



**RFP Template**

**SUMMARY**

[1 Introduction 3](#_Toc26998093)

[1.1 What is an Agile RFP? 3](#_Toc26998094)

[2 Product manger nomintation 3](#_Toc26998095)

[3 Vendor Capability 3](#_Toc26998096)

[4 Executive summary of the project and company overview 5](#_Toc26998097)

[4.1 Executive summary 5](#_Toc26998098)

[4.2 Company Overview 6](#_Toc26998099)

[4.3 Acquisition Context 6](#_Toc26998100)

[5 Agile RFP 8](#_Toc26998101)

[5.1 RFP Changes 8](#_Toc26998102)

[5.1.1 Government’s requirements and expectations of the contractor's 8](#_Toc26998103)

[5.1.2 Instructions for providing the proposal 16](#_Toc26998104)

[5.1.3 Decision for the award 17](#_Toc26998105)

[5.1.4 CDRLs - Contract Data Requirements List 18](#_Toc26998106)

[5.1.5 Cost Proposal and Schedule Instructions 20](#_Toc26998107)

[6 Project Goals 22](#_Toc26998108)

[7 Context 22](#_Toc26998109)

[8 Scope of services and deliverables 22](#_Toc26998110)

[8.1 Scope 22](#_Toc26998111)

[8.2 Stories 23](#_Toc26998112)

[8.3 USER EPIC 23](#_Toc26998113)

[8.4 Benchmark estimates 23](#_Toc26998114)

[8.5 Project Management 24](#_Toc26998115)

[8.6 Infrastructure – IT Environment 24](#_Toc26998116)

[8.7 Technology preferences 24](#_Toc26998117)

[8.8 Functional Design 24](#_Toc26998118)

[8.9 Product Requirements 25](#_Toc26998119)

[8.10 Development 25](#_Toc26998120)

[8.11 QA 25](#_Toc26998121)

[8.12 Product Management 25](#_Toc26998122)

[8.13 Definition-of-Done (DoD) 25](#_Toc26998123)

[9 Project Timeline 26](#_Toc26998124)

[10 KEY PERSONEL 27](#_Toc26998125)

[11 Suggested structure for vendor bids 28](#_Toc26998126)

[12 Selection criteria 28](#_Toc26998127)

[13 Project Costs 28](#_Toc26998128)

[14 Pricing 29](#_Toc26998129)

[15 ANNEXURES 30](#_Toc26998130)

[15.1 Annexure I - Existing Roadblocks Or Technical Issues 30](#_Toc26998131)

# Introduction

## What is an Agile RFP?

Agile RFP is a practical way of value-based competition between invited or pre-selected software vendors. Rather than asking and replying to questions and than determining which answer and proposal are better, you try it, experience the vendors and see the real results.

# Product manAger nomination

The product manager /Product team, who will be accessible for the bidders during the project *(someone on the business side from the existing Agile project being purchased). It can be also a small group of, e.g. 3 people to cover all business aspects, knowledge and needs, but also for the case of substitutability.*

|  |  |
| --- | --- |
|  | Name |
| The product manager |  |
|  |  |
|  |  |

Prospective Bidders requiring any further information or clarification of the Bidding Documents may notify the Client in writing at the Client's mailing address indicated below:

|  |  |
| --- | --- |
| Bid Clarification | Technical Clarification |
|  |  |

# Vendor Capability

The Invitation to bid is open to all suppliers and authorized dealers who fulfill the following:

|  |  |
| --- | --- |
| Requirement | Bidder comment |
| Have the same/similar environment (internal or external) |  |
| Have the same (claimed) team size and wanted to know the changes for the sprints during the sprint planning |  |
| We have prepared benchmark stories (with an expert estimate from our side) mandatory to all vendors |  |
| Velocity achieved with claimed team in RFP was expected as the initial one in the subsequent project. |  |
| Ownership of the delivered code at the end. |  |
| Definition of Done (e.g.: conducted all demo sessions, showing the sprint plans, having code in GIT for the demo, …) |  |
| Set the size of allowed vendors’ development team (keep it small, something like 3 to 5 people including analysts and testers) to have comparable results and deliverables.  Ask for openness if vendors use other people to help, e.g. UX people, other analysts, architect helping with design. |  |
|  |  |

# Executive summary of the project and company overview

## Executive summary

MY COMPANY (MC) is running a Request for Proposal (RFP) process to identify a partner to plan, execute and deploy a complete agile development project for MC. MC is looking for a partner that can demonstrate expertise in working with companies like ours and delivering successful site projects.

Agile methods have proven effective in rapidly and responsively delivering functionality in commercial environments in which products are developed based on internally identified needs and offered for sale to multiple uncommitted customers. Agile approaches are becoming more and more attractive to businesses in light of recent budget pressures and a widespread need to bring capabilities to users more quickly than in the past.

Constantly evolving technology systems have presented a demand for an acquisition process that is able to respond quickly to emerging requirements and rapidly changing environments. To address this, our company has encouraged the following characteristics in acquisitions:

1. **Flexibility**: tailoring program strategies and oversight
2. **Responsiveness**: rapid integration of advanced technologies
3. **Innovation**: adopt practices that reduce cost and cycle time
4. **Discipline**: use of program baseline parameters as control objectives
5. **Effective Management**: decentralization to the extent practicable

These characteristics have led to an increased focus on flexible development approaches that include Agile philosophies and integrated program management tools such as Earned Value Management.

This approach to procurement was thought in Agile methods since by design these agile methods comprise technical and programmatic practices employed to reduce wasted effort, decrease bureaucratic overburden, and enhance the productivity and engagement of developer teams.

