

Alguns esclarecimentos preliminares.

Pretende-se, com este texto, sensibilizar os leitores ser grandes conhecimentos informáticos, pelo que se torna conveniente esclarecer alguns conceitos fundamentais, ainda que muito ligeira e sumariamente.

Podemos considerar o computador constituído por dois elementos fundamentais:

1) Hardware (quinquilharia) é o conjunto dos elementos electrónicos e electromecânicos, isto é, tudo o que tem peso e ocupa espaço.

2) Software (sem tradução, nome dado por oposição ao hardware) - programas de um modo geral, quer sejam de gestão do computador, quer de aplicações. O facto de um programa residir em memória, não vem aumentar o volume ou o peso total do computador.

Entre o funcionamento de um computador (Hardware - Software) e um vulgar motor de automóvel pode-se estabelecer no paralelo grosseiro, mas sugestivo.

Consideremos um motor com os seus componentes fundamentais, cilindros, válvulas, engrenagens de sincronização, árvores de cames, distribuidor, velas, etc.

Numa 1ª fase consideremos o motor dessincronizado, desafinado, enfim, impossibilitado de funcionar.

Numa 2ª fase consideremos o motor perfeitamente sincronizado e afinado, "programado" pela mão do mecânico, podendo funcionar e produzir trabalho.

Da 1ª fase para a 2.ª fase, o motor não sofreu qualquer alteração de peso ou de estrutura, porém, passou a funcionar apenas porque foi afinado, isto é, a sequência no tempo das várias operações elementares foi convenientemente programada, externamente, pelo mecânico.

Voltando ao computador, a 1ª. fase poderá ser associada ao Hardware desorganizado e incapaz de produzir trabalho; a 2.ª fase poderá associar-se ao conjunto Hardware + software, cabendo a este último a tarefa de "organizar" o Hardware, internamente, de molde a esta simbiose poder produzir trabalho.

Palavras chave

Hardware

Software

1. Introdução

No início da indústria dos computadores os fabricantes forneciam todo o Software e Hardware num preço só, de aluguer ou venda. Esta técnica, conhecida por "bundling", englobava programas para qualquer tipo de aplicações, completamente livres de encargos, que os fabricantes forneciam para estimular e apoiar a venda dos seus computadores. Esta situação tem vindo a alterar-se ao longo do tempo, e em resposta não só a pressões dos utilizadores e a acções anti-trust do governo dos EUA, mas também a evolução dos custos relativos do Hardware e do Software (fig. 1.1), a maioria dos fabricantes adoptou o "unbundling", isto é, Separou os preços do Hardware e do Software.

A tendência actual dos constructores é para incluírem no preço base apenas o Software básico - sistema de operação e compiladores, tendo o restante software preços próprios de aluguer ou compra.

Da análise dos custos relativos do Hardware e do Software, verifica-se que os custos deste último

excedem os do hardware (New Scientist - 1974), atendendo ainda à evolução dos custos de desenvolvimento de Software (ver fig. 1.1), torna-se economicamente sensato, promover a sua circulação, através de permuta ou comercialização, ao esquecendo porém, os condicionalismos deste tipo de transferência.

Face à evolução dos custos de desenvolvimento de Software, tornando -se economicamente sensato, promover a sua permuta e, ou comercialização na medida do possível.

Palavras chave
Bundling

Unbundling

2.A Indústria de Software

Os fabricantes de Computadores escreveram os primeiros programas e forneceram -nos, gratuitamente, aos seus clientes; muitas vezes receberam destes, programas desenvolvidos nas respectivas instalações, os quais eram posteriormente distribuídos a outros utilizadores. Este intercâmbio de programas acabou por dar origem a associações de utilizadores – SHARE, por exemplo, é uma das maiores associações de utilizadores IBM, e foi formada por esse propósito de intercâmbio.

Nesses termos iniciais da computação era de facto impensável propor-se a venda de um programa.

Os fabricantes forneciam directa ou indirectamente, programas de todos os tipos, cuja qualidade variava bastante. No entanto, devido ao seu grande interesse em satisfazer os clientes, esforçavam-se para que tudo trabalhasse e trabalhasse bem.

Cerca de 1950 apareceram as primeiras companhias de Software cuja tarefa era desenvolver e instalar programas de aplicações e certos programas de sistemas para outras firmas. A maioria destas "casas de Software" desenvolveu-se nos EUA, quando o governo federal "descobriu" que não pagava salários, suficientemente elevados para contratar bons programadores, tendo por isso, de recorrer a indústria privada de Software. Uma das características, dessa indústria, residia no facto de o cliente das "casas de Software" receber programas feitos, especialmente para si.

