ADMINISTRAÇÃO PÚBLICA DIGITAL

ELIMINANDO O RISCO DA ADMINISTRAÇÃO PÚBLICA DIGITAL

Robin Carnahan,
Randy Hart e Waldo
Jaquitharal

TRADUZIDO E ADAPTADO PARA PORTUGUÊS POR

PEDRO TEIXEIRA

Table of Contents

- O pedido de proposta incluirá requisitos de como o sistema irá operar? Em caso afirmativo, quantos requisitos estão incluídos?
- Quanto tempo você espera que o pedido de proposta demore?
- Você prevê a emissão de um contrato de preço fixo ou de tempo e materiais?
- Qual valor será entregue aos utilizadores em seis meses?
- Quem será o dono do produto?
- Qual processo de desenvolvimento de software será usado?
- Na equipa que preparou esta solicitação, quem tem experiência no desenvolvimento de software?
- Com que frequência o trabalho será colocado em produção?
- O projeto irá automatizar os testes? Integração? Entrega? Testes de segurança?
- Quanto custarão os pedidos de alteração?
- Como você saberá se o projeto está no caminho certo e se os empreiteiros estão cumprindo o prometido?
- Quem será o proprietário do software?

Introdução

Apenas uma pequena percentagem dos grandes projetos de software do governo são bem-sucedidos. De uma forma geral, os projetos estatais de tecnologias de informação costumam correr mal porque o estado não possui conhecimentos básicos sobre o desenvolvimento de software moderno e dependende de processos de aquisição desatualizados. Todos os anos, o governo gasta muito dinheiro em projetos para manter e modernizar os sistemas de informação usados, mas os esforços para modernizar esses sistemas antiquados fracassam a uma taxa alarmante e com grande custo para o nosso orçamento.

A nossa administração pública depende cada vez mais de software e hardware modernos para implementar programas e fornecer serviços essenciais ao público, e o sucesso de qualquer iniciativa depende do sucesso da infraestrutura de software subjacente. Todas as entidades governamentais enfrentam desafios semelhantes, enfrentando restrições de orçamento e de pessoal enquanto lutam para modernizar os sistemas de tecnologia antiquados que estão desatualizados e são inflexíveis, caros e ineficazes. Funcionários e serviços da administração pública muitas vezes contam com os mesmos processos antigos que conduziram aos problemas em primeiro lugar.

O público merece um governo que forneça a mesma tecnologia de topo que consegue obter no mercado comercial. A confiança no governo depende disso.

Este manual foi elaborado para executivos, especialistas em orçamento, legisladores e outros tomadores de decisão que financiam ou supervisionam projetos de tecnologia do governo e implementam a tecnologia necessária para apoiar a administração pública. Ele pode ajudá-lo a preparar esses projetos para o sucesso, fazendo as perguntas certas, identificando os resultados certos e,

igualmente importante, capacitá-lo com o conhecimento básico dos princípios fundamentais do design e construção de software moderno.

Este manual também fornece as ferramentas de que precisa para começar a lidar com problemas relacionados, tais como:

- A necessidade de simultaneamante usar, manter e modernizar sistemas antiquados;
- Organizações isoladas e culturas avessas ao risco;
- Longos ciclos de orçamento que nem sempre correspondem às práticas modernas de design de software;
- Ameaças à segurança;
- Contratação, pessoal e outras restrições.

Este documento foi escrito especificamente para a aquisição de software feito à medida, mas é importante reconhecer que o software comercial genérico (SCG) é frequentemente personalizado e que o Software como Serviço (*Software as a Service* ou *SaaS*) frequentemente requer personalização. Assim que qualquer personalização for feita, a maioria dos conselhos neste manual aplicam-se também a essas ofertas comerciais. (Consulte "Cuidado com a armadilha de software comercial personalizado" para obter detalhes.)

Como funcionários governamentais, devemos ser bons administradores do dinheiro público, exigindo ferramentas digitais sustentáveis, económicas e fáceis de usar pelo público e funcionários da administração pública. Este manual ajudá-lo-á a fazer exatamente isso.

Sobre os autores

> *Este documento foi traduzido e adaptado do original "De-risking Government Technology - State Field Guide" com a autorização dos autores. Abaixo segue uma breve descrição dos autores originais deste documento:*

Trabalhamos para a 18F, parte da quipa de Serviços de Transformação de Tecnologia da Administração de Serviços Gerais (GSA) nos Estados Unidos da América. Coletivamente, nós três temos muitos anos de experiência em compras para o governo, desenvolvimento de software e cargos eletivos a nível estadual.

No trabalho financiado pelo GSA 10x, passamos um ano explorando como reduzir o custo das concessões federais de tecnologia aos estados, enquanto melhorando simultaneamente os resultados. Subvenções federais fornecem 31% dos orçamentos estatais, e como um grande investidor em grandes projetos de sistemas de software estadual, o governo federal está profundamente interessado em garantir um bom retorno sobre esse investimento. Para conseguir isso, reunimos com legisladores estatais, equipas fiscais legislativas, funcionários do orçamento estadual, contratantes e consultores de política governamental. Este manual foi originado do processo de aprender e ensinar centenas de pessoas em dezenas dos estados. Estamos gratos às muitas pessoas que contribuíram com seus tempo e conhecimento ao longo desse processo.

Robin Carnahan, Randy Hart e Waldo Jaquith

Princípios Básicos

Princípios básicos do design moderno de software

As probabilidades de sucesso de um projeto de tecnologia aumentam quando os líderes governamentais "não técnicos" que o financiam e supervisionam entendem seis conceitos básicos do desenvolvimento de software moderno: **design centrado no utilizador**, **desenvolvimento ágil de software**, **DevOps**, **construção com peças independentes**, **contratação modular** e **propriedade do produto**. O leitor não precisa ser um tecnólogo para entender esses conceitos gerais. Depois de entendê-los, parecerá que ganhou um novo superpoder, permitindo-lhe distinguir o jargão e os detalhes técnicos e manter o foco no básico para orientar com sucesso qualquer projeto de software.

Design centrado no utilizador

Todo o desenvolvimento de software deve ser centrado nas necessidades dos utilizadores finais reais do software, as pessoas específicas que devem usá-lo. Esses "utilizadores finais" podem ser cidadãos-alvo, funcionários de atendimento ao público, funcionários, outros funcionários do estado ou qualquer um de inúmeros outros grupos.

Projetar **com e para** os utilizadores reduz os riscos do projeto, garantindo que o software está resolvendo problemas reais (ao contrário do que algumas partes interessadas pensam que os problemas realmente são). Esses problemas são identificados por meio de uma variedade de táticas de investigação que incluem entrevistas e testes de usabilidade.

No design centrado no utilizador, todo o trabalho está a serviço das necessidades dos utilizadores finais. Esse trabalho é identificado e priorizado em colaboração próxima e regular com os utilizadores finais e é informado por, mas não subserviente a, quaisquer restrições técnicas. (Ou seja, o objetivo do trabalho é entregar valor aos utilizadores, o que envolve lidar com as realidades das linguagens de programação ou software de servidor aprovados, mas o trabalho nunca deve ser omitido por causa da perceção de que restrições técnicas o tornariam impossível.) A equipa técnica e os utilizadores finais revêem regularmente o trabalho, conforme ele vai sendo executado, e o trabalho de desenvolvimento no novo software não é considerado concluído até que os utilizadores finais concordem que suas necessidades foram atendidas. Projetar com e para os utilizadores reduz os riscos do projeto, garantindo que o software está resolvendo os problemas dos utilizadores.

