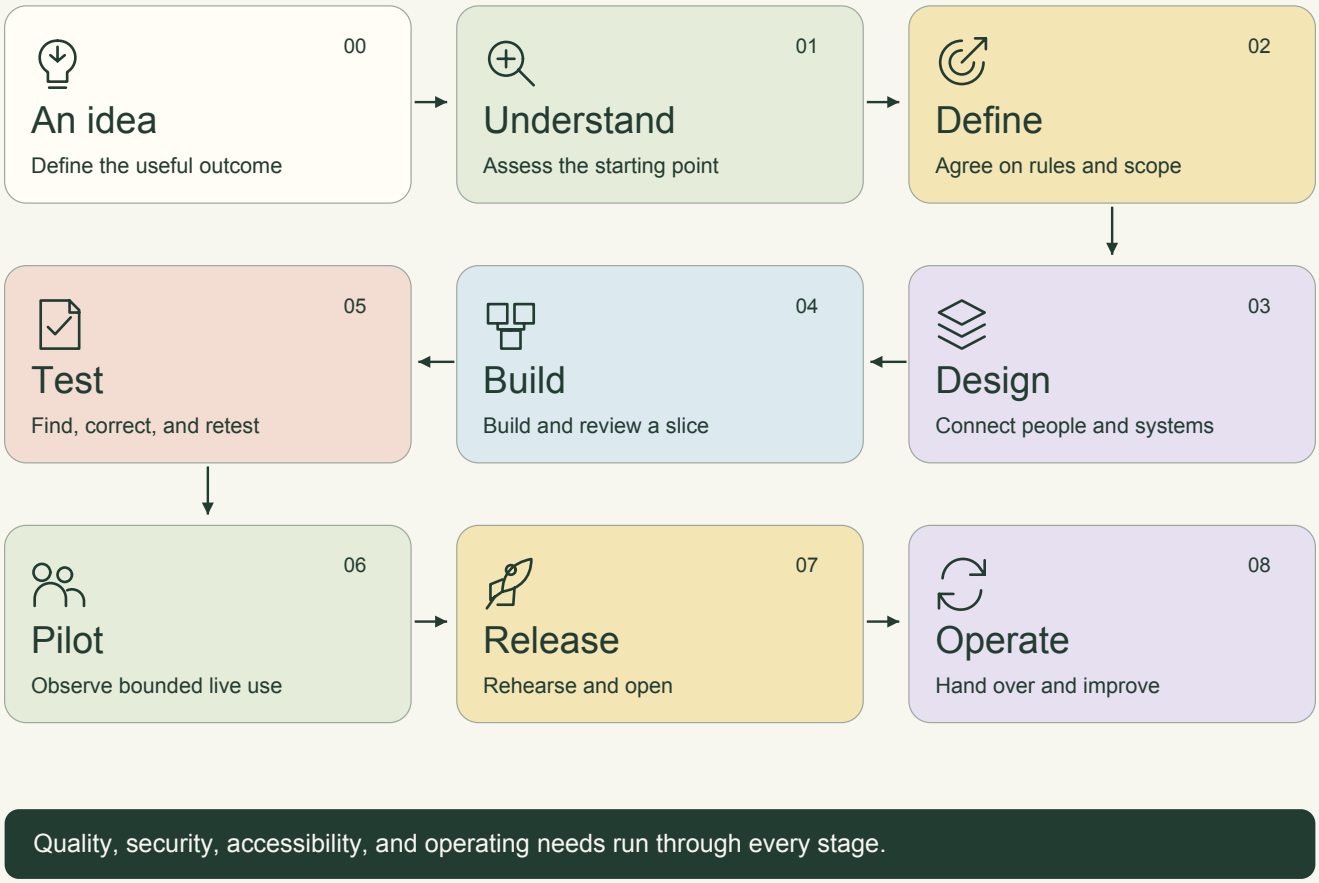
THE WHOLE PICTURE

From an idea to a working product

A visual route through the decisions, working results, and checks. Start with the work someone needs to finish.

A ROUTE THROUGH THE WORK



Quality, security, accessibility, and operating needs run through every stage.

The route is iterative. New evidence can reopen scope, design, or an earlier decision. Use the following pages to see what each stage needs and produces.

BEFORE STAGE 01

Start with the work someone needs to finish

An app idea becomes useful when you can describe who needs it, what they are trying to do, and where that work breaks down today.

01

A person

Who needs to finish the work?

02

A difficulty

What happens today?

