

PRACTICAL AUTOMATION GUIDE

Agent handoff

A practical guide to the places automations break most often - where information, responsibility or control passes between systems and people.

Reliable steps still fail at weak handoffs.

Define what crosses the boundary, how completion is confirmed and who takes over when the next step cannot begin.

01

Name the boundary.

Show exactly where one tool, person or process hands work to another.

02

Confirm the transfer.

A sent record is not the same as a received and usable one.

03

Plan the exception.

Decide who is alerted and what safe recovery looks like.

The baton pass deserves as much design as the run.

[Use this guide →](#)

MISSION & SCOPE

Automation usually breaks between the steps.

A well-defined action can work perfectly while the whole process fails because the next system received the wrong format, the wrong status or nothing at all.

THIS GUIDE IS USEFUL IF

- Work crosses between tools or teams.
- Failures appear only after a customer notices.
- People assume the previous step completed.

THIS GUIDE WILL NOT

- Diagnose a specific vendor integration.
- Promise that connected systems never change.
- Replace testing, monitoring or recovery ownership.

How handoffs work

01

Prepare

Create the record, message, file or status the next step needs.

02

Transfer

Move it through a defined connection and expected format.

03

Confirm

Prove the receiving step accepted and can use it.

**A relay is rarely lost in the middle of the run.
The weak moment is the baton pass.**

Good individual steps do not create a reliable process unless the boundaries between them are observable, confirmed and recoverable.

BOUNDARY INSPECTION

Six tests for every handoff.

Inspect each boundary separately. One workflow may contain several different failure modes.

Test the baton pass.

A transfer is complete only when the next step has received usable information and can continue safely.

01

Payload

What exact information must cross the boundary?

02

Format

Can the receiver interpret every required field?

03

Confirmation

What proves the next step accepted the transfer?

04

Timeout

How long before silence is treated as a failure?

05

Alert

Who learns that the handoff did not complete?

06

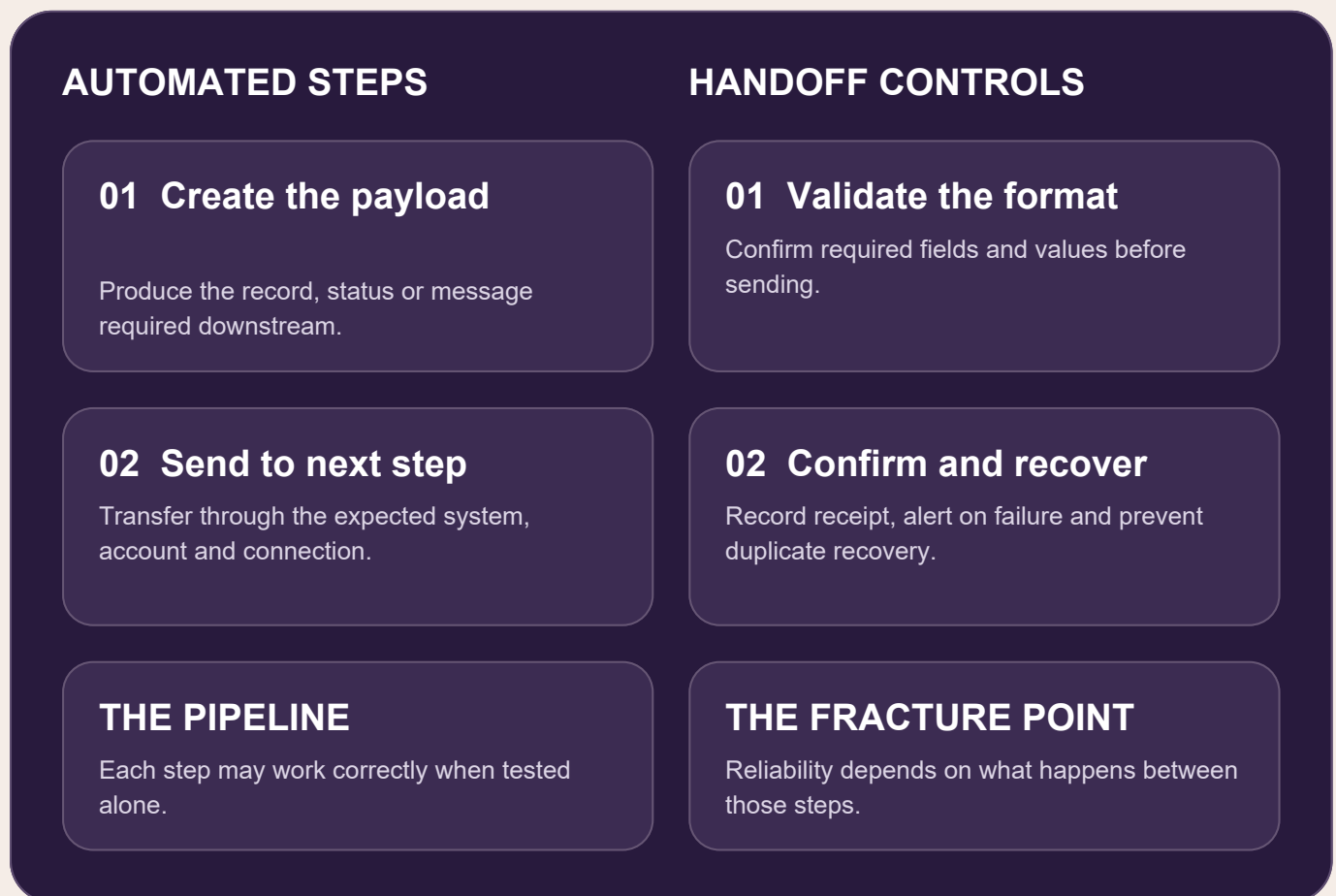
Recovery

Can the process resume without duplication or loss?

