Mary Eltresse
WEB DEVELOPER & OPTIMIZATION SPECIALIST

FREE RESOURCE FOR DIGITAL AGENCIES

The WordPress Agency Technical Rescue Kit

A practical toolkit for identifying, prioritizing, and handing off technical WordPress work across client websites.

Use these five resources in your own project workflow. Print the kit, save it as a PDF, or copy the worksheet prompts into your ticket system. Keep sensitive client information and credentials out of shared copies.

RESOURCE 1 OF 5

WordPress Troubleshooting Triage Checklist

Capture the evidence before making changes.

- [ ] Record the affected URL, expected behavior, actual behavior, device, browser, time and steps to reproduce. Separate confirmed observations from assumptions.
- [ ] Reproduce the issue and note whether it affects logged-in users, logged-out visitors or a specific role. Preserve screenshots or a short recording where useful.
- [ ] Check the browser console and network requests for relevant errors. Record the error and affected resource without including tokens or personal data.
- [ ] Review available PHP errors and WordPress debug logs in an appropriate environment. Keep logs private and avoid displaying debugging output on a public site.
- [ ] Record recent plugin, theme, WordPress, PHP, hosting or configuration changes. Note the last known working state.
- [ ] Investigate plugin and theme conflicts on staging or another agreed safe environment. Do not disable business-critical functionality on production as an unplanned test.
- [ ] Decide whether staging is required before making changes. Confirm backup, rollback and approval arrangements.
- [ ] Summarize what you know, what remains unknown and the next investigation step. Escalate payment, security or data-loss concerns promptly to the responsible person.

Prompt	Project notes
Issue / affected URL	
Steps to reproduce	
Evidence and recent changes	

Prompt	Project notes
Safe test environment	
Next step / owner	

RESOURCE 2 OF 5

Custom Development Requirements Worksheet

Define the outcome before choosing the implementation.

Prompt	Project notes
Business objective	What task must a visitor, client or administrator be able to complete?
Existing WordPress setup	Theme, editor/builder, WordPress and PHP versions, hosting constraints, relevant plugins.
Required functionality	Describe inputs, behavior, permissions, outputs and error states.
ACF requirements	Field groups, field types, validation, relationships and who maintains the content.
Custom post types	Content entities, taxonomies, archive/single templates and existing content migration.
Search / filtering	Searchable fields, filter combinations, sorting, pagination and empty states.
Dependencies / integrations	Plugin licences, external systems, API documentation, data ownership and limits.
Acceptance criteria	Concrete scenarios the reviewer will use to confirm the work is complete.
Constraints / exclusions	What is out of scope? Note accessibility, performance and maintenance requirements.
Approver / delivery needs	Named reviewer, desired date and any dependencies on other teams.

RESOURCE 3 OF 5

Website Pre-Launch Technical QA Checklist

Agree which checks apply to the project and record anything not tested.

- [ ] Responsive layouts: review key pages on real mobile devices and desktop. Check navigation, text, images, overlays and horizontal overflow.
- [ ] Forms and email delivery: test required fields, validation, confirmation messages and delivery to the intended inbox with permitted test data.
- [ ] WooCommerce: test product selection, cart, coupon conditions, shipping and checkout using a safe testing environment and approved payment test methods.
- [ ] Broken links: follow primary calls to action, navigation, important internal links and download links. Check custom error pages.
- [ ] Performance: review representative landing, product and content pages using consistent test conditions. Record findings rather than relying on a single score.
- [ ] Caching: check that dynamic content, login, cart, checkout and user-specific views behave correctly for the installed caching setup.
- [ ] Security and backups: confirm access permissions, the agreed update state, current backups and a practical restore path. Record any specialist review still required.
- [ ] Launch review: confirm the correct environment, production URLs, indexing intentions, analytics requirements and stakeholder approval.
- [ ] After deployment: repeat critical smoke tests and confirm that forms, shopping journeys and key content still work.

Prompt	Project notes
Pages and journeys tested	
Devices / browsers / date	

Prompt	Project notes
Known issues / not tested	
Release approver	

RESOURCE 4 OF 5

Development Priority Planner

Prioritize impact and dependencies, not just whoever asks most loudly.

Prompt	Project notes
Issue description	What is broken or missing? Add evidence or a ticket reference.
Business impact	Who is affected, and what task or business process is blocked?
Urgency	Immediate / planned / backlog. Explain the deadline or reason.
Estimated complexity	Small / medium / large / unknown. Mark estimates as provisional.
Assigned developer	Owner and reviewer.
Deadline	Requested date, confirmed date and dependencies.
Status	New / investigating / scoped / in progress / review / complete / blocked.

Working issue list

Issue / impact	Priority / complexity	Owner / deadline	Status

Issue / impact	Priority / complexity	Owner / deadline	Status

Confirm unknowns before committing to a fix estimate. Review priorities when new evidence changes the impact or scope.

RESOURCE 5 OF 5

Developer Handoff Checklist

Make it easier for another developer to start without guessing.

- [ ] Access requirements: list the accounts and permissions needed, who grants them and when they expire. Share credentials through an approved secure channel, never this worksheet.
- [ ] Staging environment: identify the URL, deployment relationship, environment differences and any restrictions on production data.
- [ ] Backup confirmation: record the latest agreed backup, who can restore it and where the restore procedure is documented.
- [ ] Technical documentation: provide repository details, branch workflow, setup notes, dependencies, licences and relevant existing decisions.
- [ ] Deployment procedure: agree release steps, production changes, cache handling, rollback triggers and who can approve deployment.
- [ ] Testing and approval: list acceptance criteria, test journeys, reviewer, revision scope and sign-off process.
- [ ] Completion: document changes, outstanding issues and maintenance considerations, then revoke temporary access where appropriate.

Prompt	Project notes
Project / agency / contact	
Scope and exclusions agreed	
Staging / backup owner	
Repository and documentation references	

Prompt	Project notes
Deployment and rollback approver	
Review / sign-off date	

When WordPress gets complicated, I step in.

Need help turning the findings into working code? I offer custom WordPress development, troubleshooting, performance work and agency overflow support at $25 USD/hour, with fixed-price projects available.

Discuss your WordPress project or visit Mary's contact form. Scope, availability and access arrangements are confirmed before starting.

Mary Eltresse · maryeltresse.com