03

A useful outcome

What should become possible?

1. Watch a recent task from beginning to end. Ask what the person did, where they waited, and what they had to repeat.

2. Describe one complete useful outcome before listing screens or features.

3. Collect constraints, existing tools, and unanswered questions. Compare improving the current process, using an existing product, and building software.

WHAT YOU LEAVE WITH

A clear problem, intended users, a first outcome, and questions to investigate.

BEFORE MOVING FORWARD

You can explain the work in ordinary language and identify people who can confirm or correct your understanding.

At the studios, a customer needs a confirmed visit and staff need one trustworthy schedule. A colorful calendar alone cannot provide either result.

STAGE 01 / CHAPTERS 1,2,3,4

Understand the starting point

Look at the current work, prototype, dependencies, and failures before choosing the next move.

STAGE 01 / HOW THE WORK MOVES

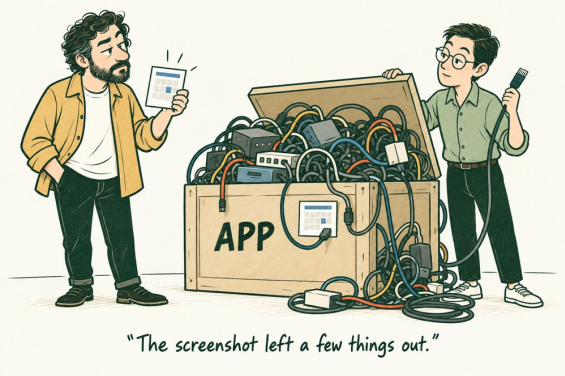
INPUT	WORK	RESULT
Starting material Work, files, access	Investigation Reproduce and compare	An informed choice Reuse, repair, or replace

A new finding can send the work back to an earlier decision.

1. Show a complete task and a recent failure. Bring the prototype, source files, and the people who know the work.

2. Have the team run a preserved copy, identify connected services, and separate observed failures from untested concerns.

3. Compare options with their consequences. Give each unanswered question an owner and a next check.



The screen is only the visible part of the product.

WHAT YOU LEAVE WITH

An assessment, a system map, a failure log, and a reasoned next step.

BEFORE MOVING FORWARD

The next assignment has a question, inputs, an owner, a deliverable, and a way to accept it.

AT THE STUDIOS

The team preserves the booking prototype, reproduces its failures, and establishes which source archive matches the deployed application.

STAGE 02 / CHAPTERS 5,6,7,8

Agree on the result

Turn observed needs into a bounded release, explicit rules, and checkable outcomes.

STAGE 02 / HOW THE WORK MOVES

INPUT	WORK	RESULT
Observed work People and constraints	Rules and scope Choose the first release	Acceptance criteria Define the expected result

A new finding can send the work back to an earlier decision.

1. Study actual customer and staff tasks, including exceptions. Resolve differences between studio rules.

2. Choose one useful release boundary. Keep deferred requests with reasons for waiting.

3. Describe success, failure, timing boundaries, and repeated actions. Estimate the selected work with assumptions and dependencies.

A small-looking action can hide a long chain of work.

WHAT YOU LEAVE WITH

A release map, agreed rules, acceptance criteria, and an engineering estimate.

BEFORE MOVING FORWARD

The chosen path reaches a useful outcome. Decisions still missing are visible beside the work they block.

AT THE STUDIOS

A cancellation is eligible exactly at its saved policy cutoff. An unconfirmed hold expires exactly at its deadline. Each rule needs its own check.

STAGE 03 / CHAPTERS 9,10,11,12

Design the system

Connect the user journey, interface states, data, and external services before building them together.

STAGE 03 / HOW THE WORK MOVES

INPUT	WORK	RESULT
User journey Actions and screen states	System design Data and responsibilities	Agreed contracts How parts work together

A new finding can send the work back to an earlier decision.

1. Map what people see while an action succeeds, waits, or fails. Include keyboard use and recovery from interruption.

2. Define who owns each record and how conflicting changes are handled. Compare structures that fit the team and operating needs.

3. Trace each external exchange, repeated result, and uncertain outcome. Assign both automatic handling and human follow-up.

More separate components also mean more connections to operate.

WHAT YOU LEAVE WITH

A system diagram, data and state models, interface designs, and integration rules.

BEFORE MOVING FORWARD

The design explains ownership and what happens on conflict, expiry, interruption, and repetition.

AT THE STUDIOS

A short protected change claims the provider interval. The customer then pays while a saved hold exists; the database is not locked for ten minutes.

