Porque usar arquiteturas abertas é importante no BI

Arthur Jalmusny
arthur@yellowfin.com.br
(51) 99985-6883

Ago/2024

A importância das arquiteturas abertas

As escolhas de uma arquitetura podem fornecer alguma proteção contra a tomada de decisões erradas. É muito importante identificar a arquitetura e entender o que está bloqueado e o que está aberto.

Neste white paper investigamos a importância das arquiteturas abertas ao adotar uma plataforma de Business Intelligence para flexibilidade de longo prazo visando ajudá-lo a uma tomada de decisão mais eficaz e eficiente.

Arquitetura aberta e código aberto

Arquitetura Aberta (Open Architecture) e Código Aberto (Open Source) não significam a mesma coisa. Software de código aberto significa que seu código-fonte é aberto. Por outro lado, arquitetura aberta é o desenvolvimento e integração de diferentes empresas em um mesmo produto.

Arquitetura aberta pode ser definida como uma arquitetura de computador ou software que foi desenvolvida usando padrões abertos e torna mais fácil adicionar, alterar e substituir novos componentes. Esta abertura se refere ao uso de padrões publicados e disponíveis gratuitamente nas interfaces de hardware e software. O padrão aberto permite o desenvolvimento por outras empresas usando o mesmo padrão.

O que é arquitetura aberta?

Novas soluções surgem, algumas amadurecem e se tornam populares, e outras desaparecem. Os tomadores de decisão, por sua vez, têm a tarefa de escolher novas soluções de tecnologia que podem ter várias implicações de longo prazo para as organizações para as quais trabalham. As quantidades de escolhas disponíveis são enormes.

1. Com quais fornecedores de tecnologia devo fazer parceria?
2. Quais padrões de arquitetura devem ser adotados?
3. Quais plataformas são padronizadas?

Nos negócios esses tipos de decisões são importantes e precisam ser feitos baseados no conhecimento e plena consciência de que a base dessas decisões mudará em um futuro próximo. Mas tomar a decisão errada sobre qual tecnologia adotar poderá afetar os custos, a experiência do cliente, a competitividade e, por fim, o sucesso de uma organização.

Não há garantia de que as decisões certas sempre serão tomadas, ou de fato que uma decisão certa hoje (com base em todas as evidências disponíveis) não será a decisão errada amanhã. Portanto, a flexibilidade e a capacidade de responder rapidamente às mudanças quando necessário, fazendo novas escolhas, deve sempre ser uma parte importante de qualquer decisão sobre a arquitetura para evitar o bloqueio de uma solução

Por que o aprisionamento tecnológico é ruim?

O aprisionamento tecnológico consiste na criação de uma dependência do consumidor em relação ao fornecedor. Essa dependência costuma vir acompanhada de altos custos, relacionados geralmente a atualizações e melhorias. Muitas vezes a escolha de um fornecedor de determinado produto ou serviço pode parecer atrativa, à primeira vista. Entretanto, à medida que se usa esse produto ou serviço, vai se criando uma dependência em relação ao fornecedor, até que se chega em um ponto em que os custos de migração para outra tecnologia ou outro fornecedor tornam-se altos demais. O aprisionamento ocorre quando o custo ou esforço de se afastar de uma escolha específica (plataforma, fornecedor) supera o benefício, mesmo se essa escolha for boa para os negócios.

Muitas empresas já consideraram ou experimentaram o desejo de aproveitar as vantagens de uma plataforma mais barata, software com mais recursos ou de terminar o contrato com um fornecedor que não está entregando. Mas a dificuldade de mudar é simplesmente grande demais para pensar em fazê-lo. Esse é o estado de se ficar preso em uma tecnologia.

Ninguém quer ficar preso a uma solução, a menos que essa solução forneça um valor atraente em uma base contínua, com a Apple sendo um exemplo popular. Imagine um dia no futuro em que a Apple fique para trás. Como será difícil desligar a vida do ecossistema da Apple?

As apostas ficam significativamente mais altas quando se trata de decisões de tecnologia organizacional. Em particular, o aprisionamento pode ser problemático ao escolher novas plataformas de software. Alguns podem argumentar que é do interesse do fornecedor prendê-lo, principalmente na era dos modelos de software como serviço (SaaS) e de assinatura, em que o uso do software exige pagamentos contínuos. Quanto mais difícil mudar a tecnologia, maior será a probabilidade de esses pagamentos continuarem no futuro.