HANDOFF PIPELINE

The fracture appears where responsibility changes.

Map the transfer and the evidence beside each boundary.



Found a weak handoff? Place the human checkpoint →

COMMERCIAL EFFECT

A silent handoff failure creates delay before it creates evidence.

The cost often appears downstream: missing records, duplicate work, late responses or a customer discovering the break first.

Validated
payload

Usable transfer

Required fields and values are checked before the next step begins.

Format mismatch

Silent data loss

Information arrives incomplete, malformed or in the wrong place.

Completion
confirmed

Visible state

The receiving system records acceptance and usable status.

Completion
assumed

False confidence

A sent event is treated as proof that every later step completed.

Recovery tested

Controlled exception

People know whether to retry, resume, reverse or escalate.

Manual
improvisation

Duplicate or lost work

A rushed fix repeats an action or loses the original context.

Ask where the business would first notice the break.

Handoff failures often have a lag. The earlier the confirmation and alert, the less downstream work must be reconstructed.

FIVE SIGNS THE HANDOFF IS BREAKING

Problems to catch where systems and people meet.

The central step may still appear healthy while the overall process is failing.

Critical

Sign 1

No behaviour is defined for an unhandled case

The happy path is documented but exceptions have no destination.

WHAT IT COSTS

The process stalls or takes an unsafe default when conditions differ.

Major

Sign 2

Connected systems expect different data formats

The sender and receiver disagree about names, values or required fields.

WHAT IT COSTS

Records arrive incomplete, map incorrectly or disappear without a usable error.

Major

Sign 3

No one is notified after partial failure

Monitoring shows activity but does not route a useful alert to an owner.

WHAT IT COSTS

Delay and rework grow before anybody knows intervention is needed.

Moderate

Sign 4

A person assumes the automated step completed

The interface or status gives confidence without confirming the real outcome.

WHAT IT COSTS

Downstream work proceeds with missing, stale or incorrect information.

Moderate

Sign 5

Handoffs are not retested after system changes

Updates on either side alter the boundary while the workflow remains untouched.

WHAT IT COSTS

A once-reliable connection decays as fields, rules or permissions change.

A green step does not prove the next step received usable work.

Test the boundary, the confirmation and the recovery path together.

CORRECTION AND SELF-ASSESSMENT

Five questions before a handoff is called reliable.

Common myth: once built, it just runs. Better framing: handoffs decay as the systems and rules on either side change.

01

What exact payload crosses this boundary?

List required fields, values, formats and destinations.

02

What proves the receiving step accepted it?

Use an observable state, not an assumption based on sending.

03

How does the business learn about failure?

Name the alert, owner, timing and information they receive.

04

Can recovery create a duplicate action?

Test retry, resume and rollback behaviour before real work depends on it.

05

When will this handoff be retested?

Tie review to system, permission, field or process changes.

WORKING HANDOFF STANDARD

A reusable boundary record for every connection.

Document the sender, receiver, expected payload, confirmation and recovery. The boundary should be testable without relying on memory.

01 Name both sides

Identify the sending and receiving system, team or role.

02 Define the payload

List required fields, format, destination and timing.

03 Confirm receipt

Choose the observable state that proves acceptance.

04 Set the timeout

Decide when silence becomes a failure.

05 Route the alert

Name the owner and the context they need.

06 Test recovery

Prevent lost work, unsafe retries and duplicate actions.

INSTEAD OF

“The systems are connected.”

No receipt, timeout, alert or recovery.

Reusable handoff record

SENDER:

RECEIVER:

PAYLOAD:

FORMAT:

CONFIRMATION:

TIMEOUT:

ALERT OWNER:

RECOVERY PATH:

DISCUSS HANDOFF →

Make the boundary observable.

Map one transfer, its evidence and safe recovery.

DUE DILIGENCE

Inspect the boundaries, not only the central automation.

Three rules and two next steps for a recoverable handoff.

01

Validate before sending

Check required data and format before the payload leaves.

02

Confirm after receiving

Record an observable state that the next step accepted usable work.

03

Test failure and retry

Prove alerts, ownership and recovery before the process is unattended.

Reliable automation is a chain of confirmed boundaries.

If a provider cannot demonstrate failure, alert and recovery plainly, keep digging.

CONTINUE LEARNING

Human-in-the-Loop

Place a checkpoint where consequence is highest.

[Read next →](#)

AI Agents & Automation

Continue through the rest of this collection.

[Explore collection →](#)

Design the baton pass, not only the run.

A reliable handoff defines the payload, confirmation, timeout, alert and recovery before connected work depends on it.

PRIMARY NEXT STEP

Human-in-the-Loop

Place one useful checkpoint at the highest-consequence step.

[Continue →](#)

AI Agents & Automation

See every guide in this collection.

[Explore collection →](#)

All guides

Browse every topic and problem.

[View library →](#)

Find the weak boundary.

Bring one automation path. Trace the transfers, evidence, alerts and recovery without changing the whole system at once.

[Discuss the handoff →](#)