Code review checklist

Checklist

- User Interface
  - Do we have screenshots of all UI changes?
  - Does this change introduce any new concepts / models? Evaluate.
  - How much expertise and context will a prospective user need?
  - Do we have anything similar elsewhere? Can we build on existing knowledge and habits?
  - Do we do anything similar but just different enough to cause problems with old habits?
  - Are there any new multi-step workflows? How long and complex are they?
  - If users make mistakes or errors, what are the failure conditions? Can mistakes be undone?
  - How is the copy? Are sentences well-formed and clear? Any spelling errors or typos?
  - Is the UI layout localizable?
- Visual Design
  - Are controls laid out in way which makes sense given visual scan order?
  - Check for alignment
  - Check for layout, spacing, padding
  - Check for visual consistency
  - Does the UI match existing design patterns?
- API
  - Does it follow existing API patterns?
  - Does it include links where relevant?
  - Does it add any new response codes? Is this necessary? (existing: 200, 404, 400, 405, 500, 401)
  - Are API calls documented?
  - Are there behat tests?
  - Is it checking for correct oauth scope / user permissions?
- Coding style
  - Does it meet coding style requirements? for PHP, HTML, JS, CSS
  - Is code readable?
  - Are variables, functions, classes, etc well-named?
  - Is it obvious what the code does? If not, is it sufficiently commented?
  - Could code be made easier to understand?
  - Is the code extensible?
- Correctness
  - Correctness of algorithms
  - Loop iteration and off by one
  - Incorrect behaviour
  - Are errors well handled?
  - Relying on external resources? Do we fail gracefully if they're missing/incorrect?
  - Switch / break fall through
  - Exception safety - Code has reasonable behavior when an exception is thrown. ie. resources should not be leaked, error is reported to the user, doesn't leave junk data behind and continue in a usable state
- Security
  - Are user inputs sanitized and/or validated?
  - Are DB queries parameterized or using the query builder?
  - Are files created and read/written? Check permissions, races (ie TOCTTOU)
  - Validate remote host cert validity/identity
  - Crypto use? Does it follow best practice? Does it use trusted libraries?
  - Are we checking for scope & user permissions?
  - Is output escaped? Is it using the correct escape function? (ie. not using HTML escaping in JS or other mismatch?)
  - Does this open CSRF vectors? (ie. create/update/delete actions with direct URL?)
  - Adding a new library? have we reviewed that library for sec issues?
  - Check OWASP top 10
- Licensing
  - Are we adding a library? Does it use a compatible license?
  - Non staff contribution? Do we have contributor agreement for the author?
- Documentation
  - Have docs been added to the wiki?
  - Does it have inline PHP docs?