Desenvolvimento ágil de software

Planos detalhados e de longo prazo para grandes projetos de software feitos à medida sempre foram a norma no governo. Mas, como os engenheiros de software e gestores de projeto aprenderam ao longo dos anos, esses planos nunca são cumpridos à risca. Eles precisam de muitas modificações caras, levando a pedidos de mais dinheiro para pagar "pedidos de alteração". Está na altura de os executivos do governo e responsáveis por orçamentos pararem de pedir planos detalhados de longo prazo e, em vez disso, fazerem o orçamento para projetos de software de uma nova forma.

O planeamento antecipado de um projeto inteiro é conhecido como desenvolvimento em "cascata". Imagine planear um mês de férias com a família numa viagem de carro pela Europa. Sob a metodologia de cascata, isso envolveria um planeamento antecipado da agenda de cada dia, incluindo percurso exato, a reserva de cada quarto de hotel, o pré-pagamento de todas as refeições, a pré-compra de ingressos para entrada em atrações, etc. Isso nunca funcionaria porque as coisas mudam, surgem opções inesperadas, e nenhuma

pessoa racional gostaria de tomar todas as decisões no início da jornada quando não sabe o que a jornada trará. Em vez disso, a maioria das pessoas planearia a rota geral a ser seguida e planearia apenas algumas paragens principais - os detalhes seriam resolvidos à medida que progredissem ao longo do caminho.

"Desenvolvimento ágil de software" refere-se ao uso dessa metodologia de planeamento de viagem para construir e modernizar sistemas de software. Em vez de depender de anos de planeamento caro e "compilação de requisitos" antes de começar a escrever software real, os projetos de desenvolvimento ágil são planeados apenas em linhas gerais, com uma descrição bem definida do objetivo geral do projeto e uma forte preferência por *apenas começar*. Uma equipa pequena, com poderes e automotivados (geralmente de 5 a 9 pessoas, incluindo desenvolvedores, gestores de produto, investigadores, escritores e / ou especialistas em segurança) dedica-se a cumprir esse objetivo, usando design centrado no utilizador, trabalhando em ciclos curtos para entregar software que realmente funcione.

No primeiro dia, a equipa planeia apenas o que fará nas próximas duas semanas. (A duração dos ciclos de um projeto pode ser tão breve quanto uma semana ou até quatro semanas - duas semanas é o mais comum.) Cada tarefa em que eles trabalharão tem a forma de uma "história de utilizador" - uma necessidade específica do utilizador revelada pela investigação. A coleção inteira de histórias de utilizador a serem trabalhadas é chamada de "backlog".

A equipa trabalha num grupo selecionado destas "histórias de utilizador" por duas semanas e, no final, a equipa analisa o trabalho que eles fizeram, testa-o com os utilizadores finais e, em seguida, planeia as próximas duas semanas retirando mais histórias de utilizador do backlog. Repetir. Cada um desses ciclos de duas semanas é denominado "sprint".

No início, o software que produzido pode não parecer muito (e pode até mesmo ser substituído por algo mais tarde), mas irá informar gradual e

sistematicamente a abordagem técnica do projeto e ajudar a equipa a integrar o projeto de forma sensata no sistema pré-existente.

Software funcional é entregue no final de cada pedido de proposta, sem exceção - totalmente testado, totalmente documentado, pronto para ser usado. Dessa forma, o valor é entregue constantemente, até que o software seja bom o suficiente para ser usado amplamente. A equipa continua a trabalhar até cumprir todas as metas ou ficar sem dinheiro, o que ocorrer primeiro.

O fornecedor é pago pelo tempo de seus funcionários, não por um sistema de software. Tudo criado pelo fornecedor - software, documentação, investigação, projetos, *tudo* - é propriedade do governo, entregue ao governo no final de cada pedido de proposta. Mudanças tecnológicas, mudanças nas políticas governamentais, mudanças nas regulamentações, mudanças nas leis e mudanças nas prioridades da liderança - qualquer projeto planeado em grande detalhe no início não será capaz de se adaptar a essas mudanças e correrá um risco significativo de falha, derrapes de custo e prazo significativos ou "pedidos de mudança" caros.

Ao combinar a agilidade com o design centrado no utilizador, uma equipa de desenvolvimento pode iterar constantemente para resolver as necessidades dos utilizadores finais de maneiras que seriam impossíveis de aprender antecipadamente.

Propriedade do produto

A reapropriação dos projetos de software pelo governo requer que as equipas deste se concentrem nos resultados e não nas métricas. Isso significa mudar de algumas das práticas tradicionais do gestão de projetos para uma mentalidade orientada ao produto.

A palavra "produto" pode soar incomum num contexto governamental, mas é uma parte importante do jargão tecnológico. "Produto" é uma abreviatura para qualquer coisa que será criada: um site, uma aplicação iOS, uma aplicação web, etc. Embora a palavra faça mais sentido para uma empresa que está a vender um produto tangível, tudo o resto sobre este conceito traduz-se perfeitamente para o governo.

O dono do produto é a pessoa-chave para qualquer projeto de software e *deve* ser um funcionário do governo. Ele trabalha com utilizadores, partes interessadas, tecnólogos e o fornecedor para prever a direção do produto, com o objetivo de entregar valor aos utilizadores finais o mais rapidamente possível. Ele prioriza e define iterativamente o trabalho para a equipa de desenvolvimento, como parte do processo ágil. Ele mede o progresso em relação a indicadores de desempenho claros e comunica com as partes interessadas e a equipa multifuncional que está a construir o produto.

O dono do produto não precisa ser um tecnólogo forte. Em vez disso, ele deve conhecer os utilizadores do sistema, os objetivos e as restrições legais e políticas.

Um dono de produto forte garante que a visão é clara, a estratégia é clara e que há espaço para que as equipas que criam o software aprendam, assegurando-se de que estão construindo a coisa certa para, incrementalmente, mostrar valor aos utilizadores. Eles priorizam implacavelmente para garantir que o produto corresponda às necessidades do utilizador e que a atividade e a atenção sejam focadas nas necessidades de maior prioridade. Eles são habilitados pelas suas chefias para representar as partes interessadas e tomar de decisões rápidas sobre o produto, sem que haja a necessidade de muitas camadas de aprovação. Esse posicionamento garante que o dono do produto entende tudo o que a equipa de desenvolvimento está a fazer e que as necessidades do governo sejam totalmente representadas.

Isto é diferente da gestão de projeto típico em tecnologias de informação (TI) governamental. O dono do produto não terá cronogramas ou um plano detalhado de 5 anos. Mas ele terá uma visão dos resultados que serão entregues aos utilizadores e terá um caminho para a sua execução. A sua função principal é entender o que a equipa de desenvolvimento está a fazer e garantir que se encontra o equilíbrio certo entre as necessidades do governo e as necessidades dos utilizadores finais.

É possível que um proprietário de produto principiante aprenda durante o decorrer do processo, mas é melhor que seja treinado com antecedência. Existem muitas fontes de treino ágil e *scrum*, algumas específicas para proprietários de produtos, entre séries de vídeos do YouTube e aulas presenciais de vários dias para se tornar um "Dono de Produto Certificado Scrum". Quanto mais importante o projeto, mais formal e rigoroso deverá ser a formação do dono do produto.

DevOps

Historicamente, as equipas de desenvolvimento de software são separadas das equipas de TI responsáveis por operar o software. Um fornecedor podia passar anos a criar um novo software e, em seguida, uma equipa de TI do governo (ou um fornecedor que desempenhe essa função) podia exigir muitos meses de trabalho para fazer com que o software funcionasse corretamente nos seus servidores. Este processo geralmente é acompanhado por frustração e acusações, e pode levar a falhas e atrasos no projeto. Para resolver isso, as agências governamentais costumam insistir que o fornecedor que constrói o software também o aloje indefinidamente na sua própria infraestrutura, o que tem o efeito de descartar a maioria dos fornecedores de software (que não estão no negócio de alojamento) e criar uma dependência do fornecedor, com os altos preços e riscos associados. Contar com essas abordagens antigas vai render-lhe menos e custar mais do que se adotasse as metodologias de software modernas que são padrão no setor privado.

