

UNIME — União Metropolitana de Educação e Cultura



Sílvio Vanderlei Araújo Sousa

Estratégias Competitivas na Indústria de Software

Lauro de Freitas

2005

RESUMO

Este trabalho compreende um ensaio teórico acerca das estratégias competitivas adotadas na indústria de software. Para a sua realização foi efetuada a revisão do referencial teórico sobre estudos de competitividade acrescida de estudos e trabalhos concernentes às empresas e à estrutura de mercado da referida indústria. Para tanto, inicia-se com a introdução onde são discutidos objetivos e justificativas para a realização deste estudo. Em seguida, é realizada uma breve discussão sobre seus processos produtivos, destacando-se os produtos obtidos em cada etapa de sua cadeia de valor. Completando o trabalho, são analisados os padrões de concorrência predominantes, juntamente com a percepção das estratégias competitivas empregadas aos segmentos de mercado da indústria.

PALAVRAS-CHAVE: ESTRATÉGIAS COMPETITIVAS – SOFTWARE;
COMPETITIVIDADE – INDÚSTRIA – SOFTWARE; DESENVOLVIMENTO ECONÔMICO
– SOFTWARE.

SUMÁRIO

1 INTRODUÇÃO	4
2 O PROCESSO PRODUTIVO	8
3 ESTRATÉGIAS COMPETITIVAS NA INDÚSTRIA DE SOFTWARE .	14
4 CONCLUSÕES	30
REFERÊNCIAS	34

1 INTRODUÇÃO

Neste trabalho, entende-se por software uma seqüência lógica de instruções que coordenam a operação de um processador com função e comportamento definidos. O software exige para a sua construção um projeto de engenharia conforme especificação definida pelo usuário – cliente, o qual se encontra dividido em várias etapas. Dessa forma, percebe-se que a produção de software envolve uma série de etapas interdependentes, as quais exigem conhecimentos e competências específicas.

Seguindo essa abordagem, considera-se a indústria de software como o conjunto de firmas que possuem capacitações similares para a prestação de serviços nas etapas do processo de construção de software.

A definição tradicional de indústria tenderia à formação de vários tipos de indústria de software. O problema se encontra no conceito de similaridade de software, que deve ser entendido de forma diferente da similaridade para um produto manufaturado, pois, embora existam diversos softwares para áreas de aplicações comuns, suas funcionalidades e objetivos podem ser bens distintos, não cabendo nesses casos as classificações de similaridades.

Neste trabalho, opta-se também pela definição de serviços adotada por Teixeira e Guerra (2002, p.195). Esses autores observam que “[...] a tradicional divisão entre setor primário, secundário e terciário apóia-se na distinção entre atividades que resultam em trocas no mercado de bens materiais e todas as demais que não possuem um claro relacionamento com algo tangível são identificadas como serviços”.

Portanto, de acordo com esses conceitos, considera-se que a indústria de software se constitui de empresas cuja atividade principal se concentra na troca de serviços de construção de software com o mercado.

A escolha desse setor como objeto de estudo da pesquisa está baseada no papel que o software representa na dinâmica das economias modernas. O crescimento econômico se manifesta cada vez mais associado à introdução de inovações tecnológicas inerentes à Tecnologia da Informação (TI), que está rapidamente reestruturando as formas de competir em praticamente todos os setores produtivos.

A expansão do setor de software seria fundamental para o aumento da produção e da exportação de bens e serviços de alto valor adicionado, seguindo o exemplo da indústria de software de regiões localizadas em países como Irlanda, Índia e Israel.

A questão central sobre o software é que seu emprego se tornou uma característica dominante nas economias modernas devido à contribuição para a aceleração do processo de desenvolvimento tecnológico. A maciça utilização da TI, em paralelo aos acordos internacionais de comércio, aos investimentos nas malhas de transporte e ao desenvolvimento dos sistemas de telecomunicações, eliminou algumas das antigas barreiras de comércio e proporcionou a melhoria da eficiência do fluxo de distribuição de bens e serviços.

As atividades relacionadas à TI, teriam, nesse contexto, um importante papel no desenvolvimento social e econômico de uma determinada região. Se, por um lado, poderiam fortalecer as empresas inovadoras, por outro, poderiam servir de ferramenta para o desenvolvimento social, seja através da geração de novos postos de trabalho e renda, seja através do suporte tecnológico à difusão de informação na

sociedade.

