[Project Title]

# Evaluation Process: What to look for and red flags

[Partner Agency]

[Date]

18F

# Background

[Agency] is about to embark on an important procurement to find a vendor team to assist them in [scenario - i.e. building open source software that will replace their legacy system].

The evaluation team will be reviewing vendors' proposals and identifying the advantages and disadvantages in order to identify the vendors that should continue to the next phase of the procurement process, vendor interviews. They will rank the the vendors technically, based on their advantages and disadvantages in both written and verbal responses, in order to rank the vendors that have the best technical approach.

More information on the evaluation plan can be found in the [agency's RFQ].

# Purpose of this guide

This document is intended to support the evaluation team and help them identify indicators of a strong vendor proposal. We've included the three technical evaluation criteria from the RFP and both red and green flags that you might come across while reading proposals. **Note**: This is not intended to be an exhaustive list — every procurement is different.

# Evaluation Factors

## Technical Approach

> [include agency-specific language, for example]
> In evaluating a contractor's technical approach, the agency will consider (a) the quality of the contractor's plans to provide the open source, agile development services required, including user research and design, (b) the extent of the contractor's understanding of the details of the project requirements, and (c) the extent to which the contractor has identified potential obstacles to efficient development, and has proposed realistic approaches to overcome those potential obstacles.

What to look for:

- Competency. They should propose using the right tools in the right way and can defend their recommendations.
- A lack of novelty. The best approaches will recommend time-tested software and infrastructure, employing design patterns that are known to work.
- A lack of certainty. Maybe the vendor's idea is a good one, maybe it's a bad one — they can't really know yet, and they need to know that. This should include highlighting weak points or areas of uncertainty in their technical approach.
- A vision. The vendor needs to *see* this, in a way that can act as a catalyst to the [agency's] vision.
- Understands program goals. The vendor should have a clear grasp on what the [agency] is doing. There should be no serious misunderstandings of information that was described clearly in the RFQ.
- Experience developing open source software.
- Collaboration and communication. They expect the agency's product owner to be a valuable, active member of the team. They also expect to communicate proactively about risks and roadblocks, so that they can work as effectively as possible and their actions will not be a surprise.
- Regular and ongoing user research to understand users' goals and needs, and what to build that supports them; combined with usability testing to ensure that users can achieve their broader goals in using the software, and that it addresses their needs along the way. They plan to conduct user research, and test everything from rough prototypes to more polished software with actual users, throughout the entire design and development process.
- They follow a human-centered design process, and are able to explain how they make design decisions in relation to broader user goals and specific needs learned through their research.
- They focus on automation, reliability, testability, configuration-as-code, and/or other core DevOps principles. The quote includes references to modern automation and deployment tooling like Jenkins, Puppet, Chef, Travis CI, CircleCI, Kubernetes, Terraform, AWS and Heroku.

Red flags:

- Don't seem to understand program goals. There is a serious misunderstanding of information that was described clearly in the solicitation.
- Misidentifying the name of technologies in a way that indicates a lack of experience communicating about them (e.g. "we'll index records with an

Elasticsearch," instead of "with Elasticsearch," or "we recommend using JAVA," instead of "Java").

- Excessive complexity.
- Shirking page-limit rules (with tiny fonts, reduced leading, etc.) because they believe their technical approach to be so brilliant that it can't possibly fit within the prescribed limit.
- Basing their solution on fundamental misunderstandings of the [agency's] needs that they should have understood.
- Proposing the use of arcane platforms and technologies, especially when those arcane platforms and technologies are house specialties of the vendor.
- They never once mention accessibility of the software to the 1 in 5 Americans with a disability or do not identify how they will evaluate whether their software meets accessibility standards
- They don't consider, explicitly or implicitly, that user research will ultimately determine what is built, which in turn will dictate the technical approach.
- Uses terminology like "requirements will be collected from the business owner."
- They're proposing to outsource what should be core competencies, e.g. DevOps, or Javascript.
- They propose a process that includes working for long stretches of time without interacting with the [agency] and/or users of what's being built.
- They propose relying on focus groups, instead of structured, one-on-one usability testing sessions.
- They're proposing to outsource what should be core competencies, e.g. DevOps, user research or usability testing, or UX design.
- They describe the goal of research as being to "test the app with users", "find problems', or ask users what they "like," "want," or "might do" (shows that they are drawing conclusions based on what users *say* instead of observing and learning from users what they *do).*
- They use the term "user testing" instead of "usability testing" (not testing the user, testing the functionality of the system).
- They propose relying on focus groups, instead of structured, one-on-one research interviews or usability testing sessions.
- They prioritize aesthetics over usability and usefulness, and cannot explain why they made design decisions.
- Doesn't demonstrate that testing is important.
- They propose long-term staff augmentation.

## Staffing Approach

> [include agency-specific language, for example]
> In evaluating a contractor's staffing approach, the agency will consider (a) the skills and experience of the Key Personnel and other individuals that the contractor plans to use to provide the required services, (b) the mix of labor categories that will comprise the contractor's proposed development team, and (c) the contractor's proposed number of hours of services to be provided by each member of the contractor's proposed development team.