Nos anos 60 foi a vez dos grandes constructores de computadores verem esgotar-se os seus recursos de programação, esta situação obrigou-os a recorrerem as "casas de Software", nomeadamente a IBM, para a produção de Software do sistema/ 360. Por outro lado a proliferação de Computadores na indústria provocou uma falta de programadores e analistas qualificados, não podendo a generalidade das empresas manter uma boa equipa informática, todavia, podiam encomendar os seus programas a empresas privadas de Software.

A situação descrita provocou um grande BOOM na indústria privada de Software, e em finais de 1968, cerca de 2800 firmas consideravam-se a si próprias "casas de software". No entanto, devido a falta de estruturas, muitas faliram e no Início da década de 70, o seu número era pouco superior a 1000. Durante o período de expansão, as "casas de Software" aperceberam -se que podiam criar programas de grande qualidade e de aplicação tão geral, que podiam ser vendidos sem modificação a vários clientes. De facto, o programa não era vendido, mas apenas uma cópia e o direito de utilização.

Estavam criadas as condições necessárias para se passar da manufactura de software para uma nova fase de cariz tipicamente industrial.

Em 1972, das 1000 firmas então existentes, cerca de 700 ofereciam para venda ou "leasing", conjuntos de programas prontos a funcionar para aplicações genéricas. De um total de quase 3000 programas diferentes, cerca de 60 deles já tinha ultrapassado um milhão de dólares em vendas.

Os valores que se indicam para as vendas da indústria de Software, segundo a "Computer Usage", são sugestivos do seu desenvolvimento; e 1969 as vendas atingiram 20 milhões de dólares, em 1973, 410 milhões de dólares e para 1985, cor. um certo optimismo, há quem preveja um mercado de 4 bilhões de dólares.

Actualmente existem vários tipos de fornecedores de software, aos quais se encontrar associadas diferentes condições de transferência; poderemos considerar, fundamentalmente os seguintes tipos

- Fabricante de Computadores - tradicionalmente fornece o software de sistema, quase todos têm uma grande variedade de Software de aplicações cruciais e científicas. É normalmente uma boa fonte de fornecimento, com

uma assistência na implementação e utilização bastante razoável, ou mesmo boa, porém, os preços praticados são bastante desencorajadores.

-Casas de software

Estas casas especializadas, no desenvolvimento e comercialização de Software, representam de facto uma alternativa válida. Contudo algumas precauções devem ser tomadas, nas brochuras publicitárias os programas fazem maravilhas e são muito fáceis de implementar, porem a realidade é por vezes, bem diferente e bem amarga.

-Comerciantes de software

Companhias que apenas comercializam software desenvolvido por terceiros. A qualidade dos programas varia consideravelmente, devida á diversidade de origens que podem ter. Um dos inconvenientes de adquirir software através destas companhias reside na dificuldade ou impossibilidade de contactar os programadores originais.

-Grupos de utilizadores

Desde os primórdios da computação que os utilizadores trocaram entre si software, a documentação anexa a estes programas, não é, dum modo geral, completa, mas a facilidade de contacto com o programador original pode permitir a superação das dificuldades que possam surgir.

-Universidades

Actualmente, as Universidades são uma fonte de software de boa qualidade, facto reconhecido pela indústria em geral e pelos próprios fabricantes de computadores, que frequentes vezes a elas recorrem para desenvolvimento do seu software de exploração ou de aplicação.

Esta fonte tem como principal atractivo a qualidade do software desenvolvido, o relativo baixo custo e a facilidade de contacto com o programador original, porém, a, manutenção dos programas torna-se problemática, devido a normalmente não existirem estruturas para o efeito.

Palavras chave

Associação de utilizadores Manufactura de software

SHAPE Indústria de software

"Casas de software" Comerciantes de software

Fabricantes de computadores universidades

3. Transferência de software

Antes de dar início a uma tarefa deverão os responsáveis, Programadores e Analistas, procurar documentar-se sobre os trabalhos já realizados nessa área ou áreas, afins, devem pôr a si próprios algumas permutas como as seguintes:

-Esta situação é totalmente nova?

-Onde existirão publicações sobre este assunto?

-Procurei bem e não estou a duplicar o trabalho que alguém já fez?

Estas questões devem de facto ser postas e respondidas tão exaustivamente quanto possível, porém, o facto de se, concluir que a situação não é original e que já existe muito trabalho feito, não deverá ser motivo suficiente para se comprar um programa, ou um conjunto de programas que "resolvam" o problema, há factores de vária ordem que de vem ser ponderados.