Levando-se em consideração o diamante de Porter (1989), percebem-se ainda outras justificativas para a viabilização de um estudo na indústria de software. Esse autor destaca algumas implicações importantes decorrentes dos relacionamentos entre clientes e fornecedores. Entre elas, pode-se destacar o papel peculiar da indústria de software para o desenvolvimento das demais indústrias de uma determinada região. Evidencia-se, aqui, a importância do seu apoio às demais indústrias provendo a base tecnológica necessária às atividades dinâmicas das grandes empresas da atualidade.

A existência de uma indústria de software consolidada se faz essencial à manutenção dos negócios das empresas, haja vista o intenso uso da TI nas organizações modernas. A cooperação com a indústria de software poderia promover maior desenvolvimento tecnológico e reduzir os custos de transação. Em paralelo, fomenta-se a criação de postos de trabalho, proporcionando maior desenvolvimento social à região.

Um país sem uma posição favorável em serviços perderia cada vez mais em renda nacional, bem como em vantagens para a posição competitiva de outras indústrias. Logo, a indústria de software também se apresenta importante sobre o aspecto da manutenção dos demais empreendimentos empresariais.

Ademais, busca-se o conhecimento sobre as estratégias competitivas adotadas na indústria de software e o grau de penetração nos segmentos de mercado.

Neste sentido, tentar-se-á documentar a perspectiva da concepção das estratégias para competição não só sob uma orientação de valorização do capital, como também do volume das capacitações, do conhecimento acumulado e das

possibilidades de desenvolvimento social.

Afinal, a análise do conteúdo apresentado até aqui permite esboçar o objetivo central do trabalho que se refere ao mapeamento das estratégias competitivas em consonância às especificidades da indústria de software.

Dadas as justificativas apresentadas anteriormente, a estrutura deste documento foi dividida em quatro seções de forma a orientar o leitor à compreensão do tema: esta, que trata das justificativas e problematização acerca do trabalho, e outras três que contemplam os aspectos mais relevantes oriundos da realização do presente estudo.

Na segunda seção, é apresentada uma abordagem para a produção de software, descrevendo-se as etapas e subprodutos gerados na cadeia de valor.

Na terceira, são discutidos os aspectos relevantes para a elaboração e implementação das estratégias competitivas na indústria e suas implicações nas estruturas de mercado.

Por fim, insere-se uma seção contendo as conclusões do trabalho, tomadas como base as percepções apresentadas nas demais seções.

2 O PROCESSO PRODUTIVO

Atualmente, a produção de um software se compõe de várias fases, nas quais são utilizados conjuntos diversos de técnicas, modelos e ferramentas encadeadas logicamente de forma a guiar o analista de sistemas e o programador no processo de sua construção. Para o desenvolvimento das fases, geralmente são empregadas metodologias cujos principais objetivos são aumentar a qualidade do produto final e melhorar os índices de produtividade do processo.

As etapas de construção do software são bem distintas e modulares. Embora não exista uma metodologia única de desenvolvimento de projetos de softwares, uma proposta de ciclo de vida de projeto pode ser concebida dividindo-se as atividades nas seguintes etapas:

- Proposta Técnica;
- Especificação de Requisitos;
- Análise do Sistema;
- Prototipação;
- Projeto do Sistema;
- Implementação;
- Testes;
- Implantação;
- Manutenção.

Cada etapa é concretizada de forma a se produzir resultados a serem utilizados nas etapas seguintes. A transição entre etapas deve ser reforçada principalmente na passagem da Especificação de Requisitos para a Análise do

Sistema e na passagem da fase de Prototipação para o Projeto do Sistema. Isto se deve ao fato de que estes são pontos críticos da metodologia de desenvolvimento, em que a validação do usuário é condição essencial para que se possa avançar no processo de desenvolvimento. As demais transições são mais suaves e às vezes imperceptíveis.

A elaboração da proposta técnica é uma atividade essencial para a definição do escopo do sistema. Esta definição deve especificar claramente os limites de atuação do sistema, suas principais funcionalidades e arquitetura tecnológica a ser utilizada na solução. É o documento que irá servir para quantificar os recursos necessários a todo o processo de desenvolvimento e implantação do software.

A especificação de requisitos é a fase em que o analista de sistema e o gerente do projeto especificam, detalhadamente, as condições que o produto final deverá satisfazer. Estas condições representam a tradução dos requisitos funcionais, de informação e de interface do projeto.

A análise do sistema é a fase onde o analista identifica as classes de negócio da aplicação, seus relacionamentos, principais atributos e métodos.