STAGE 04 / CHAPTERS 13,14,15,16

Build working components

Build in small slices. Keep the requirement, changed files, and actual check results together.

STAGE 04 / HOW THE WORK MOVES

INPUT	WORK	RESULT
A bounded task Criteria and dependencies	Build and review Connect the working slice	An identified version Run the affected checks

A new finding can send the work back to an earlier decision.

1. Break the release into complete, testable outcomes. Make dependencies and missing access visible.

2. Implement, review, and connect the interface, server, database, and provider integrations.

3. Demonstrate against the agreed criteria. Record the checked version and the impact of new requests.

Written, reviewed, checked, and released are different states.

WHAT YOU LEAVE WITH

Repeatable working slices, reviewed changes, actual check results, and visible unfinished work.

BEFORE MOVING FORWARD

The demonstrated outcome meets the shared quality conditions on an identified build.

AT THE STUDIOS

A repeated payment result must not create an extra confirmation action. The corrected version is checked again, with the result retained.

STAGE 05 / CHAPTERS 17,18

Test quality and risk

Test the whole outcome, its boundaries, and its failure paths. Check the saved state as well as the message.

STAGE 05 / HOW THE WORK MOVES

INPUT	WORK	RESULT
An identified build Expected outcomes	Tests and observation Failures, access, workload	Fix and retest Check neighboring behavior

A new finding can send the work back to an earlier decision.

1. Exercise ordinary paths, overlaps, exact boundaries, delayed and repeated events, and lost responses.

2. Check permitted and forbidden actions on the server. Observe usability, accessibility, and behavior under the intended workload.

3. Reproduce defects, correct them, and repeat the failed and neighboring checks on the new version.

A success message cannot tell you everything that changed.

WHAT YOU LEAVE WITH

Versioned results, explained defects, and remaining risks with owners and workable restrictions.

BEFORE MOVING FORWARD

Critical behavior has evidence for the intended next use; any remaining restriction can actually be maintained.

AT THE STUDIOS

A reschedule appears successful, but the old provider interval stays occupied. Looking at saved state exposes the defect that the success message misses.

STAGE 06 / CHAPTERS 19,20

Pilot and accept

Observe a bounded live use, learn from the work, and separate software acceptance from permission to expand.

STAGE 06 / HOW THE WORK MOVES

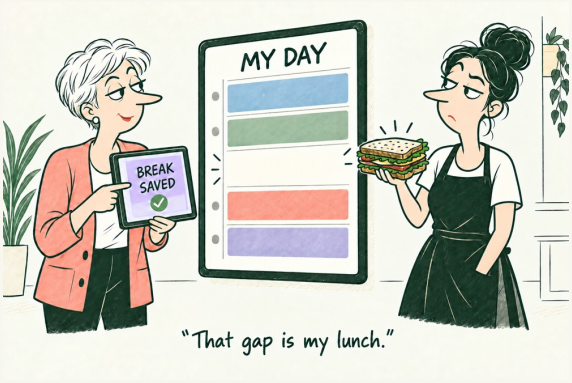
INPUT	WORK	RESULT
A bounded audience Coverage and stop rules	Observed use Real tasks and exceptions	Acceptance decision Corrections and next scope

A new finding can send the work back to an earlier decision.

1. Choose participants, provider/date inventory, support coverage, and conditions for stopping or pausing enrollment.

2. Keep one process authorized to write the transferred schedule. Observe complete tasks and record any help people need.

3. Correct findings and inspect the current candidate. Make acceptance and opening decisions from their own evidence.



Actual work reveals what a tidy demonstration can miss.

WHAT YOU LEAVE WITH

A pilot record, observations, corrections, acceptance results, and explicit operating gaps.

BEFORE MOVING FORWARD

The next audience has the required protection and coverage. Existing commitments remain supported.

AT THE STUDIOS

The service provider's lunch break reveals a problem in the daily view. A bounded pilot gives the team a chance to correct it before wider use.

STAGE 07 / CHAPTERS 21,22

Release version 1.0

Rehearse the operating response, protect existing records, and open the checked version under explicit conditions.

STAGE 07 / HOW THE WORK MOVES

INPUT	WORK	RESULT
Operating rehearsal Alerts, access, recovery	Controlled transfer Preserve accepted work	Opening and observation Named people on duty

A new finding can send the work back to an earlier decision.