## Company Overview

*Provide your company background. We recommend that you make sure to give out the following information:*

* Company size
* Geographic scope
* Mission and vision
* Brand perimeters/restrictions
* Company culture

*This is a great space to give a thorough overview of your company, which will significantly help suppliers answer to your company needs. Don’t skimp the details here.*

# Agile RFP

## RFP Changes

Traditional RFPs are generally based on the government expecting a traditional waterfall-structured program approach for creating the system in question. The traditional government procurement process and associated guidance do not necessarily facilitate employing Agile methods as part of the solution. This section explores options that would either directly request Agile responses as part of the solution.

### Government’s requirements and expectations of the contractor's

this section provides the buyer’s requirements and expectations of the contractor's performance in the form of a SOO or SOW as part of the RFP. The SOO reflects a performance-based acquisition (PBA) and is best suited for an Agile acquisition. If a SOO is provided, the government will normally expect the contractor to provide a SOW or a performance work statement (PWS) as part of its proposal. The buyer-provided SOW is best suited for a traditional acquisition in which the buyer has a high degree of confidence in the ability to specify (both qualitatively and quantitatively) the expected approach and product end state.

Table 1 highlights the differences.

*Table 1: SOO versus SOW*

|  |  |  |
| --- | --- | --- |
| SOO | Factor | SOW |
| The buyer understands the objectives but expects the end state to evolve. | buyer **Understanding** | The buyer has a high level of confidence in the end state. |
| Change is expected to be a significant factor in achieving the end state. | **Change** | Change is not anticipated, or if encountered will not be disruptive. |
| This approach provides the offerors trade space and flexibility in developing their proposals. It should probably be used unless the totality of the work effort required is very well understood by the buyer. | **Constraint** | Constrains offerors to the specific tasks identified, so it must be unambiguous and comprehensive. The buyer needs to apply specific constraints on the tradeoff space of lifecycle cost, performance, interoperability, logistics/training, etc. |
| This approach conforms to PBA practices. | **Performance-Based Acquisitions (PBA)** | NA |

#### The Statement of Objectives (SOO)

The SOO is used in PBA “which structures the acquisition around the purpose and outcome desired as opposed to the process by which the work is to be performed.” With the performance-based acquisition, the buyer no longer develops a prescriptive SOW directing how the contractor will achieve project milestones. Instead, the buyer develops a SOO or PWS that describes the overall outputs and objectives but does not specify how to achieve those outputs.

The SOO is “incorporated into the solicitation that states the overall performance objectives. It is used in solicitations when the buyer intends to provide the maximum flexibility to each offeror to propose an innovative approach.” It focuses on the buyer’s objectives and identifies the broad, basic, top-level objectives of an acquisition/procurement. The SOO is developed after performing a risk assessment that highlights the high and moderate risks for business, programmatic, and technical areas against the requirement document. It is best applied when the buyer can describe the objectives but is not ready to specify the end state and expects change as a factor in achieving the end state. It provides offerors the flexibility to develop cost-effective solutions and propose innovative alternatives to meet the objectives.

The SOO provides enough information and detail for the offeror to structure a sound program, designed to be executable and satisfy buyer objectives. The offeror uses the SOO to develop the contractor statement of work (CSOW), the contract(or) work breakdown structure (CWBS), the integrated master schedule (IMS), and other documents supporting and defining the proposed effort.

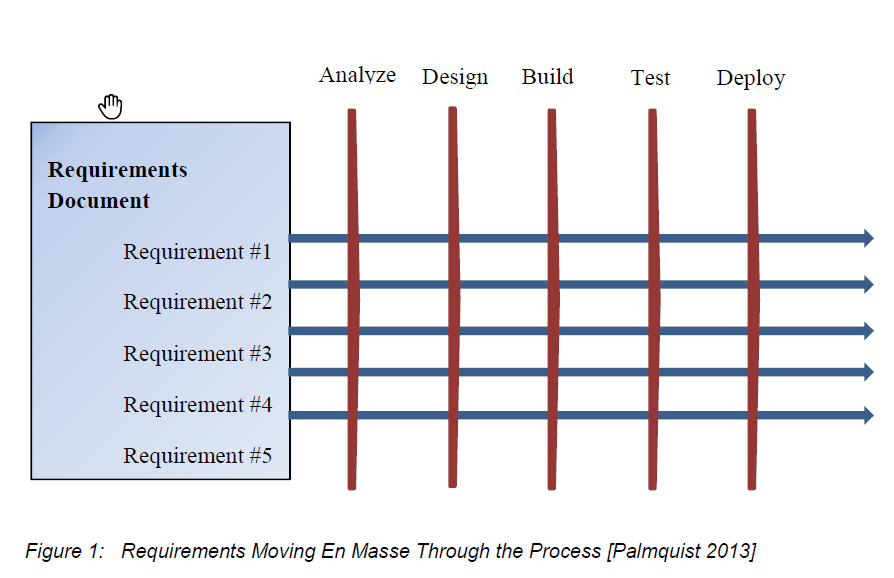
The contractor will sell off the buyer objectives by meeting the predefined and negotiated definition of done when using Agile methods. Once the definition of done is met, the buyer objectives can be said to have been met.