Na prototipação, são criados modelos do software que permitam ao usuário navegar entre telas e simular as funcionalidades do sistema, conhecendo o seu potencial e apontando eventuais falhas e não conformidades.

O projeto do sistema deve ser iniciado após a aprovação do protótipo e irá incorporar as classes que implementam as funcionalidades do software para fazer com que o sistema funcione. São classes de suporte a acesso a bancos de dados, construção de interfaces, comunicação em rede, conversão de dados, criação de objetos etc. Idealmente, este conjunto de classes deve estar predefinido e

organizado em um *framework* de forma a facilitar o projeto da arquitetura do software e sua futura implementação. O *framework* representa a arquitetura do sistema e se mostra essencial nesta etapa para ratificação dos padrões de desenvolvimento e obtenção de ganhos de escala na produção.

A implementação do sistema é a etapa na qual o programador codifica o sistema e deve ser a atividade mais simples de todo o processo, pois todas as definições conceituais, lógicas e de arquitetura já devem ter sido realizadas nas etapas anteriores. Nesta etapa, recomenda-se a utilização de ferramentas automatizadas para reduzir o tempo de codificação através da geração automática de código.

O teste do sistema é a etapa onde se validam as especificações de requisitos e análise do sistema e são executados pela equipe de desenvolvimento à medida que os módulos ou componentes vão sendo disponibilizados. Ao final do processo, é necessária a realização de testes de integração entre as partes do sistema e o ambiente do cliente.

Todas estas etapas devem estar sempre sendo acompanhadas pelo comitê de liderança, principalmente pelos representantes do cliente, para que se evite a descoberta tardia de problemas de concepção do software.

A implantação do software é um processo que requer um planejamento específico, pois envolve mudanças operacionais nas rotinas dos usuários, inspira medo e desconfiança nas pessoas e pode comprometer todo o trabalho de desenvolvimento. É preciso conscientizar os usuários da importância da utilização do software para a organização, contar com o apoio da alta gerência e planejar todos os passos do processo (instalação, treinamento, entrada de dados, estabilidade, *performance*, etc.).

Ao final do trabalho, espera-se que o software esteja atendendo e superando as expectativas do cliente, pois a falta de adequação aos requisitos se traduz em deficiência de qualidade e, em alguns casos, poderá implicar o fracasso do projeto.

Em relação à manutenção, esta ocorre ocasionalmente após a etapa de implantação e decorre geralmente de melhorias ou correções nos sistemas.

No decorrer das etapas do processo de desenvolvimento de software ocorre uso intensivo de capital intelectual apoiado pela aplicação de vários métodos, como FPA (*Function Points Analysis*), CMM (*Capability Maturity Model*), Detecção e Prevenção de defeitos, ferramentas CASE (*Computer-Aided Software Engineering*), *Frameworks* e normas, como ISO/IEC 9126, ISO/IEC 12119, ISO/IEC 12207, de forma a se obter maior produtividade e melhoria da qualidade global do projeto.

Análises dessas etapas permitiram observar que se dá uma maior adição de valor nas fases de análise e projeto do sistema, enquanto as etapas de programação e testes, instalação e manutenção se posicionaram no nível mais baixo de retorno da cadeia de valor (HEEKS, 1996).

Além disso, os processos envolvidos em cada fase não acontecem necessariamente na localidade do cliente, nem do fornecedor. A capacitação de diversos centros produtores de software, aliada ao desenvolvimento das tecnologias de comunicação, permitiu que algumas etapas do processo de produção de software fossem destinadas a outros centros, exigindo, entretanto, um rigoroso controle da qualidade do projeto. Isto permite que algumas empresas se concentrem em etapas do processo que lhe proporcionem maior retorno e sejam mais críticas ao projeto e destinem às outras empresas atividades menos intensiva em conhecimento, haja vista a aplicação de ferramentas automatizadas para estas fases. Nesta linha, a

análise de requisitos de software para um projeto particular pode acontecer no local demandante, utilizando-se os analistas de sistemas, enquanto a etapa de programação pode ser implementada em outro local, utilizando-se mão-de-obra menos especializada, uma vez que já se tenha realizado toda a fase de projeto do sistema. Na terminologia do mercado de software, essa prática costuma-se se chamar de *outsourcing* e vem sendo muito utilizada pelos países desenvolvidos destinando a países periféricos, como Israel e Índia, o papel de codificar as especificações dos sistemas (HEEKS, 1996).