1. Send an alert through the actual contact route, including acknowledgment and backup coverage. Rehearse recovery with current records and external effects.

2. Identify the release candidate, configuration, transfer point, and compatible return path. Prevent competing writes during the transition.

3. Run opening checks, watch the first hours, and keep unfinished actions and existing customer commitments visible.

A backup becomes useful when the recovery procedure works.

WHAT YOU LEAVE WITH

A release record, checked operating procedures, assigned coverage, and follow-up work.

BEFORE MOVING FORWARD

Both the identified version and its operating route pass the checks required for opening.

AT THE STUDIOS

The team's first recovery takes four hours against a one-hour target. After correction, the same complete rehearsal takes 48 minutes.

STAGE 08 / CHAPTERS 23

Take over and improve the product

Transfer usable knowledge and responsibility. Feed operating evidence into the next release.

STAGE 08 / HOW THE WORK MOVES

INPUT	WORK	RESULT
A usable handoff Code, access, instructions	Daily operation Support and observation	The next improvement Return to rules and scope

A new finding can send the work back to an earlier decision.

1. Have a capable receiving operator start and restore the product using the handoff package and their own authorized access.

2. Assign recurring technical and studio work. Keep support instructions available when the application is down.

3. Review observed use and unfinished work. Choose the next outcome, update its rules, and repeat the affected design and checks.

The operating knowledge needs to stay when its author leaves.

WHAT YOU LEAVE WITH

A working handoff, acknowledged support coverage, current restrictions, and a prioritized next release.

BEFORE MOVING FORWARD

The receiving side can operate the product without relying on the original developer's memory.

AT THE STUDIOS

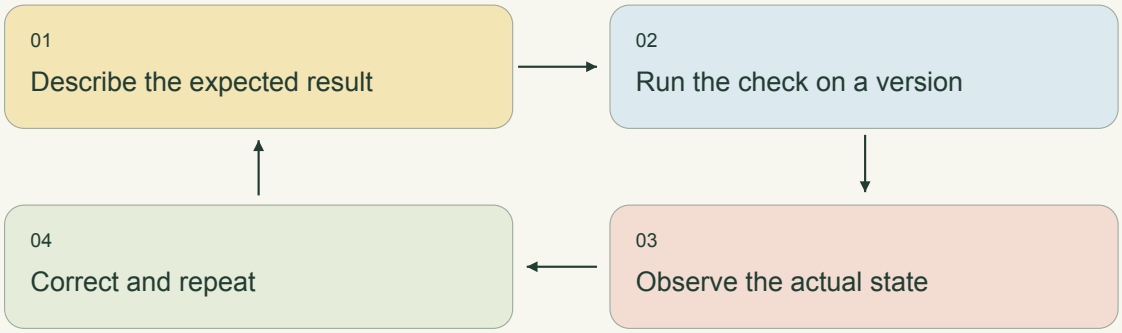
The handoff succeeds when another operator can follow it and recover the identified state. A folder of files alone does not establish that result.

QUALITY THROUGHOUT

Testing is a loop through the whole project

Start with a checkable promise. Every correction returns to the failed check and to the behavior it could have affected.

MAKE THE CHECK A LOOP



Before code

Check the need, scope, rule, and expected outcome.

During each change

Review the change. Check the component and its connections.

Before live use

Exercise the full journey, access restrictions, workload, and operating response.

After opening

Observe the service. Investigate failures and use them to improve the next version.

THREE DIFFERENT DECISIONS

Works in a check

The identified version met the stated criterion.

Accepted for its scope

The Client accepted the agreed software outcome.

Ready to open

Live-use protection, coverage, and operating procedures are ready.

THE WORK CONTINUES

The same work, with different people doing it

Working with AI changes how you organize the work. It does not remove the need to investigate, decide, inspect, test, and operate.

01 **Keep the context**

Preserve current files, rules, decisions, versions, and open questions.

02 **Request one change**

Give the task its inputs, expected result, and boundaries.

03 **Inspect the outcome**

Read the change and run checks on the actual version.

04 **Bring in competence**

Have unresolved critical behavior examined by someone able to assess it.

WITH A TEAM

The professional-team route reaches a controlled general release with its own operating evidence.

REWIND / WORKING WITH AI

Rewind reaches a bounded Studio A pilot with its own review, recovery rehearsal, and support coverage. Its results belong to that installation.

THE PEOPLE

Build the team around the work.

A role names a function. Agree on the person, their competence, their available time, and the result they own.