A maneira de resolver isso é com DevOps. DevOps é a prática de coordenar o trabalho desses dois grupos para automatizar os testes ao software e disponibilizá-lo num servidor ativo onde as pessoas possam usá-lo - misturando **desenvolvimento** de software e **operações** de sistema. Os programadores escrevem uma série de processos automatizados para garantir que o software funciona corretamente em produção, durante a criação do próprio software. Os programadores não podem simplesmente dar o seu trabalho por concluído e entregá-lo à equipa de operações do sistema e declarar que "funciona para nós" - eles são responsáveis, tanto prática quanto contratualmente, por que o seu código funcione corretamente no ambiente designado.

Há uma grande probabiliadde de que a maioria do software que você usa todos os dias, seja no telefone ou no computador, tenha sido escrita exatamente assim. Em DevOps, a qualidade e a segurança do software é testada automaticamente, fundir o trabalho de vários programadores é automático e disponibilizar o software completo nos servidores é automático.

Construindo com peças desacopladas

Projetos de software grandes e complexos tendem a entrar em colapso sob o peso da sua gestão. Nenhum programador pode compreender todo o sistema para o qual está contribuindo, e cada membro de equipa adicionado aumenta a complexidade das interações da equipa inteira, obrigando a novas funções de supervisão tais como "arquitetos de software", com quem os programadores devem conferir antes de fazer qualquer trabalho. Os colaboradores precisam de coordenar-se entre si cuidadosamente para evitar conflito e sobreposição entre os seus esforços. À medida que uma equipa cresce, eles são forçados a gastar cada vez mais tempo na gestão do projeto, assim diminuindo o tempo que passam a realizar o trabalho propriamente dito.

De forma a evitar este destino, é mais inteligente dividir grandes projetos num punhado de pequenos projetos de software quase independentes. Neste modelo, cada componente comunica com outros componentes por meio de padrões simples e modulares, de forma a que qualquer peça possa ser trocada a qualquer momento. Em vez de construir um monólito que todos lamentarão dentro de alguns anos, você constrói um pequeno ecossistema, no qual cada peça pode ser atualizada e modificada facilmente, conforme a necessidade. Cada componente é mantido por uma única equipa ágil, que documenta a interface de programação (API) do componente - as regras gramaticais que outros componentes podem usar para se comunicar com ele. A necessidade de coordenação entre equipas é mínima, porque elas podem simplesmente seguir a documentação da API de outros componentes com os quais precisam interagir.

Quando cada componente usa APIs abstratas (pense nelas como padrões comuns para usar essa tecnologia), isso é conhecido como "arquitetura orientada a serviços" (SOA) . Este é o mesmo conceito de "peças intercambiáveis" que tornou possível a revolução industrial. Os acoplamentos padronizados são o conceito subjacente à computação em nuvem, tomadas elétricas, USB, Legos, comboios e inúmeros outros produtos e práticas modernas.

Construir sistemas de TI usando partes desacpoladas, conectadas por APIs abertas e disponíveis, é a "solução mágica" que permite construir sistemas flexíveis e sustentáveis que atendem às necessidades do utilizador e custam menos com o tempo.

Contratação modular

Ao combinar o design centrado no utilizador, a agilidade, a propriedade do produto, DevOps e a construção com peças desacopladas, é possível partir um contrato grande e arriscado num punhado de contratos menores. Um contrato deve ser pequeno o suficiente para que a agência não tenha escrúpulos em não

continuar a dar mais trabalho a um fornecedor com baixo desempenho, substituindo-o por um novo fornecedor. Nesse caso os restantes fornecedores poderão continuar a trabalhar, e a perda total de velocidade será mínima. Um novo fornecedor não deveria ter dificuldade em substituir o antigo, já que o antigo estava entregando software completo, documentado e testado a cada duas semanas. Outro benefício é que pequenos contratos podem chegar abaixo do limite de aquisição simplificado da administração pública, o que significa que as organizações podem rapidamente escrever um convite, publicá-la e conceder um contrato.

Esta abordagem exigirá coordenação e aceitação dos responsáveis pelas aquisições e contratações. Estes funcionários geralmente estão acostumados à abordagem tradicional de terceirizar projetos de TI: uma grande aquisição baseada em longos documentos de pedidos de proposta, solicitando dos fornecedores propostas longas, certificações e qualificações desatualizadas. Geralmente, os fornecedores que usam métodos ágeis e centrados no utilizador não têm ideia do que seja "CMMI" ou "ISO-9000" - tais padrões já não são considerados melhores práticas para a criação de sistemas de software flexíveis e económicos. Esta é uma barreira de entrada para muitos dos fornecedores que poderiam ser contratados pelo governo mas que não desejam gastar todos os recursos necessários para se qualificar e escrever uma proposta.

Os processos modernos de desenvolvimento de software são baseados no design centrado no utilizador, desenvolvimento ágil de software, propriedade do produto, DevOps, construção com peças desacopladas e contratação modular. Ao compreender esses conceitos básicos, você está em uma ótima posição para entender como orçamentar um projeto de desenvolvimento de software da forma mais eficaz e para entender o restante deste manual.

Orçamentar e supervisionar projetos de tecnologia

Pense sobre o risco de uma nova maneira

Nas últimas décadas, as agências governamentais deixaram de usar funcionários internos, contando com fornecedores externos para construir as suas ferramentas de tecnologia digital. As decisões para fazer isso foram baseadas em trocas compensatórias que pareciam opções de menor risco - muitas vezes impulsionadas por uma capacidade limitada no governo e promessas de ferramentas mais baratas "disponíveis no mercado" oferecidas por contratantes.

No entanto, aprendemos com exemplos de vários projetos que, embora o governo possa facilmente externalizar o trabalho de criação de novos sistemas digitais, ele não pode externalizar o risco de falha. Projetos que correm mal refletem-se também no governo, não só em empreiteiros ou fornecedores de software.

O governo é, em última análise, responsável pela sua missão, e por consequência, as entidades governamentais precisam de ter controle e responsabilidade pelos projetos que suportam essa missão. O problema que um convite a orçamento de TI visa resolver não é um problema técnico; é um problema relacionado com o cumprimento da missão do governo, e a tecnologia é simplesmente um meio para esse fim.

Isso não significa que os departamentos precisem de realizar todo o trabalho internamente; significa que os departamentos precisam de definir expectativas claras sobre os resultados humanos e padrões técnicos relacionados com a segurança de dados, uso, interoperabilidade, monitorização e avaliação.

O conhecimento técnico é acessível e abundante, mas saber como administrar um serviço ou agência estatal é uma habilidade rara e valiosa. O governo deve abraçar e assumir a sua responsabilidade e o risco de falha, reconhecendo que os fornecedores de tecnologia são contratados apenas para ajudar e devem poder ser facilmente substituíveis se não tiverem o desempenho esperado.

LISTA DE VERIFICAÇÃO

- O projeto tem um dono do produto dedicado e autorizado, que é funcionário do departamento / serviço - não é um contratado e não é funcionário da agência de TI do estado;
- As partes interessadas reconhecem que a abordagem existente (desenvolvimento em cascata) falha na maioria das vezes e que mudar para o desenvolvimento ágil e aquisição modular é, de fato, significativamente menos arriscado;
- As partes interessadas consideram os fornecedores externos como peças substituíveis para cumprir um objetivo, em vez de como os "donos" de um projeto ou do seu resultado.

