# Check the dependencies

Use this worksheet with your website brief before comparing migration quotes. Record what each plugin or custom feature does, what will replace it and what still needs testing.

**Start with a customer journey:** where it starts, what the visitor provides, where the details go and what they should see afterward. List the tools involved, then complete pages 1-2 for each feature. Mark unknowns **To confirm**.

**Keep it safe:** Do not enter passwords, API keys, customer records or private form submissions. Use fictional examples. Save a copy of the PDF after filling it in.

## 1. Feature and business purpose

What does the customer or team need to do?

## 2. Current implementation

Plugin, custom code or external service. Do not include credentials.

## 3. Required behaviour

Trigger, fields, destination, confirmation and exception cases.

## 4. Proposed approach

Webflow feature, existing tool, new tool, custom connection, keep outside Webflow or remove by agreement.

## 5. Compatibility evidence

Official documentation or demonstrated test, date checked, exact action and relevant plan limits.

# Agree the work and ownership

Use the same feature name on both pages. An account being connected is not proof that the required action, fields or subscription plan are supported.

### 6. Content or data to move

Categories and approximate volume. Handle personal-data details separately with the appropriate owner.

### 7. Build work included

Exact deliverable and who supplies configuration and content.

### 8. Recurring cost

Provider, currency, billing interval, usage allowance and overage terms. Unknown is not zero.

### 9. Success check

Observable result and the evidence to retain.

### 10. Failure check and fallback

What happens when the normal route fails?

### 11. Operational owner

Who notices problems, resolves them and maintains the connection?

### 12. Decision

Ready to scope / prototype needed / keep current arrangement / exclude by agreement.

# A quote request reaching the office

This invented example describes a proposed requirement. It is not a Goliath client result, guaranteed integration or included package.

## Purpose and current setup

A homeowner requests a quote for a planned job. The office needs their contact details, work location and a short job summary. The current WordPress form sends an email and creates a request in an external job-management system through a connector.

## Required behaviour

A valid request creates one intake record, keeps the work location separate from the requester's billing address and reaches the named office queue. The visitor sees an accurate receipt message. The form does not reserve a technician or appointment.

## Proposed approach: prototype needed

Rebuild the form and assess whether the existing connector supports the required fields and request type. Keep the feature marked **prototype needed** until the service, account plan and supported action have been checked. Do not substitute "create booking" for "create request" simply because it is easier to connect.

## Questions before quoting

Does the connector support the necessary record type? Which system stores a recoverable copy? What happens if the same event arrives twice? How will an office user see a failed delivery? Does the subscription cover expected usage? Who maintains the connection after launch?

## Acceptance evidence

With separately authorised fictional test data, demonstrate one intended request, its correct property and owner, the visitor's confirmation, and the agreed handling of a duplicate or failed event. Record the tested version and date. A page animation or HTTP success response alone does not prove office receipt.

## Costs and owner

Implementation, subscription and support costs remain **To confirm**. The business appoints the office owner; the proposal names technical support responsibility. No real customer submission is needed to prepare this record.

# Compare the same scope

Keep this worksheet alongside the website brief. Page inventories, URL planning, launch acceptance and handover still need their own agreed scope.

## Four dependencies worth resolving early

**Plugin-generated article content** Check a representative sample for shortcodes, custom fields, tables and embedded media. Render it in the proposed structure and record content differences.

**Project search or filters** Agree essential fields, filters, result counts and mobile interactions. Demonstrate a realistic sample, including no-result behaviour.

**Download library** Inventory the files, choose new locations and check links. Confirm downloads will still work when the old WordPress host stops serving them.

**Member area or online store** Assess customer, payment, subscription, access and fulfilment rules separately. Do not assume a general marketing-site migration covers these workflows.

---

## Choose a decision for each feature

**Ready to scope:** The behaviour, approach, test, cost assumptions and owner are clear enough to quote.

**Prototype needed:** A business-critical capability remains uncertain. Agree a bounded test before promising the full migration.

**Keep the current arrangement:** The move would create an unacceptable gap or cost. Keep a feature in its existing service, or stay on WordPress for now.

**Exclude by agreement:** The business no longer needs the feature. Record content, customer and subscription consequences before removing it.

When comparing quotes, check the same behaviours and ongoing responsibilities. "Forms included" is too broad if only one quote covers routing, failure handling and a named support owner. This is a scope aid, not a promise of rankings, leads or savings.

**Further reading**

Goliath: WordPress to Webflow migration

Webflow: Migrate your site from WordPress to Webflow

Google: Site moves with URL changes