Client

Senior Administrator

Analyst

Tech Lead

Designer

Tester

COVER THE RESPONSIBILITIES

The Client chooses product priorities. Contributors bring designs, working software, and evidence. Someone must own the service after opening. Roles can overlap; an unassigned responsibility still needs an owner.

The next pages show three team compositions, every role card, and the contributions and handoffs at each stage.

TEAM COMPOSITION

Choose the functions your product needs.

Start with the work and its risks. Assign an owner to each function, then decide which responsibilities can be combined.

A focused browser product

A bounded workflow, a small set of users, and few external connections.

Client and studio representative / Analysis and delivery coordination / Design and browser/server implementation / Testing, release, and operating cover

The Analyst can coordinate delivery; a Designer can cover UX and UI; a Full-Stack Developer can cover browser and server work. Assign testing and operating ownership explicitly.

Reconsider: Separate a function when its workload, risk, or missing expertise is slowing or weakening the work.

A service with consequential integrations

Payments, shared inventory, several staff roles, or work that must survive failures.

Client, Senior Administrator, and Project Manager / Analyst, Designer, and Tech Lead / Frontend and Backend Developers / Tester, DevOps, and Security Engineer

These are functions to cover. The Tech Lead may also implement server work or architecture; the team still needs time and competence for permission checks, payment behavior, and recovery.

Reconsider: Bring in database or reliability expertise when data changes, recovery, capacity, or operating needs exceed the team's experience.

A product with a mobile build

The chosen delivery path needs device capabilities or a distributed mobile app.

Product decisions, analysis, and user research / Design and Mobile Developer / Server and integration work where needed / Device testing, distribution, support, and operation

Keep shared product rules under one owner. Mobile and server contributors agree on messages, offline behavior, and version compatibility.

Reconsider: A responsive browser product does not itself require a separate Mobile Developer. Choose the delivery path before staffing it.

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

One useful outcome. The feature tower can wait.

Client

Chooses the outcome, scope, operating restrictions, and opening decision. Supplies the studios' rules and asks for evidence without taking over engineering implementation.

OUTPUT

Approved priorities, timely decisions, accepted results, and named owners for the Client side of the work.

ASK: Which decisions do I need to make this week?

Stages: 01, 02, 03, 04, 05, 06, 07, 08

The calendar has two bookings. The studio still has one chair.

Senior Administrator

Represents staff work across the studios. Brings exception cases, checks policy and screen meaning, owns schedule handovers, and makes unfinished customer work usable for staff.

OUTPUT

Working cases, staff acceptance observations, inventory handovers, and assigned follow-up on unfinished visits, payments, and reminders.

ASK: Which arrangement is current when a move fails?

Stages: 01, 02, 03, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

The date is waiting for three decisions.

Project Manager

Connects decisions, dependencies, contributors, and review points. A blocked task must have a next action; a release meeting must have evidence to decide from.

OUTPUT

Plan, status, risk list, and agreed working process.

ASK: What currently determines the schedule, and which dependencies matter?

Stages: 01, 02, 03, 04, 05, 06, 07, 08

Every extra bridge needs someone to maintain it.

Tech Lead

Owns the technical approach and its connections. Also acts as Architect, comparing the operating costs of separate services with the shared application.

OUTPUT

Agreed engineering decisions and a technical plan.

ASK: Which technical risks matter now?

Stages: 01, 02, 03, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

One simple rule. A surprisingly long list of exceptions.

Analyst

Turns working cases into explicit rules, scope, states, and criteria. Also performs product work: which difficulty matters and what belongs in the release.

OUTPUT

Current scope, rules, examples, acceptance criteria, and open questions with decision owners.

ASK: Which statement came from an observed case, and which is an assumption?

Stages: 01, 02, 03, 04, 05, 06

The screen is beautiful. Where does the person go next?

Designer

Works on the user flow and interface together. Tests what people understand, then specifies states, content, and interactions for implementation.

OUTPUT

A flow and state record with content, allowed actions, accessibility behavior, and observation results.

ASK: What does a person understand while this action is waiting?

Stages: 01, 02, 03, 04, 05, 06

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

A waiting spinner is not a confirmed booking.

Frontend Developer

Builds working browser behavior and connects it to server results. Also owns UI engineering for reusable components, keyboard behavior, and status display.

OUTPUT

A working interface with agreed behavior.

ASK: Which browsers and devices are supported?

Stages: 03, 04, 05, 06, 07, 08

Two requests arrive. One slot is available.