O aprisionamento deve ser diferenciado de aderência. A aderência ocorre onde os clientes desejam continuar usando o software, não quando são forçados a isso. Os recursos e a usabilidade da plataforma, os comerciais e os níveis de serviço são tais que o cliente experimenta um valor contínuo e permanece de boa vontade com a plataforma, embora não se sinta limitado.

Onde o aprisionamento pode aparecer no software?

O aprisionamento pode se manifestar de muitas formas diferentes. Não ocorre apenas quando se deseja mudar um software específico, mas também quando se deseja mudar o ambiente em que esse software está operando.

Provedores de nuvem

Aprisionamento de soluções de BI que fazem parte do software de propriedade de um dos principais fornecedores de computação em nuvem (Amazon Web Services, Microsoft Azure, Google Cloud) e só funcionam, ou funcionam perfeitamente, em sua nuvem.

Sistemas operacionais

Aprisionamento de soluções de BI desenvolvidas para uma plataforma de sistema operacional ou ambiente fechado, como apenas no Windows sem suporte para Linux, ou diferentes versões do software em diferentes sistemas operacionais.

Recursos proprietários e conjuntos de habilidades

Aprisionamento de soluções de BI com plataformas de software que apresentam scripts e linguagens proprietárias que basicamente se tornam habilidades descartáveis. Todo software requer algum grau de familiarização ou treinamento, dependendo de sua complexidade e, inevitavelmente, mudar de uma solução altamente proprietária resultará em um esforço de treinamento extenso (e caro).

Softwares que tentam fazer tudo

Aprisionamento de soluções de BI com soluções que fornecem ampla funcionalidade que pode ir além de sua finalidade inicial podem ser mais difíceis de remover ou substituir.

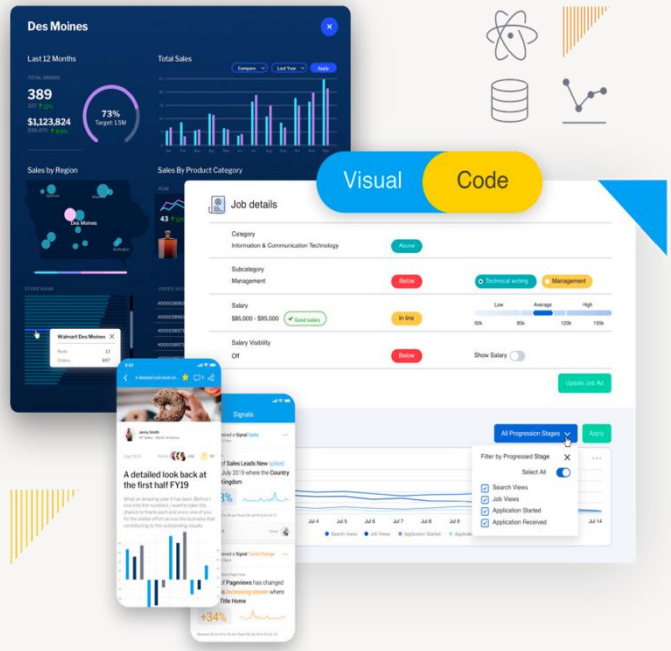
Como fazer escolhas mais abertas na adoção de tecnologia?

Em primeiro lugar, aprenda onde pode ocorrer o 'lock-in' em sua organização. Faça perguntas durante todo o processo, esteja informado sobre as escolhas que você fizer ou trocas de tecnologias envolvidas.

Considere não apenas o software que está comprando, mas também outros elementos de seu ambiente.

Exemplos:

- E se você mudar do local para a nuvem
- E se você mudar de provedor de nuvem
- E se você decidir mover toda a sua carga de trabalho de computação do Windows para Linux ou vice-versa
- E se você mudar os provedores de banco de dados analíticos



Mantenha um nível saudável de dúvida todo o processo de compra. Podem ser anos de distância, mas pense nas consequências de uma decisão diferente para o futuro.