#### Statement of Work (SOW)

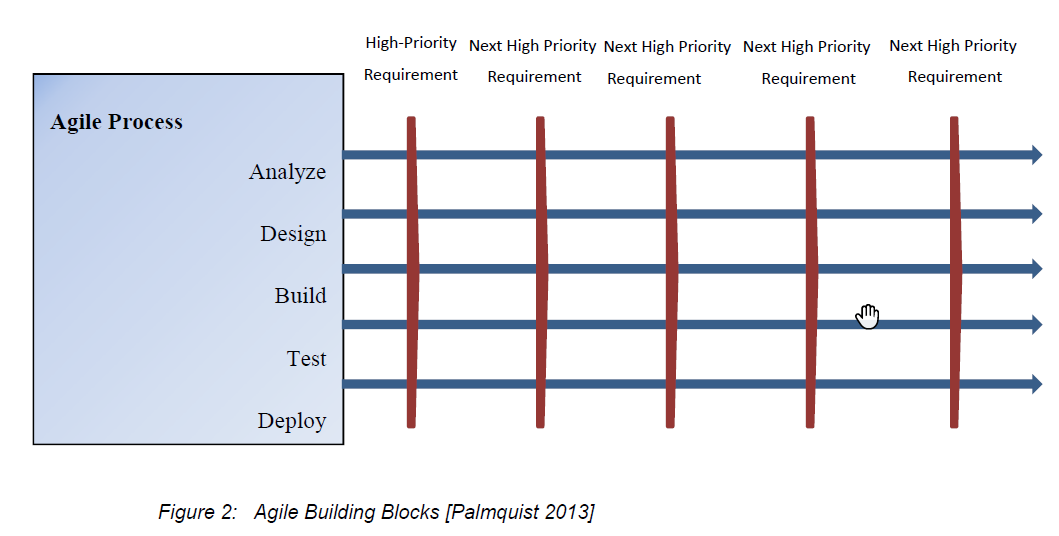
The SOW describes the buyer’s needs for the work to be done in developing or producing goods or services. The SOW is “a portion of a contract which establishes and defines all non-specification requirements for contractor's efforts either directly or with the use of specific cited documents.” It also facilitates the preparation of a proposal and aids the buyer in conduct of the source selection and contract administration after award.

The preparation of an effective SOW requires both an understanding of the goods or services that are needed to satisfy a particular requirement and an ability to define what is required in specific, performance-based, quantitative terms. This requires the buyer to have a high level of confidence that it understands the end state and doesn’t expect change to be disruptive. The SOW should reference qualitative and quantitative design and performance.

#### Reviews in an Agile Environment

In a traditional waterfall model, technical reviews are used as control gates to move from one sequential phase to the next (e.g., concept > requirements > design > develop > test and integration > deployment). Figure 1 shows the traditional progression of requirements through the waterfall model. A traditional program tends to be risk-averse and uses reviews for risk mitigation.

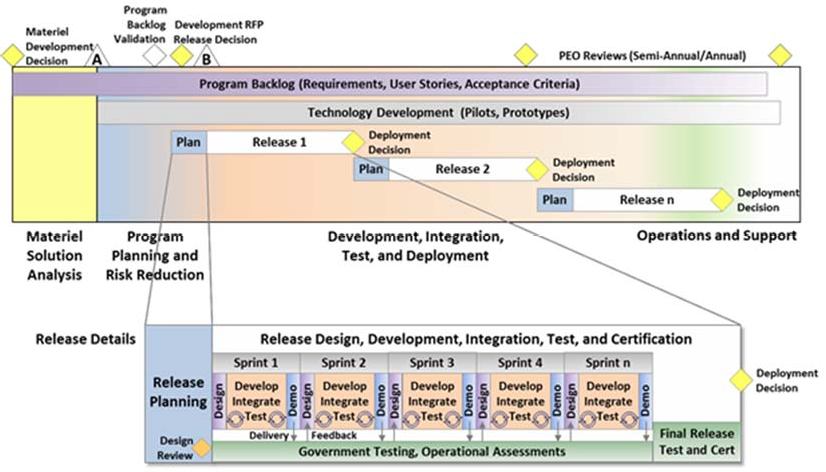
In an Agile program, the focus is on completing each work unit from analysis to deployment instead of completing each phase in sequential order. For example, as shown in Figure 2, the Agile world uses the same building blocks but looks at them differently than the traditional world [Palmquist 2013]. An Agile program tends to use technical reviews as opportunities for information sharing, face-to-face coordination, and confidence-building.



For this reason, Agile programs may treat some reviews (most frequently design and test readiness) incrementally as part of the design-develop-integrate-test (DDIT) cadence, which allows for a faster start (from contract start to first incremental design review) and more opportunities to address changes in technology and the mission environment. In general, the larger part of the program lifecycle that is Agile, the greater the benefit, but Agile methods can provide value even when applied only to software development. The decision about which reviews are conducted sequentially and which incrementally should be based on program needs and challenges. Keep in mind there is no one correct way to implement Agile concepts within a program. Having provided this caution, we are providing in the following paragraphs two examples of how a program may implement Agile concepts within the milestones typically seen in a government program.

A generic Agile program development pattern is shown in 0Figure 3.

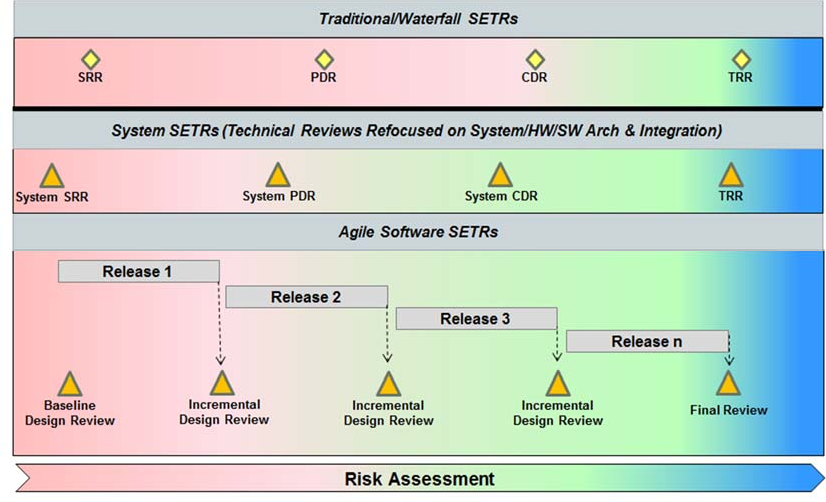
In this example, the pattern is based on a structure where the period of performance contains a series of time-boxed releases (1- n). Each release (as shown in the expanded portion) begins with release planning, followed by a series of DDIT iterations (e.g., sprints 1-n) and concludes with a system demo followed by a deployment decision. Notice that a design review is shown at release planning and a deployment decision is shown at the end of the release. These reviews can be used as tailored alternatives to satisfy the preliminary design review (PDR) and critical design review (CDR).