Backend Developer

Implements shared booking operations and data changes, payment result handling, and durable pending work. A successful response must agree with saved records.

OUTPUT

Server functions and data exchanges that support the scenarios.

ASK: What happens when two actions occur together?

Stages: 02, 03, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

A green checkmark still needs a closer look.

Tester

Turns requirements into reproducible checks, finds differences between expected and actual results, and repeats checks after corrections on the candidate being considered.

OUTPUT

Test plan and results, with reproducible bug reports.

ASK: Which requirements and risks do the checks cover?

Stages: 01, 02, 03, 04, 05, 06, 07, 08

Before going up, check the way back.

DevOps Engineer

Prepares environments, release controls, monitoring routes, and recovery. Tests whether an operator can reach the problem and restore usable service.

OUTPUT

Identified environments and release path, actionable alerts, compatible return routes, and a rehearsed recovery procedure.

ASK: Who owns the resources and manages access?

Stages: 03, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

The locked button has an unlocked side entrance.

Security Engineer

Examines direct requests against role and record permissions, then checks that forbidden operations fail and legitimate work remains possible.

OUTPUT

Threat model, requirements, assessment findings, and risk-remediation priorities.

ASK: Which data and operations need particular protection?

Stages: 01, 02, 03, 04, 05, 06, 07, 08

Each piece passed. Now put them together.

Technical Consultant

Reproduces the Rewind build, exposes the missing expiry/payment combination, reviews its correction, and checks critical behavior and operating preparation before a bounded pilot.

OUTPUT

Reproduced findings, bounded correction or review work, retest evidence, and the remaining opening conditions.

ASK: What will you examine first, and why?

Stages: 04, 05, 06, 07

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

The next feature needs a reason to exist.

Product Manager

Chooses product outcomes and priorities from user needs and evidence. Agree how that authority relates to the Client.

OUTPUT

Product goal, hypotheses, priorities, and success criteria.

ASK: Which problem will the first version address?

Stages: 01, 02, 06, 08

Urgent is not a place in the queue.

Product Owner

In Scrum, owns effective Product Backlog management and value decisions. Establish whether the Client retains or delegates each decision.

OUTPUT

An ordered backlog and a clear Product Goal.

ASK: Who can reorder the work?

Stages: 02, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

The process was straight until the first exception.

Business Analyst

Describes work rules and consequences with the people who perform the work.

OUTPUT

Process description, business rules, and requirements.

ASK: Which exceptions and disputed cases remain?

Stages: 01, 02, 03, 05, 06

Both systems understood the message differently.

Systems Analyst

Specifies system interactions, data, states, and boundary behavior so implementations agree.

OUTPUT

System requirements, state diagrams, and exchange descriptions.

ASK: What happens on retry, cancellation, or failure?

Stages: 02, 03, 04, 05

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

Watch the route people actually take.

User Researcher

Plans interviews and observations, preserves what happened, and distinguishes findings from interpretation.

OUTPUT

Observations, findings, and questions for further research.

ASK: Who are we studying, and why these participants?

Stages: 01, 02, 03, 06, 08

A short task should not need an expedition.

UX Designer

Designs the path through a task and checks whether people understand and complete it.

OUTPUT

Flows, prototypes, and tested interaction scenarios.

ASK: How does the user complete the main task?

Stages: 01, 02, 03, 05, 06

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

The button exists. Finding it is another task.

UI Designer

Designs readable hierarchy, controls, content, and visual states.

OUTPUT

Mockups, components, and state-presentation rules.

ASK: Which devices and screen sizes are covered?

Stages: 03, 04, 05

A component has to work from the keyboard, too.

UI Engineer

Implements reusable interface components, focus, keyboard behavior, and states; agrees on who connects them to live data.

OUTPUT

Working components and usage rules aligned with design and frontend implementation.

ASK: What do you implement, and what remains with the Designer and frontend team?

Stages: 03, 04, 05

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

The tiny product does not need a castle.

Software Architect

Compares structures and their consequences for data, delivery, and operation. In the story, the Tech Lead performs this function.

OUTPUT

Component diagram, key decisions, and their reasons.

ASK: Why does this option fit our scale?

Stages: 01, 02, 03, 04, 07, 08

Both ends of the feature have to agree.

Full-Stack Developer

Implements browser and server work. Breadth of title does not establish equal depth in database, security, testing, and operation.

OUTPUT

End-to-end functions connecting the interface and server.

ASK: Which product components are your responsibility?

Stages: 03, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

It fits this phone. What about the others?