Os benefícios da transferência de software repousam em cinco crenças (para as quais existem contra argumentos).

i. Evita a "reinvenção da roda". Argumentam os profissionais que o desenvolvimento de software dentro da instalação resulta mais eficiente se for "reinventado".

ii. Todos ganham. A satisfação profissional entre os programadores desce, porque a implementação é menos excitante que a programação.

iii. É mais barato que o desenvolvimento. De facto é possível obter uma poupança considerável se se verificar um conjunto particular de condições, cuja ausência só se reconhece com "lupa" isto é, é necessário que alguém "mergulhe" num programa que não conhece, á procura das condições particulares que geralmente se encontram omissas na documentação.

iv. Coloca ao dispor de todos, todas as capacidades do programa. Existe uma baixa procura de utilização dessas capacidades por que nem todos precisam realmente dessas capacidades.

v. Permite uma rápida "fixação de tecnologia". Provavelmente não existe nenhuma aplicação que descreva exactamente o problema a tratar, nem todo o problema fica enquadrado pelo programa; a aplicação é tão genérica que se torna ineficiente para tratar a situação concreta que se nos depara. Apesar de todos as reservas que se possam fazer á transferência de Software, as quais devem ser convenientemente ponderadas, é óbvio que é uma hipótese a ser tomada em consideração, antes de se desenvolver internamente qualquer aplicação.

A evidência sugere e a pratica confirma que a transferência é tão mais frutífera, quando o dador e o receptor possuírem níveis elevados de know-how e experiência.

Antes de entrar em detalhes mais alguns aspectos de carácter geral devem ser focados. Uma restrição adicional a considerar é a existência de documentação, não só sobre a implementação, mas também sobre os algoritmos á volta dos quais o Software foi construído.

Deverá ainda ser analisada a existência de tipos particulares de dependência dos programas relativamente a:

-estruturas de dados.

-configuração da instalação onde foi desenvolvido.

3.1 - Natureza do produto e portabilidade

Ao discutir a portabilidade do Software (possibilidade de transporte de um computador para outro) é útil distinguir três tipos de aplicações:

- "Packages"

-Programas

-Rotinas

Os "packages são desenhados de forma a proporcionar facilidades de trabalho a utilizadores sem experiência informática, são frequentemente um quebra-cabeças para serem mudados de um computador para outro.

As rotinas são mais facilmente transportáveis, mas requerem conhecimentos de programação para serem utilizadas. Os programas estabelecem uma transição entre os dois.

Os Packages são desenvolvidos para serem o mais geral possível, de modo a abrangerem uma grande fatia de potenciais utilizadores, o seu modo de utilização procura ser próximo da linguagem do utilizador, esta filosofia acarreta consequências quando se trata com aplicações específicas (ex.: Processamento automático de imagens, acarreta a manipulação de grandes volumes de dados).

Os Programas, por outro lado, são mais específicos nas suas aplicações e portanto abrangem um menor grupo de potenciais utilizadores.

A portabilidade dos packages e dos programas pode levantar problemas delicados:

- i. Dificuldade devido a diferença de códigos.
- ii. Dificuldade devido á inexistência do compilador adequado.
- iii. Dificuldade devido a terem sido utilizadas instruções não standard da linguagem.
- iv. Dificuldades ao correr o programa por não caber na memória.
- v. Dificuldades mesmo quando o programa corre, mas os resultados obtidos são diferentes dos da máquina original devido is características de Hardware.

Apesar de todos estes obstáculos, um programador experiente consegue, geralmente, pôr a funcionar, se tiver a vontade de despende o esforço suficiente para tal. Este esforço que não é pouco, é amplificado quando a documentação é insuficiente e pode ser facilitado, se for possível contactar o programador original.

A terceira hipótese envolve rotinas para as quais existem algumas vantagens específicas.

As rotinas são muito mais portáteis que os programas ou packages; com algum cuidado e possível programá-las de forma a não ficarem dependentes de qualquer computador; oferecem grande flexibilidade na sua utilização, o utilizador pode combiná-las convenientemente, ajustando-as ao algoritmo escolhido para executar uma determinada tarefa; o desenvolvimento e programação de rotinas permite que a investigação e implementação do Software continue com unidades bem definidas e manuseáveis, bem como permite responder com flexibilidade e velocidade razoáveis a novos requisitos.

Independentemente da hipótese considerada Package, Programa ou Rotina e necessário ter em conta a linguagem de programação utilizada, de molde a providenciar os meios que garantam ou verifiquem se o produto se mantém dentro das normas standard.

Para a linguagem FORTRAN IV existe um excelente programa que faz as verificações dessas normas PFORT (Lab. Bell, Julho 1975).