**Figure 3: Agile Development Pattern (Example 1)**

Note that the graphic implies deployment at the end of a release. A more sophisticated Agile model separates deployment from the release time box. In this model, the software is still developed on cadence (the release time box) but released on demand. Rather than push new software to the system, new software is always available for the system to pull at any time.

Our second example, Figure 4, depicts the contrast between typical milestones and reviews in a traditional waterfall environment versus incremental reviews in an Agile model. In this model, the traditional System Requirements Review (SRR), PDR, and CDR events are replaced by incremental design reviews and, if needed, system-level reviews. The system-level reviews are smaller in scope than the corresponding events in the traditional waterfall programs, focusing on the system architecture, hardware integration, system-level risks, and other system-level concerns.



*Figure 4: Another View of Reviews in Agile Environment*

#### Key Technical Reviews

Technical reviews should reflect the Agile principles, the Agile principles as defined in the Marbach paper or some combination thereof, depending on the environment and needs of the individual program.

“Business personnel, customers or their advocates, and implementers must work together daily” and “Continuous attention to technical excellence and good design enhances agility”—are the most applicable in this area.

To summarize reviews must add value to the program—not just be held because status quo demands them; keep them to a minimum while keeping them small, quick and agile to minimize potential changes in the technical and operational environments. In addition, apply the heuristic of “Minimize the distance between decision and action.

Key technical reviews are listed below.

**System Requirements Review (SRR)**

***When Conducted****.* At the beginning of the period of performance and any time at which there is a major change in the capabilities baseline.

***Purpose****.* The System Requirements Review (SRR) is a multi-disciplined technical review to ensure that the developer is ready to proceed with the initial system design.The review will include the contractor’s understanding of the program’s objectives, capabilities and non-functional requirements baseline, constraints such as architecture, time-critical deliveries, and milestones.

***Input****.* Capabilities baseline, program vision, and roadmap.

***Output****.* Initial program backlog.

**Preliminary Design Review (PDR)**

***When Conducted****.* Post SRR/system functional review (SFR) (in traditional/waterfall); scheduled when appropriate to verify architectural consistency between the hardware and software, technical performance measures (TPMs), and the like (in an Agile development program, only if needed).

***Purpose****.* In a traditional waterfall program, PDR demonstrates that the preliminary design meets all system requirements with acceptable risk and within the cost and schedule constraints. It establishes the basis for proceeding with detailed design. In Agile, the large scale and formality of a PDR are replaced with more frequent, less formal interactions between the customer and the contractor. Examples include release planning and release demos. As such, a traditional PDR is not recommended on purely Agile programs. In some hybrid programs, a PDR may be desirable. For example, in a program using Agile software development and a waterfall approach to hardware development, a system-level PDR might be used to assess overall program progress. In that case, the focus of the PDR event should be verifying architectural consistency between the hardware and software, TPMs, and hardware components

***Input****.* Artifacts demonstrating that the preliminary design meets all system requirements with acceptable risk within cost and schedule constraints.

***Output****.* The decision to proceed with detailed development (in traditional/waterfall); insight into current program status regarding architectural consistency between the hardware and software, TPM maturity, etc. (in an Agile development program).

**Critical Design Review (CDR)**

***When Conducted****.* Post PDR (in traditional/waterfall); scheduled when appropriate to verify architectural consistency between the hardware and software, TPMs, etc. (in an Agile development program, only if needed).

***Purpose****.* In a traditional waterfall program, CDR demonstrates that the maturity of the design is appropriate to support proceeding with full-scale fabrication, assembly, integration, and test. CDR determines that the technical effort is on track to complete the flight and ground system development and mission operations, meeting mission performance requirements within the identified cost and schedule constraints. As is the case with PDR, a traditional CDR is not recommended on purely Agile programs for the same reason; however, again, on some hybrid programs, a CDR may be desirable and would be focused on the same type of topics as an “Agile PDR,” but the expected level of maturity (of TPMs for example) would be greater.

***Input****.* Artifacts demonstrating that the maturity of the design is appropriate to support proceeding with full-scale fabrication, assembly, integration, and test, within cost and schedule constraints.

***Output****.* Decision to proceed with fabrication, assembly, integration, and test (in traditional/waterfall); insight into current program status regarding architectural consistency between the hardware and software, TPM maturity, etc. (in an Agile development program).

**Release Planning and Review**

*When Conducted.* At the beginning of the release (nominally a week in duration).

***Purpose****.* Demonstrates confidence (sufficient understanding of the requirements and design approach) for moving into the DDIT activities that produce the release code baseline. The release planning combines elements of a traditional PDR and CDR, with enough confidence in maturity to move into the next increment; the remaining level of maturity is achieved during the DDIT phase. Release planning meeting and design reviews are iterative—the first release is typically different from subsequent ones in that there is a greater focus (sometimes an exclusive focus) on architectural concepts and enabling elements. Typically, the first release also does not involve adjudicating high-priority defects against features, although it may require effort to refactor inherited system artifacts from legacy programs. The design items presented at release planning pertain to the work (capabilities and architecture) allocated to that release. When an item (e.g., architecture) from a previous release changes in the current release, the changes would be presented as an update (i.e., “updates as needed”). The program backlog drives release planning and determines the priorities of the system component requirements allocated to the release. Pro- gram and business stakeholders establish release priorities and objectives. Development teams derive stories from the requirements and commit to their team objectives, which are rolled up to program objectives.

*Input.* Release backlog (allocated from the program backlog).