Mobile Developer

Implements device and platform behavior and its releases. A browser app alone does not require a separate mobile developer.

OUTPUT

A mobile build prepared for the chosen distribution method.

ASK: Which devices and OS versions do we support?

Stages: 03, 04, 05, 06, 07, 08

A useful check makes failure visible.

Test Automation Engineer

Builds dependable automated checks and test data; preserves useful failures and investigates misleading passes.

OUTPUT

Automated checks with understandable results and a maintenance owner.

ASK: Which important flows have automated checks?

Stages: 02, 04, 05, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

A backup earns its place when you can restore it.

Data and Database Specialist

Designs data storage, queries, migrations, backup, and recovery. A DBA focuses on database operation; a data engineer may focus on data flows.

OUTPUT

Data model or pipelines, migration plan, and a tested recovery procedure.

ASK: Where is the authoritative data source?

Stages: 02, 03, 04, 05, 07, 08

Which alarm tells someone what to do?

Site Reliability Engineer

Turns reliability needs into measures, response practices, capacity work, and recovery exercises.

OUTPUT

Monitoring, response rules, readiness assessment, and runbooks.

ASK: How do we detect a problem before complaints build up?

Stages: 02, 03, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

A case needs a next owner, not another inbox.

Support Specialist

Receives cases, follows authorized procedures, preserves references, and escalates work beyond those procedures.

OUTPUT

A working support process and escalation path.

ASK: What information do we collect when a user reports a problem?

Stages: 05, 06, 07, 08

The meeting ended. Did the obstacle move?

Scrum Master

Helps a Scrum team use and improve Scrum. This accountability is not automatically the Project Manager or a required position on every project.

OUTPUT

A working meeting cadence, visible impediments, and action on them.

ASK: What is the meeting cadence, and where am I needed?

Stages: 02, 03, 04, 05, 06, 07, 08

ROLE CARDS

The people and their work.

Use the role to agree on a responsibility, a useful result, and the question you need answered.

A rising line needs a meaningful measure.

Product Analyst

Defines and examines product-use measures with their events, denominators, and observation periods.

OUTPUT

Measures, reports, and findings for decisions.

ASK: Which events and measures are we collecting?

Stages: 01, 02, 06, 08

Two more distinctions

Duty engineer is an operating assignment. Someone must acknowledge the alert, act or escalate, and hand the case to the next responsible person. Name that owner and their coverage.

AI can assist with analysis, design, implementation, and review. A role prompt focuses the task; it does not supply independent expertise. The Client checks outcomes and brings critical unknowns to a qualified person.

TEAM AT STAGE 01 / UNDERSTAND

Discover what the work really needs.

Who contributes, what they hand over, and whose decision moves the work forward.

Client

Names the useful outcome and priorities.

Senior Administrator

Shows real staff work and exceptions.

Analyst

Separates observed cases from assumptions.

Tech Lead

Inspects the prototype and technical risks.

Tester

Reproduces the failures on a named version.

THE HANDOFF

Current cases + prototype evidence > an agreed starting point

The Analyst joins working cases with the Tech Lead's inventory and the Tester's evidence. The Client decides which difficulty the next stage must address.

Work with: Project Manager, Designer, Security Engineer.

Additional expertise when needed: User Researcher, Business Analyst, Product Manager.

TEAM AT STAGE 02 / DEFINE

Turn the need into decisions the team can use.

Who contributes, what they hand over, and whose decision moves the work forward.

Client

Approves priorities and operating rules.

Analyst

Writes scope, states, exceptions, and acceptance criteria.

Tech Lead

Checks feasibility and dependencies.

Designer

Makes the main task and unclear states visible.

Tester

Turns each promise into a checkable example.

THE HANDOFF

Approved rules + acceptance criteria > design and test planning

The Client settles product decisions. The Analyst passes approved rules and open questions to design, implementation, and testing.

Work with: Project Manager, Senior Administrator, Backend Developer, Security Engineer.

Additional expertise when needed: Product Owner, Systems Analyst, Business Analyst.

TEAM AT STAGE 03 / DESIGN

Connect the screens, rules, and system.

Who contributes, what they hand over, and whose decision moves the work forward.

Designer

Specifies flow, controls, content, and failure states.

Tech Lead

Chooses components and records the tradeoffs.

Frontend Developer

Checks how states and data reach the interface.

Backend Developer

Defines shared operations and data changes.

DevOps Engineer

Checks environments, deployment, and recovery needs.