Exemplos de BI com bloqueio de Tecnologia:

Armazenamento de dados proprietário

Por muitos anos, cubos de dados ou armazenamento de banco de dados em memória eram um recurso das plataformas de inteligência de negócios. A ingestão de dados nas camadas de armazenamento de dados proprietárias dessas ferramentas frequentemente fornecia um aumento de desempenho para os processos de descoberta de dados, especialmente quando cópias pessoais de dados podiam ser extraídas de armazenamentos de dados corporativos. Essa conveniência, no entanto, teve um custo: os dados proliferaram de forma não governada entre as organizações, mas o mais importante é que quanto maior o investimento nessas camadas de dados, mais difícil se torna a troca. Os dados armazenados nesses tipos de camadas proprietárias geralmente são acessíveis apenas a partir das ferramentas fornecidas pelo provedor da plataforma de BI. As organizações hoje não têm mais opções com bancos de dados analíticos rápidos e a necessidade de armazenar dados em formatos proprietários em uma plataforma de BI não é mais necessária.

Exemplos: **Tableau Hyper, Qlik Associated Engine, MicroStrategy Cubes, Sisense Elasticubes**

Provedores de plataforma na nuvem

Os mega-provedores de nuvem, Amazon Web Services (AWS), Google Cloud Platform (GCP) e Microsoft Azure estão em uma guerra para ganhar o maior número de organizações possível. Isso não significa ser capaz de executar qualquer tipo de carga de trabalho de computação, mas também oferecer ofertas de software atraentes com descontos que são exclusivas ou funcionam de forma otimizada em sua plataforma. Esses fornecedores vêm desenvolvendo suas próprias ofertas ou acumulando recursos por meio de aquisições. O desafio surge quando as organizações desejam transferir a carga de trabalho de computação para outro provedor para obter melhor preço ou desempenho. As escolhas de software muitas vezes os prendem a um provedor de nuvem específico, deixando-os com uma tarefa cara, complexa e difícil de abandonar.

Exemplos: AWS QuickSight, Microsoft Power BI, Google Looker, Salesforce's Tableau, Oracle Analytics Cloud

Habilidades proprietárias

Algumas plataformas de business intelligence oferecem a oportunidade de aumentar os recursos da sua plataforma. Por exemplo, criando conectores personalizados para fontes de dados, implementando gráficos personalizados, criando cálculos personalizados ou implementando elementos de interface de usuário (IU) personalizados. Muitos fornecedores também exigem que os desenvolvedores aprendam linguagens e estruturas de script ou programação específicas. Isso significa que estes elementos específicos provavelmente precisarão ser refeitos se uma nova plataforma precisar ser escolhida, e um tempo significativo será necessário para aprender essas linguagens proprietárias, o que é completamente desperdiçado, pois sua equipe não será capaz de usá-las depois de mudar para uma nova ferramenta.

Exemplos: **Looker LookML, Microsoft DAX, Qlik Expressions, ThoughtSpot Modeling Language**

Como o Yellowfin aborda as arquiteturas abertas

O Yellowfin sempre foi construído com base no princípio de maximizar a escolha dos nossos clientes. Nós desenvolvemos uma série de recursos para fornecer opções adequadas para a mais ampla variedade de necessidades.

Flexibilidade: por ser construído na plataforma Java, Yellowfin é uma solução totalmente integrada e portátil que pode ser executada em qualquer sistema operacional moderno, executado em servidor local ou em Cloud.

Baseado na Web: não há componentes de desktop do Yellowfin; todo o acesso do usuário é feito via navegador, simplificando a implantação, desenvolvimento, utilização, administração e manutenção..

Direct Query: Yellowfin não requer o carregamento de dados; em vez disso, ele se conectará a qualquer fonte de dados desejada e executará consultas em Real-Time na sua fonte de dados (OTLP, ROLAP, MOLAP, Colunares, outros bancos analíticos ou serviços).

Extensibilidade: Programar no Yellowfin não requer conhecimento de linguagens específicas ou scripts proprietários, pois as extensões podem ser codificadas usando linguagens de desenvolvedor familiares como Java, JavaScript, CSS e HTML.

Plataformas independentes: Os trabalhos de ETL para a criação de um Banco de Dados analíticos são normalmente as fases mais demoradas de um projeto de BI, mas com o Yellowfin os resultados da preparação de dados e queries SQL podem ser aproveitados em outras ferramentas analíticas.O recurso de Histórias pode ser usado em outras soluções e integrar painéis e relatórios de plataformas concorrentes como Microsoft Power BI, Qlik e Tableau.

Menos tempo e custo: Garantimos que as futuras alterações em seu negócio possam ser feitas com o mínimo de interrupção, agilidade, menor tempo e custo para serviços de migrações.

Acreditamos que Yellowfin é um dos melhores exemplos de uma solução de business intelligence construída em torno dos benefícios da arquitetura aberta.