What to look for:

- A small number of team members, each providing a clear value. There is a purpose for everybody proposed.
- Familiarity with and use of modern software languages (e.g. Python, Ruby, PHP, C#, Javascript )
- Familiarity and use of web-based application programming interfaces (APIs), especially REST and GraphQL.
- Experience using Git for software version control.
- The lead developer's skill mix and experience covers a substantial portion of the likely work needed to be done on the [agency's] project.
- If the developers have presences on social coding platforms (e.g., GitHub), how does their work look? What expertises are evident there? Do they have expertises that don't appear on their resume, but their work shows that they possess?
- Staff resumes support their claimed expertise.  (For example, does the Content Designer have any actual content design experience, or are they a project manager in sheep's clothing?)
- The lead developer's skill mix and experience covers a substantial portion of the likely work needed to be done on the agency's project.
- The lead user researcher's background indicates an understanding of how research can inform and shape strategy, design, and development; familiarity with a variety of research and testing methods; and experience in deciding which method or methods to use based on the learning goals of the phase or needs of the project, and with recruiting users based on those goals and needs.
- The lead UX designer's background demonstrates strong craft skills and experience in generating concepts based in overall strategy, user research, and user-centered design best practices; and in communicating those concepts visually via a variety of methods including but not limited to sketching,

wireframing, prototypes, and more polished mockups — for use in research and to guide development.
- The team is generally assigned to the project full-time and will not be splitting their time across other unrelated projects. There may be acceptable exceptions, such as for a “scrum master,” but in general everyone should be fully staffed to the project. This is critical for project leads, developers, user researchers, designers, and all key personnel.

Red flags:

- Overstaffing the bid. A team that consists of people with fancy titles, or far more experience than is necessary, means that the vendor either doesn’t understand this way of working or is trying to squeeze every dime they can out of the engagement.
- Proposing positions that do not have a place in iterative development – project managers, business analysts, enterprise architects, delivery managers, etc.
- Poorly-designed websites for the company, proposed subcontractor, or proposed staff resumes.
- Proposing antiquated software technologies that don’t have an active developer community (e.g. Cold Fusion, ASP, FoxPro)
- Lack of experience with test automation, aka DevOps, aka test-driven development (TDD).
- The lead developer isn’t impressive.
- The team has no apparent experience with usability research.
- The team has no apparent experience with visual design.
- The flashiest member of the team is proposed to spend a tiny amount of time on the project.
- Key skills don’t appear in any resumes, such as:
    - Migrating from one platform to another
    - Agile development practices
    - Automated (unit/integration/end-to-end) testing
    - Continuous Integration and Continuous Deployment
    - DevOps
    - Refactoring to minimize technical debt
    - Application Protocol Interface (API) development and documentation
    - Open-source software development
    - Cloud deployment
    - Product management and strategy
    - Usability research, such as (but not limited to) contextual inquiry, stakeholder interviews, and usability testing

    - User experience design
    - Sketching, wireframing, and/or prototyping, and user-task flow development
    - Visual design
    - Content design and copywriting
    - Building and testing public facing sites and tools
- No actual technical staff, but has “access to a database of resumes.”
- Proposed staff don’t currently work for the vendor, and there is no letter of intent from the proposed staff.
- Proposed staff resumes are copied from the internet, in large part or, more rarely, in whole.
- Key staff are not proposed to be 100% full time to the project, or the project is staffed with a number of partial FTE personnel or has no full time staff.

## Similar Experience

> [include agency-specific language, for example]
> In evaluating a contractor’s similar experience, the agency will consider the extent to which the contractor has recently provided software development services for projects that are similar in size, scope, and complexity to the project described in this RFQ, and the quality of those services. In evaluating the quality of those services, the agency will consider, among other things, the revision history for all files in the source code samples provided. The agency will also consider the user research and design-related artifacts that were associated with the source code samples provided or submitted separately. In considering a contractor’s similar experience, the agency may also consider information from any other source, including contractor’s prior customers, and public websites.

**Caveat**: Don’t spend time reviewing their Git repositories if you aren’t familiar with Git. This is something that [18F technical lead] will do and provide their thoughts and recommendations back to the team as technical advisors. [18F technical lead] will be looking for things like:

- Proper use of Git, commit changes with personal accounts (not organizational), use of a branching / merging strategy, informative comments, evidence of code reviews, and use of a CI/CD pipeline.

- Code that conforms well to our RFQ's QASP.
- Git collaboration. Work was performed in a reasonable number of GitHub comments.
- Substantial projects. The projects were not created just to have something to point to for this RFQ.
- How they incorporate user feedback into their development process.
- That their tests are written well, and cover the supermajority of the code.
- Consistent, enforced code style.

What you should look for:

- Work that is conceptually similar to the [agency's] needs, i.e. a system in which a record is submitted, and then moves along an assembly line with different controls and modifications to the record along the way.
- Work that was centered on user needs, as opposed to leading with solutionism.
- Work that was completed by a team of a size that's similar to the size of the team that they're proposing.
- Design artifacts that show continuous and ongoing usability testing that show a user-centered approach to iterative design and development.

Red flags:

- The cited projects bear little evidence of having been created by the vendor.
- The projects are trivial.
- There's a finished product, but no code, or vice-versa.
- The projects do not include good design artifacts and research plans.