***Output****.* Finalized release backlog and initial team backlogs. Confidence-based decision to move into DDIT iterations jointly made between government and contractor.

**Test Readiness Review (TRR)**

***When Conducted****.* Prior to system demo.

***Purpose****.* The TRR is conducted to determine if the system under review is ready to proceed into formal testing by deciding whether the test procedures are complete and verify their compliance with test plans and descriptions.In an Agile program, the preference is for the test organization to be continuously involved beginning at release planning, where the test strategy is discussed, the backlog is prioritized to align with test events, and the like. The TRR may be needed on a program where the test organization has not been closely involved with the iteration demos or if significant hardware is involved and/or the system is being tested at a government site. The TRR will review the results of the tests conducted during the DDIT iterations and identify workarounds, shortfalls, constraints and level of confidence for conducting a system demo.

***Input****.* Iteration test results, system demo test plan.

***Output****.* Decision to conduct system demo.

**Release Demo and Deployment Review**

***When Conducted****.* At the conclusion of the DDIT iterations.

***Purpose****.* The release demo and deployment review is a technical review of the release requirements and the as-built system design. It determines, based on the results of the release demo, if the release objectives and requirements have been met and the system performs as expected. If necessary, this may include a deployment review to determine the user’s and the system’s readiness to accept the new baseline.

***Input****.* Results of the system demo.

***Output****.* Decision on the adequacy of the as-built baseline and readiness for deployment.

### Instructions for providing the proposal

The following section contains the instructions for providing the proposal to the BUYER. The key elements of the proposal are answered in the cost and technical factors. During the evaluation of the proposals, the rating for each subfactor is rolled up into an overall rating for the technical factor.

The technical subfactors are Agile development process, systems engineering practices, and system test and delivery.

1. The first subfactor is essential in describing how the entire system lifecycle will be managed and implemented using Agile concepts and methods.
2. The second subfactor requests the detailed description of the Agile systems engineering and other discipline-specific practices to be used during the development. It would also include the schedule of work and key events that will enable the integration of the total solution. These engineering processes should reflect and be compatible with Agile concepts discussed in the other subfactors since the intent is to enable a holistic Agile development approach.
3. The final subfactor is used to evaluate how the completed system will be delivered to the user, which would include several steps such as acceptance, certification, installation, training, and transition from a previous system. Again, the engineering processes in this subfactor should reflect and be compatible with Agile concepts discussed in the other subfactors.

In all cases, how the subfactors work together (since this is a request for an Agile proposal) should be included in the evaluation criteria. The following sections contain example wording of the subfactors to be used in the RFP.

#### Subfactor 1 – Agile Development Process

Describe the Agile development lifecycle process to design and build “system name” based on the Agile development methodology. When describing the process, explain how the process implements the Agile principles listed in Appendix from Principles for Agile Development. The description should include specific details on the management of requirements decomposition and the product backlog (prioritization of work with the BUYER), the orchestration of the demonstrations (collaboration with the users on product evaluations during development), and the definition of “done” (meeting the contractor's product delivery requirements defined in the CDRLs). Include shorter but more frequent integrated client/developer meetings, user involvement, and other items specific to the Agile approach.

#### Subfactor 2 – Systems Engineering Practices

Describe systems engineering and any other specific engineering processes required to develop the architecture for “system name” not included in Subfactor 1. In addition, system capabilities and features roadmap should be provided that maps capabilities and features to the development iterations and system build deliveries. This subfactor should also include further details about meeting the entrance, exit, and review criteria for the technical reviews.

#### Subfactor 3 – System Test and Delivery

Describe the testing planned prior to delivering the software to the government. While some of this information should be included in the Agile development process subfactor, specific details for automated test, functional, integration, vulnerability, and defect tracking should be added here. Also, describe how support will be provided during the contractor development test and evaluation (DT&E) and operational testing and evaluation (OT&E). This should include test environment, test data/connections to external systems/stub to simulate outside systems, test scripts, support, training, and response to issues (e.g., deficiency reports [DRs]).

Describe the process for delivering the software and its associated CDRLs to the BUYER for fielding the system including documentation, distribution, tracking, transition of ops, training, certifications, build process, archiving, and recovery.

### Decision for the award

This Section explains how the decision for the award will be made. As with traditional proposals, the evaluation of Agile proposals will be based on the following factors:

* Technical
* Past Performance
* Cost/Price
* Cost/Price Risk

Within the technical factor, the subfactors are the key discriminators to support a meaningful evaluation of competing proposals. Since these subfactors were all objective, they would receive an adjectival rating such as outstanding, satisfactory, marginal, and unsatisfactory. If a subfactor is found to be unsatisfactory, it must be rated unsatisfactory.

Table 2 provides an example of definitions for the adjective ratings.

Table 2: Example Definition for Adjectival Ratings

|  |  |
| --- | --- |
| Adjectival Rating | Definition |
| Outstanding | Proposal significantly exceeds requirements in a way that benefits the government or meets requirements and contains at least one exceptional enhancing feature that benefits the government. Any weakness is minor. |
| Satisfactory | Proposal meets requirements. Any weaknesses are acceptable to the government. |
| Marginal | Proposal contains weaknesses or minor deficiencies that could have some impact if accepted. |
| Unsatisfactory | Proposal does not comply substantially with requirements. |

Based on the example subfactors used in the previous section, the following table are examples of how the subfactors could be evaluated:

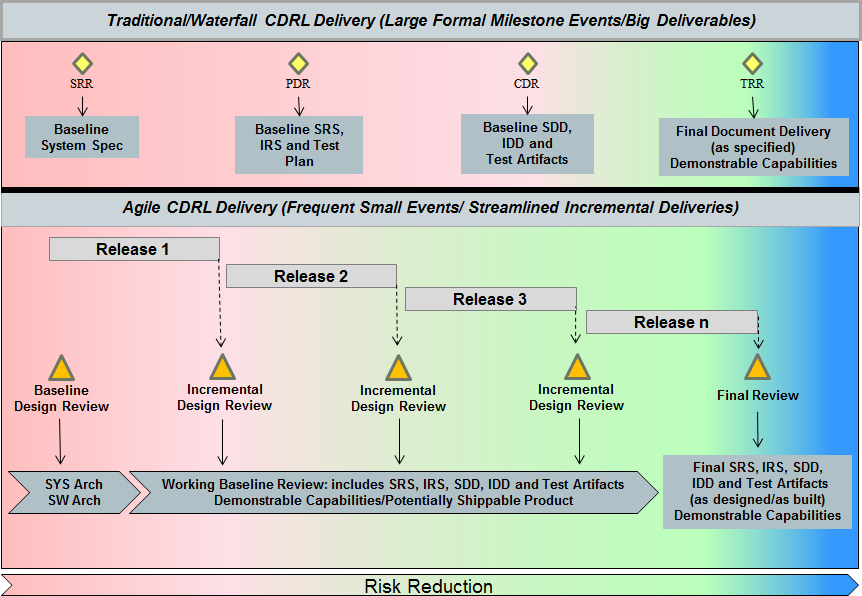
|  |  |
| --- | --- |
| Subfactor |  |
| Subfactor 1: | Agile development process subfactor will primarily be evaluated by how the proposed system development process will adhere to each of the principles of Agile development and the government’s Agile development approach. Additionally, the proposal will be evaluated on how the contractor will partner with the customer to manage the product backlog with the program office; provide software demonstrations to the program office and the user; and if the artifacts to be provided at the end of each sprint meet the program office’s definition of done. |
| Subfactor 2: | Systems engineering practices subfactor will be evaluated by how the proposed systems engineering process will be performed as part of a holistic Agile development approach while meeting the government expectations for technical reviews. Also, this evaluation will include the review of the capabilities and features roadmap and its mapping to the iteration schedule, the planned releases for integration and test, and planned schedule of activities to complete the delivery of the system. |
| Subfactor 3: | System test and delivery subfactor will be evaluated by activities, test events, and delivery of artifacts required to perform government developmental and operational testing and fielding of the system to the user. In addition, details about the test environment and supporting test structure such as test scripts, test data, and simulated connections to external systems will be evaluated for meeting government test and security requirements. Also, this subfactor will evaluate the use of automated tools used for testing throughout the development process and automated tools for installing and sustaining the software. |

### CDRLs - Contract Data Requirements List

The number and type of Contract Data Requirements List will depend on the program environment. The same approach of mindful tailoring should be applied to the CDRLs the program needs. Due to the anticipated close and continuous coordination between the government and the contractor, the number of formal deliveries of the CDRLs may be reduced (e.g., real-time information sharing between the contractor and the government via an integrated data environment). However, all items that are needed for the operation of the system, including the software, user manuals, and technical data for sustainment activities, must be on the CDRL.

For development and design documentation such as the system requirements specification and software design description, additional care needs to be taken to ensure the due dates reflect the incremental development and delivery of the products. The design documentation will reflect the “as-built” product not the “as-designed” product. Keep in mind that the contractors will most likely employ rapid documentation techniques such as auto-generated documents and wikis.

Access to any joint integrated data environment should be provided to the BUYER so that it has visibility into the technical information to meet the delivery requirement. The use of these constructs should be explained in their overall Agile development management plan. Formal deliveries should be aligned with the program cadence.

As shown in Figure 5, deliveries on an Agile program occur more frequently at smaller events than in a traditional waterfall program. As such, much of the documentation will be delivered and reviewed while it is in-process with the maturity of the CDRL increasing over time. This can reduce risk earlier than in a traditional waterfall approach as the BUYER can see early versions of the system and its documentation, then watch as risks are burned down during each subsequent iteration.

### Cost Proposal and Schedule Instructions

In a traditional waterfall model, the scope is fixed with cost with schedule and/or quality being variable. In a “customary” Agile program, cost, schedule, and quality are fixed with the scope (capabilities and features) being prioritized. Agile is flexible, however, and allows for several approaches to be used in the RFP and the cost proposal in particular. Specifically, the RFP must define the deliverable scope and, additionally, what is fixed and what is variable. The structure and approach of the contract should be based on the driving needs of the particular government entity responsible for the contract.

**Defining Deliverable Scope—What’s Fixed, What’s Variable?**

The first option, referred to as “customary” above, is to give a contractor a fixed schedule and budget while enabling the content to be variable. This is achieved through prioritization of the backlog to ensure that if there are capabilities or features that are not completed within the allocated time and cost, they are of lower priority. In this case, the BUYER is essentially buying “development capacity” for some specified period or multiple periods. This approach is similar to a “time and materials” contractual approach.

A variant of, which recommends an initial period (on the order of six months) using the time and materials approach to establish a velocity for the Agile team and then moving the contract to firm-fixed-price (FFP) periods. This approach lends itself to software-intensive systems and may not be appropriate for systems where hardware development is involved.

Another approach is to use a multi-year indefinite-delivery, indefinite-quantity (IDIQ) contract with task awards set up on a short fixed schedule (on the order of six months). A SOO describing the desired functionality (in terms of a prioritized backlog) is provided to the contractor. The contractor then proposes a PWS to address the prioritized backlog for the upcoming task order period. It is somewhat expected that not all low-priority backlog items will be completed before the end of the task order. Near the end of the existing task order, the next task order is developed with a revised and reprioritized backlog. This model enables to be very responsive to new mission needs.