THE HANDOFF

Flow + states + component decisions > implementable work

The Designer hands interaction states to the Frontend Developer. The Tech Lead and Backend Developer agree on operations; the Tester checks that the design can satisfy the criteria.

Work with: Analyst, Tester, Security Engineer, Senior Administrator.

Additional expertise when needed: UX Designer, UI Designer, UI Engineer, Software Architect, Data and Database Specialist.

TEAM AT STAGE 04 / BUILD

Build a complete path through the product.

Who contributes, what they hand over, and whose decision moves the work forward.

Frontend Developer

Builds browser behavior and shows actual server outcomes.

Backend Developer

Implements rules, permissions, and durable work.

Tech Lead

Reviews how the pieces fit together.

Tester

Checks each integrated change and reports reproducible failures.

DevOps Engineer

Maintains repeatable builds and test environments.

THE HANDOFF

Integrated change > observed result > correction and repeat

Developers integrate one usable path. The Tester checks the named build, then sends any difference back with conditions and records.

Work with: Designer, Analyst, Project Manager, Security Engineer.

Additional expertise when needed: UI Engineer, Full-Stack Developer, Mobile Developer, Test Automation Engineer, Systems Analyst.

TEAM AT STAGE 05 / TEST

Test the promises and the costly failures.

Who contributes, what they hand over, and whose decision moves the work forward.

Tester

Runs the agreed checks and records scope and results.

Security Engineer

Tests forbidden and permitted operations directly.

Backend Developer

Corrects server and data failures.

Frontend Developer

Corrects interface behavior and status messages.

DevOps Engineer

Rehearses response and recovery.

THE HANDOFF

Test evidence + open issues > acceptance discussion

Findings return to the owner of the rule, design, implementation, or operating procedure. The Tester repeats affected checks on the corrected candidate.

Work with: Tech Lead, Analyst, Designer, Senior Administrator.

Additional expertise when needed: Test Automation Engineer, Data and Database Specialist, Site Reliability Engineer.

TEAM AT STAGE 06 / PILOT

Watch real work within a controlled boundary.

Who contributes, what they hand over, and whose decision moves the work forward.

Senior Administrator

Coordinates staff work and records misunderstandings.

Designer

Observes whether people can finish without coaching.

Analyst

Distinguishes a confusing screen from an incorrect rule.

Tester

Rechecks corrections on the pilot candidate.

Client

Accepts the agreed results and decides priorities.

THE HANDOFF

Pilot observations + repeat checks > accepted scope

Customers and service providers try the tasks. The Senior Administrator brings observations back; the team corrects the cause and repeats the affected checks.

Work with: Project Manager, Tech Lead, Frontend Developer, Backend Developer, Support Specialist.

Additional expertise when needed: User Researcher, UX Designer, Product Analyst.

TEAM AT STAGE 07 / RELEASE

Make the opening decision with operating evidence.

Who contributes, what they hand over, and whose decision moves the work forward.

Client

Decides whether the agreed opening conditions are met.

Tech Lead

Brings technical readiness and remaining issues.

Tester

Identifies the verified candidate and unresolved checks.

DevOps Engineer

Executes the release, recovery, and response preparation.

Senior Administrator

Owns the studio handover and unfinished customer work.

THE HANDOFF

Accepted candidate + operating readiness > controlled opening

The Client makes the opening decision from the evidence. The release owner deploys the agreed candidate; technical and studio responders take over their assigned work.

Work with: Project Manager, Backend Developer, Frontend Developer, Security Engineer, Support Specialist.

Additional expertise when needed: Site Reliability Engineer, Data and Database Specialist, Mobile Developer.

TEAM AT STAGE 08 / OPERATE

Keep the service usable and learn from it.

Who contributes, what they hand over, and whose decision moves the work forward.

Senior Administrator

Routes unfinished work and staff exceptions.

DevOps Engineer

Monitors, responds, restores, and hands over incidents.

Support Specialist

Captures useful reports and escalates beyond its authority.

Client

Chooses the next improvement from observed needs.

Project Manager

Connects new work to owners, decisions, and review points.

THE HANDOFF

Observed use + incidents > prioritized work > another checked change

Support and operating observations become owned changes. New changes return through the relevant rule, design, build, and test steps before release.

Work with: Tech Lead, Backend Developer, Frontend Developer, Tester, Security Engineer.

Additional expertise when needed: Site Reliability Engineer, Product Analyst, Product Manager, Data and Database Specialist.