QUESTÕES-CHAVE

- Existem funcionários do governo identificados e treinados (não contratados) que servirão como donos de produtos dedicados e com poderes para definir a direção, priorizar e supervisionar o trabalho da equipa de desenvolvimento?
- Há uma cadeia de apoio para esta nova abordagem dentro do departamento e das suas chefias superiores, TI central, escritórios jurídicos e de compras, bem como o legislativo? Alguma dessas partes interessadas é capaz de bloquear a adoção desta nova abordagem? Em caso afirmativo, qual é o caminho para escalar os problemas, garantindo o alinhamento e evitando que esses bloqueadores internos coloquem o projeto em risco?
- Como é que o departamento / serviço está a assumir a responsabilidade por liderar o projeto e assumir os seus resultados, em vez de tentar externalizar o risco para um fornecedor através de um processo de contratação?

Adquira serviços, não software

Não pense em adquirir software personalizado como comprar uma *coisa.* Em vez disso, pense nisso como comprar um *serviço:* o serviço de uma equipa de programadores, designers e investigadores realizando o trabalho priorizado pelo dono do produto. Este reenquadramento leva a uma abordagem completamente diferente - uma abordagem muito mais simples - para o convite e para o contrato, e é uma distinção importante para os oficiais contratantes.

O seu Pedido de Proposta (PP) (???) deve descrever o objetivo geral do trabalho e deve incluir uma primeira tentativa de backlog do produto - uma lista do trabalho que será feito - reunida pelo dono do produto. Deve ser semelhante a uma lista de histórias de utilizadores - tarefas a serem realizadas para atender às necessidades dos utilizadores finais - que o trabalho provavelmente satisfará, claramente rotulada como indicativa dos tipos de trabalho que provavelmente estarão envolvidos, em vez de um âmbito de trabalho fixo. O PP também deve reconhecer que haverá mudanças constantes no trabalho com base na mudança de prioridades e na investigação contínua; a mudança é esperada e é fácil mudar o software quando ele é construído utilizando princípios de desenvolvimento de software modernos.

O pedido de proposta deve apresentar uma Declaração de Objetivos em vez de uma Declaração de Trabalho - ou seja, em vez das especificações de um produto que o fornecedor deve produzir, o pedido de proposta deve declarar os objetivos do projeto. Usar uma declaração de objetivos (DO) em vez de um declaração de trabalho (DT) elimina "pedidos de alteração" dos fornecedores, porque o âmbito do trabalho é tudo o que a equipa deve fazer. (Se um fornecedor ostensivamente "ágil" menciona "pedidos de alteração", isso deve ser considerado como um sinal de alerta.)

Para garantir que os fornecedores entregam um produto que vá de encontro às especificações técnicas necessárias, é importante que o PP inclua um Plano de

Vigilância de Avaliação da Qualidade (PVAQ) apropriado para métodos de desenvolvimento ágil, exigindo que o software seja inspecionado no final de cada pedido de proposta para garantir que ele é testado, seguro, acessível, documentado e implementado. Atender a este requisito requer a realização regular de demonstrações do software real em funcionamento, e não memorandos ou descrições do que o sistema deve fazer no futuro.

Historicamente, tem havido pressão para usar apenas contratos de preço fixo, na suposição de que isso reduz o risco. No entanto, se você puder medir constantemente a qualidade do software, um contrato de tempo e materiais ("time and materials") - com um teto máximo para o gasto total - permite mais flexibilidade para a equipa de desenvolvimento de software. Um contrato de tempo e materiais também permite cláusulas de rescisão muito mais fáceis se a direção do trabalho mudar ou a equipa do fornecedor não estiver a produzir software de qualidade. Se o trabalho da equipa de um fornecedor for inadequado ou se suas capacidades não forem as adequadas, nenhum trabalho adicional precisará de ser atribuído (efetivamente rescindindo o contrato) e o fornecedor poderá ser substituído.

LISTA DE VERIFICAÇÃO

- O projeto tem um dono do produto dedicado e autorizado que é um funcionário da instituição - não um contratado e não um funcionário da agência de TI do estado - cujo trabalho é priorizar o trabalho para a equipa de desenvolvimento;
- Um oficial de contratação do governo abraçou este projeto e está entusiasmado com novas formas de aquisição de software;
- O pedido de proposta será exclusivamente sobre a aquisição de serviços de desenvolvimento, não sobre a aquisição de algo tangível;

- O pedido de proposta exigirá uma equipa multifuncional de designers, investigadores e programadores;
- O pedido de proposta não terá mais de 20 páginas de texto;
- Foi criado e adicionado ao pedido de proposta um backlog de pelo menos uma dúzia de histórias de utilizador;
- Será utilizado um contrato de tempo e materiais (com um limite máximo);
- Será utilizado a forma de aquisição mais simples disponível que forneça acesso aos fornecedores-alvo.

QUESTÕES-CHAVE

- O dono do produto está autorizado a tomar decisões autorizadas rapidamente em nome da instituição?
- O dono do produto está preparado para passar a maioria das suas horas de trabalho cumprindo os requisitos dessa nova função?
- A liderança da instituição está preparada para que as decisões de produto sejam conduzidas pelas necessidades dos utilizadores identificadas, com base em conversas diretas com estes, em vez das preferências pessoais da liderança?
- O pedido de proposta estabelece requisitos claros sobre a entrega regular de código que funcione, documentação, teste e entrega da propriedade de todos os produtos de trabalho ao estado?

Cuidado com a armadilha do software comercial personalizado

O software comercial de genérico (SCG) e o Software as a Service (SaaS) podem ser ótimas maneiras de adquirir rapidamente um novo software ou

infraestrutura sem ter que criá-lo do zero. Por exemplo, faz todo o sentido comprar um processador de texto SCG em vez de construir seu próprio processador de texto personalizado.

Mas, para aquisições importantes de tecnologia especializada crítica, seja extremamente cauteloso com as alegações de que SCG ou SaaS funcionarão "fora da caixa". Os fornecedores costumam lançar seus "SCG personalizáveis" e SaaS como uma solução mágica, prometendo que respeitará os seus requisitos regulatórios e de processo específicos. E pode - mas provavelmente somente após extensas modificações.

Antes de contratar essas ferramentas, fale primeiro com outras entidades que usaram esses produtos personalizados. Muito provavelmente, você aprenderá que o que está sendo vendido como uma solução pronta a usar leva muito mais tempo e dinheiro para personalizar do que esperava.

Em vez de exigir qualquer solução na fase de orçamento, dê às agências o espaço para determinar se devem comprar ou construir várias partes do sistema. Se a alocação de orçamento exige SCG, então a instituição provavelmente terminará aprisionada numa versão altamente modificada de um produto SCG, barrada de todas as atualizações futuras por causa dessas modificações. Da mesma forma, a obrigatoriedade em usar software-como-serviço (SaaS) provavelmente forçará a agência a ajustar os seus requisitos como um sapato apertado, enquanto gasta uma quantia significativa de dinheiro adicional num "integrador de software" para ligá-lo ao sistema existente, levando ao mesmo tipo de aprisionamento indesejável.

Pode muito bem fazer sentido usar SCG ou SaaS como o núcleo de um novo sistema importante. Mas a instituição precisa de examinar isso com atenção, reconhecendo que não é provável que consigam uma solução SCG ou SaaS totalmente pronta para a utilização especializada que a instituição necessita.

LISTA DE VERIFICAÇÃO

- O orçamento não obriga ao uso de SCG, SaaS ou software personalizado, mas permite que a agência proponha uma combinação daqueles conforme achar necessário.
- As alegações dos fornecedores de que seu produto SCG ou SaaS funcionará imediatamente, sem modificação ou personalização onerosa, são investigadas de forma independente conversando com outros clientes que usaram esses produtos e passaram pelo processo de personalização e implementação.