In highly complex systems or systems requiring hardware development, a hybrid model may be more appropriate. In this case, the high-level capabilities are fixed by the BUYER, the features and their associated priority (prioritized backlog) are developed jointly between the BUYER and the contractor, and the stories (the “how”) are totally within the contractor’s control. Changes to capabilities or features usually require a contract modification (if the total scope is going to increase) or trading (reducing) other scopes so as not to impact the overall cost and schedule of the program. In this case, an FFP-type development contract is not normally appropriate. Cost plus fixed-fee (CPFF) or a similar contract vehicle would be applicable.

**Parametric Bidding: Waterfall Actuals Don’t Necessarily Apply**

Historical actuals and other metrics are key to the cost estimate. The government must understand the metrics/measures supporting the contractor’s estimate to accurately assess a contractor’s cost bid. Waterfall actuals (and to a large extent waterfall metrics) don’t necessarily carry over to an Agile development program. Actuals reflect the particular skill mix within specific disciplines and efficiencies that often correlate to different development stages in the waterfall model. In an Agile program, the focus is on optimizing the skill mix and defining metrics that provide overall value to the customer. As such, the staffing mix and staffing curves will likely look quite different when comparing Agile and waterfall programs.

When looking at historical actuals from previous Agile programs, the higher the percentage of the program that was performed using the Agile methodology, the more accurately they can be assessed and the less likely they will be broken out by discipline. If the breakdown by discipline can be determined, it will likely be inconsistent with similar programs executed using the waterfall model.

Using Story Points for Bidding

First the authors must warn that using story points for bidding is not a good practice unless there is a standard and normalized process to determine story point metrics. This is because story points are typically a sizing measure unique to each team and are not used to compare productivity across teams but rather for the team to estimate work and determine if they are improving.

However, if monetized story points are the expected work unit in the RFP, the historical actuals must be examined from the perspective of what the contractor is using as a basis for the expected productivity. The most reliable estimates will come from existing Agile teams with historical data on their velocity for the equivalent type of development. The next most reliable estimate would come from a prior Agile program. If no existing or prior Agile programs are cited, the government could use the approach cited previously, which is to start up the program using a time and materials approach to “set” the expected velocity for the Agile teams per the TechFAR. In these cases, remember to account for slower velocities during the start-up of the program, as the teams require some initial time to ramp up.

**Schedule**

Unlike a traditional schedule, which is often organized in a serial fashion following the waterfall process (i.e., design then code then test) by discipline, an Agile integrated master schedule should be focused on the development of system capabilities and features. Known dependencies between features should be clearly identified. Completion criteria for each feature should be clearly understood and the associated progress towards working capabilities (which is the overall goal) should also be easily discernible. An Agile schedule will employ a rolling baseline where details are developed two events forward. This just-in-time planning approach defers commitment-level decisions until they are ready to execute, preserving trade space and agility to respond to change. The temptation to include stories in the IMS should be avoided. Capturing that low level of detail overly constrains the contractor and likely ensures that the IMS will be in a constant state of change. Stories should be managed in the Agile toolset at the working-team level. As such, these

tools will provide the quantifiable backup data (sometimes known as inchstones) used to update the IMS.

# Project Goals

*Clearly identify what you hope to accomplish with this project and what you see as a “win” so everyone is on the same page.*

The goals of this project include:

i. [INSERT COMPANY GOAL #1]

ii. [INSERT COMPANY GOAL #2]

iii. [INSERT COMPANY GOAL #3]

To reach these goals, [YOUR COMPANY] is now accepting bids in response to this Request for Proposal.

# Context

# Scope of services and deliverables

*The details of the project are essential in an RFP. In the same way you don’t want to use vague questions, a vague project scope won’t help you find the right vendor either. Spend time really detailing your project scope so there’s no surprises later on.*

*Include a description of the project and a detailed scope of work here. The description can be in paragraph form, a bulleted list, or a combination of both.*

[EXPLAIN SCOPE OF PROJECT IN A FEW PARAGRAPHS AND/OR BULLET POINTS.]

## Scope

## Stories

* **Prepare benchmark stories**/use cases with business and IT people (description, acceptation criteria, estimates) that will be mandatory to implement and can serve as a basis for the “vendors’ comparison”.

*Although the velocity among the teams is not comparable, we’ve chosen to set story etalons to compare somehow the speed as well as thinking of vendors. We described several basic and integration stories, set their effort estimation (with IT guys), set them as mandatory and kept the rest of the scope open for modifications to see vendors’ approach, communication style, proposal ideas and business thinking. It was risky, but one of the best decisions at the end!*

As <a type of user>, I want to <perform some task> so that I can <achieve some goal>.

|  |  |  |  |
| --- | --- | --- | --- |
| **USER STORY ID** | **As a** *<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1 | Project manager | View a status report from each team member | Ensure the project stays on track. |
| 2 | Employee | Be reminded of upcoming deadlines | Complete my tasks on time. |
| 3 | Director | See the big picture view of department work | Stay in the loop. |

## USER EPIC

When user stories are created from each epic, add the User Story IDs into the columns. Add additional columns if needed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USER EPIC ID** | **AS A** | **I WANT TO** | **USER STORY ID 1** | **USER STORY ID 2** | **USER STORY ID 3** |
| 1 |  |  | As a user, I want to manage all my appointments from my phone. | As a user, I want to see my family calendar and business calendar together. | As a user, I want to have full functionality in the reminder without opening the app. |
|  |  |  |  |  |  |

## Benchmark estimates

## Project Management

## Infrastructure – IT Environment

*Describe your technology environment – all of the tools currently employed on your current site any backend system integration requirements, and any perimeter information you feel necessary for an agency to know. Also include in this section:*

Technologically, this is what we want to keep:

Technologically, this is what we never want to deal with again:

Technology that we are interested in exploring:

* **Prepare IT environment** for all vendors invited to RFP (e.g. to verify the

integration with selected internal system/interfaces).