3.2 - Custos e Propriedade

Muitas das trocas de Software têm sido feitas numa base amigável onde os custos, se existem, são meramente simbólicos e os direitos de autor são uma questão que raramente se põe. Contudo quando as aplicações se tornam demasiado interessantes ou o clima financeiro o proporciona, verificam-se alterações profundas a este quadro, podendo as peças de Software atingir preços que podem variar de 150 a 10.000 contos (\$3,000 a \$200,000) conforme as aplicações e o grau de sofisticação atingido.

Levantam-se então problemas com os direitos de autor; estabelecimentos comerciais, Universidades ou outras entidades podem reclamar a propriedade de Software desenvolvido pelo respectivo pessoal, porém, só o farão, se tiverem a garantia de poder realizar algum lucro através da venda ou aluguer do software, ou com a esperada manutenção deste. Esta situação, que varia de país para país, pode influenciar as relações de permuta de Software; por exemplo, enquanto nos EUA o Software desenvolvido com financiamento governamental é tornado do domínio público, na Grã-Bretanha é comercializado pela National Research and Development Corporation, entre outras.

Internacionalmente, a posição quanto a direitos de autor é bastante complexa. No contexto de Software, os direitos são referidos à listagem do programa, excluindo o algoritmo que lhes serve de base, isto é, a "tradução" de programas infringe os direitos de autor, enquanto a reprogramação o não faz. Convém salientar que também esta situação varia de país para país, aguardando acordo internacional.

3.3- Avaliação de um "Package"

Sendo a importação de Software uma alternativa a considerar quando se projecta a implementação de novas aplicações, apresenta-se uma lista de referência para avaliação de packages, esta lista procura avaliar packages por este software ser de difícil implementação e muito atraente para utilizadores não informáticos, porém, com pequenas alterações, poderá aplicar-se ao Software em geral.

A seguinte lista foi apresentada por R.L. Tomlinson (Canadá 1974) a potenciais compradores de Software "packages":

i. Não compre packages novos. Isto é perigoso porque o programa pode conter erros e documentação insuficiente. Se o Package for barato é natural que não seja revisto. Procure packages com dúzias, senão centenas de utilizadores.

ii. Não espere acção imediata. Um período de 4 a 6 meses de consulta é típico quando se lida com fornecedores dos EUA. (Para o caso português este período pode ser ampliado até ao dobro)

iii. Não pense que os custos terminam quando da compra, alguns custos extras serão necessários para manter e armazenar o programa.

iv. Não compre versões de programas já compilados.

v. Não espere nenhum apoio do fornecedor. Se precisar e conseguir algum, tudo bem, senão prepare-se para deitar o package fora.

vi. Seja selectivo. Escolha apenas o Software que de facto lhe convém, deixe o restante para o seu fornecedor favorito.

Para além destas premissas um pouco irónicas, mas cheias de realismo deverá ainda considerar as seguintes:

vii. Documentação - Verifique se o Departamento de informática compreende completamente a documentação e a considera suficiente sendo capaz de promover as necessárias alterações.

viii. Estabilidade do vendedor - verifique se o vendedor vai continuar no ramo e se poderá fornecer os posteriores desenvolvimentos do mesmo "package".

ix. Compatibilidade - Se for necessário despendar mais de 50% do preço do Package a modificá-lo, então provavelmente, não valerá a pena comprá-lo.

x. Funções incluídas - Verifique quantas das funções desejadas são executadas pelo Package.

xi. interface com o utilizador- a dificuldade de usar o Package? Os dados são fáceis de perceber e preparar?

xii. Flexibilidade - Pode o package continuar a ser utilizado se os requisitos mudarem ligeiramente?

4. Elacções a tirar de um caso recente

Procurando reforçar o que ficou escrito, apresenta-se a traços largos, uma situação que ocorreu no CCUL entre 1978 e 1979.

-Em Fevereiro de 1978 o CCUL recebeu da TCSC - The Computer Software Company, em regime de aluguer, um novo sistema de exploração - EDOS para o seu IBM 360/44.

-Seguiu-se um período de consultas, tendo inclusivamente vindo a Portugal um elemento da TCSC para melhor esclarecimento.

-Em Fevereiro de 1979 o novo sistema de exploração continuava por implementar.

Em Junho de 1979 o responsável pelo Suporte de Sistemas deslocou-se aos EUA para efectuar uma série de consultas na TCSC.

-Em Agosto de 1979 o programa já corria.

Desta, que é a nossa mais amarga experiência de implementação de Software importado, duas conclusões se podem tirar:

1 -O Know-how interno é indispensável para a implementação e bom funcionamento do Software importado. (Se não existir, é necessário arranjar esse know-how entre elementos geograficamente próximos).

2 -O contacto directo com os responsáveis directos pela elaboração do Software, torna-se indispensável para uma rápida e efectiva implementação do Software Importado.