QUESTÕES-CHAVE

- Uma vez que o produto foi personalizado para atender às necessidades da instituição, como serão as atualizações de software SCG feitas? Quanta personalização adicional será necessária para integrar essas modificações e quem vai pagar por essas atualizações?
- O que acontece se o fornecedor de SaaS fechar um dia sem avisar?
- O estado terá acesso fácil e gratuito aos seus dados, modelos de dados e APIs?

Requira demonstrações, não memorandos

Historicamente, o progresso em projetos de desenvolvimento de software era medido comparando o trabalho que foi feito com o cronograma de trabalho que foi estabelecido no início. Isso é feito produzindo artefatos como cronogramas e listas de tarefas concluídas. Mas isso não funciona - o desenvolvimento ágil de software tem isso como premissa. As equipas de desenvolvimento de software modernas nunca ouviram falar de "CMMI" ou "ISO-9000" e não licitarão trabalhos que incluam esses requisitos.

Uma filosofia melhor é *demos, não memorandos.* Em vez de medir o progresso olhando para artefatos feitos para o propósito, olhe para o trabalho real que está sendo feito. Junte-se às revisões que são realizadas no final de cada pedido de proposta, onde o trabalho realizado naquele sprint é demonstrado para a equipa do projeto e os utilizadores finais convidados. Experimente o site. Instale a aplicação.

Uma parte importante para garantir que o progresso não seja ilusório é que o contrato inclua um Plano de Vigilância de Garantia de Qualidade (PVGQ) que exige, no final de cada pedido de proposta, que todo o trabalho respeite padrões específicos. O PVGQ descreve o método pelo qual o governo determinará se o trabalho do fornecedor é de qualidade suficiente para ser aceite no final de cada pedido de proposta, permitindo que o fornecedor execute os mesmos testes para garantir que não haverá surpresas.

O PVGC não exige a produção de nenhum artefato explicitamente com o propósito de monitorar o trabalho - a maneira de monitorar o trabalho é *ver se ele realmente funciona.* Esta é uma maneira muito diferente de acompanhar o progresso de um projeto de tecnologia. Ela tem a vantagem adicional de ser um teste funcional mais objetivo, observável do que exigir interpretações subjetivas ou legais sobre se o trabalho satisfaz uma longa lista de requisitos.

LISTA DE VERIFICAÇÃO

- Um funcionário público autorizado e dedicado servirá como o dono do produto;
- Não haverá requisitos de planeamento ou relatório que vão contra a metodologia ágil (ou seja, não há datas para as tarefas específicas serem concluídas e nenhuma especificação de funcionalidade exata que será exigida - seja no pedido de proposta, no plano de aquisição ou legislação);

- Haverá um desenvolvedor de software empregado pelo governo que garantirá a conformidade com o PVGQ no final de cada pedido de proposta;
- As pessoas que supervisionam, acima do nível do dono do produto governamental, estão dispostas a receber principalmente "relatórios" na forma de demonstrações de software em funcionamento, combinados com uma revisão das histórias de utilizadores que foram concluídas e aquelas que permanecem por completar;
- Há uma pessoa identificada dentro da instituição que está preparada para fornecer explicações de progresso para cada nível de chefia, porque, geralmente, artefatos de medição de progresso num projeto ágil não são familiares para pessoas que estão acostumadas a projetos em cascata.

QUESTÕES-CHAVE

- É viável que a instituição forneça um apoio integral para uma abordagem tão radicalmente diferente de medir o progresso?
- Existe alguém com o poder de fincar os pés e exigir um cronograma, inviabilizando assim a metodologia ágil?

Contrate talento interno

Se ninguém na comissão de orçamentação (???) tiver experiência com desenvolvimento de software, então não estão bem equipados para considerar uma solicitação de financiamento de desenvolvimento de software. O mesmo é verdade para instituições - se ninguém na liderança de projetos tiver experiência com desenvolvimento de software, então a instituição não está bem equipada para liderar um projeto de desenvolvimento de software com sucesso. A responsabilidade recai sobre o governo, legisladores e chefias para garantir

que suas respectivas organizações priorizem a contratação de pessoas com essa experiência.

Embora possa ser tentador resolver essa lacuna de conhecimento contando com alguém do departamento de TI central do estado ou com um fornecedor, em última análise, as instituições devem ter o conhecimento interno suficiente para compreender o que precisam, o que deveriam exigir dos fornecedores e conseguir avaliar o trabalho realizado.

Para determinar se a sua comissão de orçamentação ou a sua liderança tem experiência para considerar solicitações de software ou liderar projetos de software, faça perguntas. Todas as instituições, exceto as menores, terão equipas técnica que pode se juntar à liderança do projeto, embora poucos departamentos empreguem atualmente profissionais com experiência em desenvolvimento de software.

Se você não tem o conhecimento de que precisa internamente, precisará contratar alguém que tenha - mesmo que apenas temporariamente. Um programador ou designer com experiência na criação de software moderno, idealmente para o governo, é sua melhor aposta. Além disso, considere autorizar um ou mais funcionários a passar parte de seu tempo de formação aprendendo os fundamentos do desenvolvimento ágil de software - existem "bootcamps" para o efeito, incluindo algumas opções online.

O custo de contratar um programador ou aprimorar as habilidades dos seus funcionários atuais é pequeno em comparação aos gastos com tecnologia. E uma vez que um funcionário tenha acompanhado um projeto ágil do início ao fim, ele estará melhor equipado para considerar futuras solicitações de orçamento para software personalizado.

Da mesma forma, as instituições devem empregar diretamente programadores suficientes para que possam supervisionar o trabalho que está sendo feito pelos

fornecedores. Eles representarão a instituição, garantindo que o trabalho dos fornecedores seja de alta qualidade e que os fornecedores estejam trabalhando nas coisas certas.

Embora o software nunca esteja "pronto" - você sempre precisará se adaptar às mudanças de tecnologia, política, regulamentos, leis e necessidades do utilizador - chegará um ponto em que você precisará de uma equipa muito menor para continuar esse trabalho. Nessa altura torna-se especialmente importante ter vários funcionários da agência que entendam totalmente o software e que sejam capazes de mantê-lo.

Para projetos maiores, você precisará de contratar uma equipa de desenvolvimento a termo indefinido, sob a supervisão de um dono de produto do governo. Nos projectos em cascata, isso chama-se "Operações e Manutenção", mas sob a metodologia ágil, O&M é simplesmente o processo comum de investigação de utilizadores, design, desenvolvimento de software contínuos, etc.

LISTA DE VERIFICAÇÃO

- Há um ou mais funcionários de orçamento (???) com experiência no desenvolvimento de software complexo e personalizado usando metodologias ágeis que ajudará na avaliação de solicitações de orçamento de software personalizado;
- Se não houver funcionários de orçamento (???) com experiência relevante, a legislatura (???) tem um contrato com um fornecedor sem conflito de interesses - um sem outros contratos com o estado e sem vínculos ou parcerias com quaisquer produtos SCG;
- A instituição identificou um funcionário do governo específico que fornecerá liderança técnica para o projeto, juntamente com provas de sua

experiência no desenvolvimento de software personalizado utilizando metodologias ágeis.

QUESTÕES-CHAVE

- Quando um fornecedor entrega o código no final de cada pedido de proposta, que funcionário *específico* do governo inspeciona o código para garantir a qualidade?
- Se uma agência apresenta um custo para concluir um projeto de software específico, que funcionário de orçamento (???) está equipado para saber se esse é um preço adequado? Qual funcionário do comité de orçamento legislativo (???) *específico* está equipado para saber se esse é um preço adequado?
- Quando a aquisição for concluída, quem fará a manutenção do software? A instituição emprega pessoas que sabem como mantê-la? Eles serão incluídos no processo de desenvolvimento para que possam aprender sobre como ele é construído e ajudar a garantir que seja algo que eles sejam capazes de suportar?