*You can transfer this responsibility to vendors by proposing external IT environment, but you will not verify their ability to integrate with your systems if you plan to use your internal environment. If you plan to use some cloud platforms, use it from start. This setup depends a lot on the nature of the project and your intention with it.*

## Technology preferences

* **State technology preferences and limitations if any.** Many customers have their IT strategy, preferred architecture, communication systems (e.g. SOA or MQ), databases and people skills (which is important for daily support).

*If you request specific technologies, state it in RFP conditions. But also keep your RFP open for new possibilities and technologies that can improve your IT environment as well as teams productivity.*

## Functional Design

## Product Requirements

## Development

## QA

## Product Management

## Definition-of-Done (DoD)

* **Prepare your Definition-of-Done** (DoD) for the RFP. Based on their achievement you will pay vendors for their effort (for source code ownership). Use standard DoD items like demo happened, code stored in and built from agreed repository, code reviewed by your IT people, etc.

*The amount of money you pay is up to you (remember, in traditional RFP you pay vendors nothing for their time spent on RFP answers), but it should be fair price for both sides. Depending on the length of the RFP project it can be 5.000-50.000 Euro per vendor.*

# Project Timeline

One of the keys to Agile Software Development is an insistence on a fluid development cycle that allows for and adapts to outside input and project changes during every phase of the project. As a result, we do not offer completion dates. Instead, we focus on completing each phase during a given software development project as quickly as possible, while maintaining a strong focus on design and technical excellence.

This project will be segmented into the following phases:

* Requirements Gathering
* Technical & Design Specifications
* Design Mockups
* Development
* Testing
* Delivery
* Approval

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RFI Preparation** | **Agile RFP mini project** | | | | **Project** |
| Internal preparation | Sprint1 | Sprint2 | Sprint3 | Sprint4 | Agile contract |
| Goals | Initial kick off | Planning with P.O | Planning with PO | Planning with PO | Velocity |
| Context | Release planning | Consult with PO, business, IT , end users | Consult with PO, business, IT , end users | Consult with PO, business, IT , end users | Roadmap as baseline |
| Stories | Environment preparation | Review | Review | Review |  |
| IT environment | RFP roadmap proposal |  |  |  |  |
| Involve  IT  Business  PMO  Procurement  Agile coach | Demo: RFP roadmap | Demo: 1st version of app | Demo: 2nd version of app | Demo: 3RD version of app |  |
| Meetings with vendors  Announce Agile way of working |  |  |  |  |  |

Target Deliverable Schedule

*Do you have a target deliverable schedule in mind? This needs to be included in your RFP so potential vendors can properly gauge if they have the resources and bandwidth to complete the job on time.*

**Final Project Due:** [PROJECT DUE DATE]

The expected project completion date is [ADD DATE]. If this date needs to be adjusted, please include your readjusted proposed date, as well as your reasoning for shifting the schedule. All proposed date changes will be considered.

* **Prepare the calendar of all RFP ceremonies:** kick-off workshop, separate planning sessions with vendors, consultancy with PO, IT, business users (and others if needed) and demonstration sessions.

*We discussed common or separate planning sessions also with vendors and decided to have separate session with each of them. The main reason was the voice of vendors. They wanted to use their domain expertise and knowledge during the planning sessions to propose feature updates, look and flow and we liked it.*

# KEY PERSONEL

The following personnel will be involved in this software development project:

|  |  |
| --- | --- |
| **Name** | **Role** |
| [PM.Name] | Project Manager |
| [Team1.Name] | Team Member |
| [Team2.Name] | Team Member |
| [Team3.Name] | Team Member |
| [Team4.Name] | Team Member |
| [ProductOwner.Name] | Product Owner |

Please note that while this is the projected core project team, [Sender.Company] reserves the right to substitute or supplement members of this team as necessary in order to complete the software development process in a timely manner without sacrificing technical or design excellence.

# Suggested structure for vendor bids

# Selection criteria

* **Prepare RFP evaluation criteria and templates** to be used by your internal people during the sessions (mostly demo) to evaluate the vendors. This is usually based on your RFP selection criteria, therefore items like business knowledge, added value, usability, Agile skills demonstrated are used. This would be later on completed by proposed price for the future project development, delivered value in RFP mini-project and support conditions.

# Project Costs

This project will be billed on a time-and-materials basis. Final project costs are not projected or guaranteed, a result of the agile software development approach that allows for changing requirements, methods, and designs throughout a project.

In addition to time and materials billed for software development purposes, [Sender.Company] will invoice [Client.Company] for any pre-approved travel-related costs at cost, and will make a reasonable effort to secure travel and related accommodations at affordable rates.

# Pricing

This project will be billed monthly on a time-and-materials basis. Each month, [Client.Company] will receive a net-30 invoice for services rendered during the previous 30 day period, accompanied by a detailed breakdown of hours utilized by each member of the [Sender.Company] team.

The table below includes the hourly rates for each role within our agile software development team:

|  |  |
| --- | --- |
| **Name** | **Price** |
| Project Manager | $0.00 |
| Senior Developer | $0.00 |
| Junior Developer | $0.00 |
| Senior Designer | $0.00 |
| Junior Designer | $0.00 |
| Systems Architect | $0.00 |
| Software Tester | $0.00 |
| Systems Analyst | $0.00 |

# ANNEXURES

## Annexure I - Existing Roadblocks Or Technical Issues

*Here, you’ll want to outline any time, resources, or other constraints that will affect both the proposal and the project.*

*A successful RFP is clear about any technical issues or possible roadblocks, such as: Are you dealing with custom coding or an outdated platform? Does your team have limited resources?*

*By explaining these up front, potential vendors will know exactly what they’re getting into. You’ll weed out contractors who can’t handle the task due to those constraints, but you’ll also connect with companies that know how to work around these common issues with skill and finesse.*

*Remember, it’s far better to find this out now than after you’ve accepted the bid and started work.*

[OUTLINE YOUR ROADBLOCKS OR TECHNICAL CONSTRAINTS.]