Minimize o custo de mudança

O governo existirá por muito mais tempo do que qualquer software. E isso significa que, um dia, o seu novo e empolgante sistema de informação poderá tornar-se no seu antigo sistema de informação difícil de usar.

Por melhor que seja o software hoje, eventualmente você precisará de mudar para um novo sistema - seja no todo ou em parte. E adquirir software como um monólito completo garante que ele gradualmente se tornará incapaz de atender às necessidades de uma instituição.

Mudanças tecnológicas, mudanças nas políticas governamentais, mudanças nos regulamentos, mudanças nas leis, mudanças nos requisitos de subsídios comunitários e mudanças nas prioridades da liderança - qualquer projeto que seja planeado em grande detalhe no início não será capaz de se adaptar a essas mudanças e correrá um risco significativo de falha, ultrapassagens significativas nos custos e prazos ou "pedidos de alteração" caros.

Portanto, em vez de adquirir uma peça gigante de software proprietário, insista em que os seus fornecedores adotem práticas como o uso de software de código aberto e arquitetura orientada a serviços (SOA). Dessa forma, você pode otimizar para reduzir o custo de atualização e alteração do sistema desde o início.

LISTA DE VERIFICAÇÃO

- Os sistemas, sejam nativos da **cloud** ou movidos para a **cloud**, usarão arquitetura orientada a serviços (SOA) que é independente de fornecedor e produto;
- Para garantir a portabilidade dos dados, os arquivos serão armazenados em formatos abertos não proprietários, suportados por vários fornecedores;
- As APIs usarão esquemas abertos;
- Para evitar o aprisionamento do produto, será usado software de código aberto em vez de software comercial sempre que possível;
- O governo será o proprietário de todos os produtos do trabalho do fornecedor;
- Se estiver usando componentes SCG, o fornecedor delineará um caminho de saída para um concorrente - tanto contratual quanto tecnologicamente - com uma maneira económica de exportar todos os dados armazenados.

QUESTÕES-CHAVE

- Qual é o plano para reduzir o tempo e o custo de futuras atualizações do sistema devido a mudanças de tecnologia, lei, política ou fornecedor?
- Quanto custará para mudar o sistema para refletir a tecnologia necessária ou as mudanças de política?
- As APIs estão abertas e podem ser usadas por outros fornecedores?
- Os formatos de dados são padronizados, abertos e utilizáveis por outros fornecedores?
- Manter um sistema de software atualizado exigirá um trabalho regular e contínuo - qual é o plano para conseguir isso?

Avalie o sucesso com base em resultados iterativos, não em marcos do projeto

O valor não deve vir no final de um projeto - deve ser fornecido aos utilizadores finais no prazo máximo de seis meses após a adjudicação do contrato, e constantemente a partir daí. No final do *primeiro* sprint, o código funcional deve ser entregue à agência para sua revisão, e isso deve repetir-se com cada pedido de proposta subsequente. Os utilizadores finais devem avaliar o trabalho no final de cada pedido de proposta, independentemente de o trabalho já ter sido implementado para uso diário.

Não meça o progresso em "pontos da histórias", linhas de código escritas, horas de trabalho por pessoa, etc. A única medida de sucesso que importa é o valor entregue aos utilizadores finais. Isso é melhor avaliado participando nas revisões de sprint bi-semanais e conversando com o dono do produto.

LISTA DE VERIFICAÇÃO

- A equipa do fornecedor usará metodologias ágeis;
- O fornecedor será obrigado a colocar software funcional num ambiente da propriedade do governo no final de cada pedido de proposta;
- Rotineiramente, a equipa do projeto irá entrevistar e testar o seu trabalho com utilizadores finais, tanto para informar o trabalho planeado como para determinar se o trabalho já realizado está correto;
- O pedido de proposta não fará menção a um cronograma detalhado do projeto, e não haverá menção a cronogramas ou contratos de validação e verificação independente;
- Um membro da equipa legislativa será designado para fornecer supervisão do projeto e coordenará com a liderança do projeto para monitorizar o progresso, comparecendo periodicamente às revisões de cada pedido de proposta;

QUESTÕES-CHAVE

- A agência solicitante pode providenciar valor aos utilizadores finais em seis meses? Qual é, especificamente, esse valor?
- A agência está preparada para o fornecedor entrevistar e testar continuamente o seu trabalho com utilizadores finais reais - talvez incluindo funcionários da agência?

Limitar gastos totais

Quanto maior a quantia de dinheiro gasta em um projeto de software, maiores as probabilidades de falha.

LISTA DE CONTROLE

- A agência solicitante entende que não está recebendo uma pequena percentagem dos recursos de que acredita precisar - em vez disso, está recebendo um processo inteiramente novo para adquirir software, bem como financiamento adequado sob esse modelo.

Limitar tamanhos de contrato

Usar um único fornecedor por um longo período de tempo ou para um grande número de equipas pode ser mais confortável, mas inevitavelmente leva ao aprisionamento ao fornecedor. Dividir os projetos em vários pequenos contratos incentiva os fornecedores a construir um ecossistema de software sustentável, em vez de um monólito, e torna cada contrato pequeno o suficiente para que as probabilidades de sucesso aumentem significativamente.

Deverá exigir que não sejam gastos mais do que um valor máximo num único contrato anualmente, e que nenhum contrato dure mais do que um ou dois anos, incluindo períodos de opção. Dessa forma, você não terá mais do que duas equipas de desenvolvimento de um único fornecedor. Se o projeto precisar de mais equipas de desenvolvimento, obtenha-as de outro fornecedor e faça com que trabalhem separadamente. Limite o pedido de proposta também, mantendo-a abaixo de 20 páginas; não gaste mais do que 60 dias a escrevê-lo.

Além de evitar o aprisionamento, há outro benefício em usar contratos menores: eles têm menos probabilidade de serem contestados, porque o valor não justifica o problema e os custos legais. Se você é respeitoso e transparente com os fornecedores e não exige centenas de páginas de propostas, eles provavelmente vão querer trabalhar para si no futuro.

À medida que o número de pessoas que trabalham num projeto aumenta, também aumenta a quantidade de tempo que todas essas pessoas gastam coordenando umas com as outras. A solução para isso é fazer com que trabalhem em paralelo, o que é possível ao construir com peças desacopladas.

Ter mais do que uma equipa de fornecedores trabalhando no seu projeto também fornece opções mais competitivas no caso de precisar de mudar de fornecedor.

LISTA DE CONTROLE

- Se o projeto vai exigir vários contratos, o âmbito do primeiro contrato foi identificado e há uma ideia geral da composição de outros futuros contratos;
- Se houver mais de uma equipa de desenvolvimento, será empregada uma arquitetura orientada a serviços (SOA);
- Quando possível, os contratos serão dimensionados dentro do limite de aquisição simplificado para que possam ser concedidos de forma rápida e fácil;
- O primeiro projeto identificado tem complexidade técnica relativamente baixa, baixo risco político e alto valor para o utilizador final, de modo a que as equipas possam começar a trabalhar desta forma enquanto experimentam e aprendem num ambiente de risco relativamente baixo.

QUESTÕES-CHAVE

- Os responsáveis contratantes relevantes leram este manual?
- Os oficiais contratantes entendem que não estão sendo solicitados a fazer todo o trabalho que vai para um contrato gigante? Eles entendem que contratos mais pequenos são muito mais fáceis de conceder e que, com as metodologias ágeis, eles também serão muito mais fáceis de gerir?

Financie sistemas, não monólitos

Não substitua o antigo sistema desatualizado por um novo sistema desatualizado. Insista em sistemas desacoplados ou independentes que são construídos de forma incremental. Dessa forma, eles nunca precisarão ser completamente substituídos - será apenas necessário substituir componentes individuais conforme a necessidade.

LISTA DE VERIFICAÇÃO

- Cada contrato será escrito para entregar valor aos utilizadores finais - eles não servirão para "manter servidores" ou "configurar uma base de dados", mas para "criar um website de solicitação de autorizações" ou "simplificar o processo de admissão", etc.;
- Não haverá lugar para um "arquiteto do sistema", porque a arquitetura surgirá iterativamente ao longo do processo ágil;
- Se o projeto for grande o suficiente para ter várias equipas a trabalhar simultaneamente, não há a expectativa de que todos os membros de todas as equipas estejam em reuniões juntos;
- O pedido de proposta especificará o uso de arquitetura orientada a serviços para cada componente.

QUESTÕES-CHAVE

- Existe um único ponto de falha que pode derrubar todo o sistema? (Em caso afirmativo, provavelmente trata-se de um monólito, não de um sistema.)
- Se o contrato de um fornecedor precisar de ser rescindido por fraco desempenho, os outros fornecedores conseguirão continuar a trabalhar sem interrupção?

Expanda o seu grupo de fornecedores

É pouco provável que seus atuais fornecedores utilizem as práticas de desenvolvimento de software modernas descritas neste manual - afinal eles foram contratados pelas suas práticas tradicionais. Para encontrar fornecedores que satisfaçam as suas novas necessidades, você provavelmente precisará de identificar e atrair novas empresas que usem práticas modernas de desenvolvimento de software.

Se for importante obter propostas de fornecedores locais, saiba que há uma alta probabilidade de que haja algumas empresas pequenas qualificadas que podem fornecer programas de desenvolvimento ágil. No entanto, se você deseja reduzir o preço dos serviços, é importante considerar trabalhar com equipas de fornecedores remotos ou distribuídos (em vez de no local).

Embora a equipa de compras fique tentada a procurar fornecedores que já construíram um sistema quase idêntico, isso é desnecessário e limita o conjunto de fornecedores a apenas algumas grandes empresas. Em vez disso, eles devem ampliar seu âmbito e procurar fornecedores que tenham construido algo análogo. Um fornecedor que criou um site para reservar carros de aluguer pode criar um site para solicitar licenças de acampamento em áreas remotas. Um desenvolvedor líder que construiu uma base-de-dados para rastrear as posições dos cometas pode construir uma base-de-dados para rastrear veículos. Ao procurar conhecimento relevante desta forma, a equipa de compras certamente encontrará muitos fornecedores que podem fazer o trabalho.

LISTA DE CONTROLE

- O pedido de proposta será simplificada (não mais do que 20 páginas) e compreensível por desenvolvedores de software que normalmente não trabalham com o governo;

- O plano de aquisição inclui chegar a pequenos fornecedores para incentivá-los a licitar;
- O pedido de proposta não ficará oculto por detrás de um website de compras que exige registo, mas será publicada abertamente na web para que a comunidade de fornecedores possa partilhá-la;
- O pedido de proposta exigirá que os fornecedores identifiquem o seu pessoal-chave nas suas propostas - não mais do que três pessoas - como o programador líder ou o designer líder;
- O plano de aquisição inclui entrevistar os finalistas sobre a abordagem proposta, questionar o pessoal-chave identificado, *não* a equipa de vendas do fornecedor;
- Os funcionários dos fornecedores não serão obrigados a trabalhar no local numa instalação do governo;
- As equipas de fornecedores e o dono do produto governamental terão permissão para usar um serviço de videochamada, uma ferramenta de chat em tempo real e um sistema público de controle de versões baseado em Git e outras ferramentas comumente usadas pelas equipas de desenvolvimento ágil.

QUESTÕES-CHAVE

- Há algum benefício - político ou outro - em conceder contratos a fornecedores locais, ou mesmo requisitos para fazê-lo? Isto pode limitar o grau em que você pode expandir seu grupo de fornecedores?
- Economizar muito dinheiro por cada equipa é razão suficiente para superar quaisquer objeções a ter equipas remotas?
- Foi feita uma investigação de mercado para saber quais fornecedores serão alvos do pedido de proposta, em vez de apenas emitir um pedido de

informação e esperar que resulte?

Partilhe o seu software

O software de uma instituição provavelmente será útil, no todo ou em parte, para outras instituições do país ou comunidade.

Se o software for publicado abertamente, os funcionários dos fornecedores ficarão ansiosos para trabalhar nele - torna-se um caso interessante de trabalho que eles podem adicionar ao seu currículo para empregos futuros ou partilhar com amigos, o que ajuda a garantir que você receba os seus melhores esforços. Além disso, os futuros pedidos de proposta adicionais emitidos para o projeto podem apontar os fornecedores para o código que já foi escrito, permitindo-lhes ver exatamente no que trabalharão ou com o que interagirão. O departamento do governo que financia o trabalho pode estar interessado em encontrar maneiras de partilhar o seu software com outras instituições que eles também estejam a financiar.

LISTA DE VERIFICAÇÃO

- O pedido de proposta exigirá que o código-fonte do software seja escrito e mantido em público numa plataforma de codificação social (por exemplo, GitHub ou GitLab), desde o primeiro dia.
- O pedido de proposta exigirá que o software seja explicitamente dedicado ao domínio público ou publicado sob uma licença de código aberto
- O pedido de proposta usará as melhores práticas de segurança, exigindo que o software seja estritamente separado dos dados e segredos, com testes automatizados para garantir que a separação seja mantida;

- O pedido de proposta exigirá que o software seja documentado suficientemente bem para que um desenvolvedor sem conhecimento prévio do projeto possa usá-lo para executar sua própria cópia do software.

QUESTÕES-CHAVE

- Os responsáveis pela ciber-segurança do estado ou da instituição ficarão preocupados com a perspectiva de publicar software de código aberto e assim bloquear a implantação do software?
- Existem outras agências no estado que possam beneficiar deste software? Eles podem ser consultados antes e durante o processo de desenvolvimento?
- O escritório de consultoria jurídica da agência (ou seu equivalente) fará objeções à publicação de software no domínio público ou sob uma licença de código aberto aprovada pela OSI?

Orçamento para software como despesa operacional

Ao contrário de pontes ou outros projetos de infraestrutura física, o software personalizado nunca está "pronto", por isso é importante planear para que ele seja modificado continuamente. Dessa forma, ele pode atender às necessidades de hoje, não de ontem.

Para sistemas pequenos, isso pode exigir a adição de um ou menos colaboradores a tempo inteiro à equipa de programadores da agência. Para sistemas grandes e importantes, isso pode exigir a aquisição de uma equipa para desenvolver e manter o software continuamente.

A manutenção de software às vezes é orçamentada como se fosse uma atividade diferente da construção inicial de software, mas isso é um erro. Manter o software deve significar simplesmente continuar a modificá-lo em resposta às

necessidades do utilizador identificadas, que mudam continuamente junto com as leis, regulamentos, políticas, melhores práticas e tecnologia. Isso requer os mesmos conjuntos de habilidades, metodologia e tarefas como construir um sistema de raíz. Uma proposta para a transição do desenvolvimento de software para uma fase de "operações e manutenção" ("O&M") deve ser vista como um mau sinal.

Em última análise, isso pode fornecer às agências uma fonte previsível de financiamento para projetos de software - substituindo despesas de capital imprevisíveis - ao mesmo tempo que fornece ao poder legislativo um custo anual previsível para todos os projetos de software da agência.

LISTA DE VERIFICAÇÃO

- A agência reconhece que o software deve ser melhorado continuamente enquanto estiver em uso, porque "manutenção" é funcionalmente o mesmo que construir software desde o início;
- A agência planeia adquirir serviços de desenvolvimento ágil;
- Você conversou com a agência solicitante para determinar se ela prefere receber financiamento ao longo dos anos, como um fluxo previsível de financiamento operacional, em vez de um montante fixo;
- Esta abordagem foi coordenada com todos os departamentos e agências envolvidos - esta é provavelmente uma mudança radical que exigirá confiança e cooperação entre todas as partes;
- Se a solicitação de uma agência apresentar um alto risco de falha, você alocará uma fracção do valor no primeiro ano, aumentando o financiamento conforme o projeto apresenta valor.

QUESTÕES-CHAVE

- O financiamento solicitado será gasto dentro de um único período do orçamento?
- Que valor poderá ser entregue aos utilizadores finais por uma fracçaõ do valor solicitado?
- Os responsáveis pelo financiamento do trabalho estão abertos a uma abordagem operacional para o financiamento?

Faça perguntas técnicas às agências

As solicitações de orçamento para software personalizado geralmente apresentam pessoas não técnicas fazendo propostas técnicas para outras pessoas não técnicas. Este processo não se presta a fazer perguntas-chave, como muitas das encontradas ao longo deste manual. É importante fazer todas essas perguntas técnicas difíceis e insistir em obter as respostas certas (consulte o Apêndice A para exemplos de perguntas e respostas).

Não é gentileza financiar um projeto que vai fracassar. Se a agência não sabe exatamente o que quer comprar, não vai conseguir obtê-lo.

QUESTÕES-CHAVE

- O que é que a agência quer comprar exatamente? Por quê? Quem irá beneficiar?
- Quais partes do sistema serão personalizadas? Qual será o SCG real (não personalizado)? Quanto custarão essas atualizações? O que será feito quando um componente comercial for terminado - por exemplo, se a empresa de base-de-dados sair do mercado?
- Quem são os utilizadores finais do seu sistema? Você falou com eles? O que é que *eles* querem?

- Você está preparado para quando as mudanças precisam ser feitas?
- Quanto custará para mudar para um novo sistema?
- O que você está fazendo para evitar o pagamento de custos de alteração caros no futuro?

Apêndice A: Perguntas a serem feitas

Quando estiver a considerar uma solicitação de orçamento para um projeto de software personalizado, poderá ser difícil consultar este manual para encontrar as perguntas certas a serem feitas. Aqui estão algumas das perguntas abertas básicas que você pode fazer para determinar se um projeto está configurado para o sucesso.

Quais são os objetivos do projeto? Quais resultados são priorizados?

Resposta errada: qualquer coisa de natureza técnica, em vez de melhorar a experiência do utilizador.

Resposta certa: uma ou mais necessidades específicas do utilizador são nomeadas.

Qual é a necessidade do utilizador que este projeto irá satisfazer?

Resposta errada: qualquer coisa que não identifique as necessidades claras dos utilizadores finais identificadas por meio da investigação junto a utilizadores.

Resposta certa: a agência determinou necessidades específicas com base em entrevistas com utilizadores finais e pode citar várias dessas necessidades especificamente.

Se o fornecedor selecionado não tiver um desempenho adequado, quão difícil será rescindir o contrato? Quanto tempo vai demorar para substituí-los por outro fornecedor? Quanto você acha que vai custar?

Resposta errada: "Estaríamos muito relutantes em rescindir o contrato. Levaria meses ou anos para substituí-los por um novo fornecedor. Um tempo significativo da equipa seria necessário para fazer isso, e nosso projeto atrasaria muitos meses. Uma vez tendo um sistema, teremos que começar tudo de novo se decidirmos mudar de fornecedor. "

Resposta certa: "Será um contrato de tempo e materiais, então podemos parar de atribuir trabalho ao fornecedor a qualquer momento, e isso seria o fim funcional do contrato. Poderíamos reeditar a RFP e ter um novo fornecedor integrado em seis semanas. Exigiria uma pequena quantidade de tempo da equipa e atrasaria o projeto apenas nessas seis semanas. "

O pedido de proposta incluirá requisitos de como o sistema irá operar? Em caso afirmativo, quantos requisitos estão incluídos?

Resposta errada: "Passamos o ano passado revendo os nossos requisitos de negócios e escrevemos centenas de requisitos para incluir no pedido de proposta, para garantir que obteremos exatamente o que precisamos."

Resposta certa: "Estamos mais focados nos resultados que desejamos do novo sistema. Desenvolvemos uma acumulação de histórias de utilizadores para ajudar a orientar o trabalho da equipa, em vez de produzir uma lista detalhada de requisitos técnicos."

Quanto tempo você espera que o pedido de proposta demore?

Resposta errada: "Desenvolvemos várias centenas de páginas de requisitos de sistema junto com mais 50 páginas de termos e condições padrão."

Resposta certa: "Menos de 20 páginas, e esperamos manter isso abaixo do limite de compras simplificado do estado, para tornar mais fácil, mais barato e rápido para novos fornecedores licitarem no projeto."

Você prevê a emissão de um contrato de preço fixo ou de tempo e materiais?

Resposta errada: "Preço fixo, porque é a melhor maneira de controlar os custos do fornecedor."

Resposta certa: "Tempo e materiais, porque é a melhor maneira de manter a flexibilidade de que precisamos para responder às necessidades do utilizador, lidar com desafios técnicos imprevistos e garantir que, se os fornecedores que não estiverem a entregar o que precisamos, possam ser alterados sem colocar o projeto em risco."

Qual valor será entregue aos utilizadores em seis meses?

Resposta errada: "Nenhum - ainda não estará pronto. Pretendemos mostrá-lo aos utilizadores quando tudo estiver concluído."

Resposta certa: exemplos específicos são mencionados.

Quem será o dono do produto?

Resposta errada: "O que é um 'dono do produto'?"

Resposta certa: uma pessoa específica é nomeada ou ela está treinando uma equipa interna para assumir essa função.

Qual processo de desenvolvimento de software será usado?

Resposta errada: "Queda-de-água (Waterfall)" ou qualquer resposta que indique falta de compreensão.

Resposta certa: "Agile", "Extreme Programming" (XP) ou "Scrum" são respostas aceitáveis.

Na equipa que preparou esta solicitação, quem tem experiência no desenvolvimento de software?

Resposta errada: "Ninguém".

Resposta certa: uma pessoa específica é nomeada.

Com que frequência o trabalho será colocado em produção?

Resposta errada: "Quando estiver pronto."

Resposta certa: "No final de cada pedido de proposta."

O projeto irá automatizar os testes? Integração? Entrega? Testes de segurança?

Resposta errada: "Estamos investigando isso."

Resposta certa: "Sim, desde o primeiro dia."

Quanto custarão os pedidos de alteração?

Resposta errada: Qualquer resposta que preveja pedidos de alteração de qualquer tipo.

Resposta certa: "Esperamos que o sistema mude constantemente em resposta às novas necessidades dos utilizadores, novas tecnologias e novas políticas. É por isso que estamos a propôr um contrato de tempo e materiais e uma abordagem de desenvolvimento ágil para reduzir o custo de resposta a essas mudanças. "

Como você saberá se o projeto está no caminho certo e se os empreiteiros estão cumprindo o prometido?

Resposta errada: "Estamos contratando um especialista independente de verificação e validação para nos fornecer relatórios mensais sobre o estado do

projeto."

Resposta certa: "Os fornecedores fornecerão demonstrações frequentes de software funcional que refletem nossas prioridades, atendem aos padrões técnicos do QASP e fornecem valor aos utilizadores finais. Se esses padrões não forem atendidos, o valor para os utilizadores finais não será mostrado em seis meses, o contrato com eles será cancelado. "

Quem será o proprietário do software?

Resposta errada: "O vendedor".

Resposta certa: "O estado" ou "será entregue ao domínio público".