Assim, tanto a codificação quanto a fase de teste podem ser realizadas por empresas contratadas em outros locais, entretanto dependem de vários fatores inclusive a disponibilidade de pessoal capacitado, o custo, a habilidade para cumprimento de prazos, a existência de rede de telecomunicações velozes e a utilização de metodologias de gerenciamento à distância. Já outras etapas como a instalação e a manutenção evolutiva devem ser realizadas impreterivelmente no cliente. A manutenção evolutiva, por exemplo, exige a elaboração de diagnóstico e análise, o que requer o exercício de um analista de sistema. Ademais se conclui que essas estratégias de desenvolvimento de software exigem uma boa compreensão dos processos, mostrando-se fundamental uma ampla cooperação entre as empresas que participam do projeto.

A trajetória tecnológica da indústria de software esteve desde o início atrelada à trajetória de desenvolvimento do hardware. Entretanto, esta começou a se desenvolver de forma independente, e sua autonomia originou uma série de conceitos e tipologias que guiam o pesquisador para o entendimento dessa indústria. Os estudos realizados na indústria de software tendem à formação de tipos e classificações (segmentação de mercado), possibilitando duas vertentes distintas

de análises: a) a forma como os produtos chegam ao mercado: pacote, customizado ou embarcado; b) o tipo de domínio em que será aplicado: horizontal ou vertical.

Na Seção 4 serão apresentadas as implicações dessa tipologia nas estratégias competitivas adotadas na indústria de software.

3 ESTRATÉGIAS COMPETITIVAS NA INDÚSTRIA DE SOFTWARE

De acordo com Gaio (1990), software é parte intrínseca e penetrante do paradigma de TI que tem características estranhas e complexas. Primeiro, software é um elemento nuclear da TI por implementar as características fundamentais de sistemas baseados em microeletrônica que possuem a capacidade para executar uma gama extensiva de funções. Segundo, software é uma ferramenta que gera um produto intangível, cujo valor é determinado pela capacidade efetiva das operações computacionais que são empreendidas e pela segurança da representação dos domínios da aplicação que é modelada, seguindo critérios estabelecidos pelos usuários. Em síntese, software representa a incorporação do conhecimento em produtos e sistemas de produção.

Desta maneira, o entendimento do termo "indústria de software" pretende alargar o conceito de indústria, para compreender um grupo de companhias com um produto característico (neste caso, software), independente do conceito tradicional de indústria de transformação de matéria-prima para a produção de mercadoria. Isto se torna necessário, primeiro, devido à modelagem do termo matéria-prima dentro da indústria, que se compõe basicamente de conhecimento e, também, por seus produtos que são uma sucessão de linhas codificadas denominadas "programas de computação" ou "software". Não apresenta, portanto, nada tangível, mesmo que tenham sido utilizadas ferramentas materiais (microcomputadores) na sua produção.

Além disso, a velocidade na qual as inovações são introduzidas e transformadas em produtos novos e sua rápida obsolescência são os destaques desta indústria que se mostra muito dinâmica. Dessa maneira, é possível destacar, algumas características básicas: "[...] a alta velocidade de introdução de inovações técnicas e no desenvolvimento de produtos, novos ou existentes; a competição acirrada; o baixo investimento em capital fixo; e a capacidade criativa e intelectual da mão-de-obra, que é o seu grande ativo" (BRANCO; MELO, 2003, p. 2).

Em consequência dessa dinâmica, o setor de software não se apresenta muito estável, verificando-se um elevado nível de nascimento e mortalidade de empresas. A mudança constante no perfil das organizações é outro fator que se destaca nessa indústria, tanto sobre o aspecto tecnológico, como também organizacional.

Pondé (1993, p. 21) observa que “[...] o avanço tecnológico e organizacional nas formas como o software é produzido constituía uma questão central para o futuro da indústria [...]” e a falta de capacitação (seja na qualidade do software ou nas habilidades dos profissionais) das empresas para enfrentar essa questão culminou na chamada crise do software. Este autor verificou que a dificuldade em se estabelecer um processo de construção de software com qualidade se mostrava como elemento central da discussão tecnológica sobre o progresso da indústria.

Ainda hoje, este aspecto se evidencia como fundamental para a obtenção de ganhos significativos de qualidade e produtividade na produção de programas de computador. Isso se materializa, de uma maneira tal que a difusão desigual entre as firmas do conhecimento de novas técnicas produtivas torna-se um determinante potencial na geração de assimetrias na distribuição de vantagens competitivas. O

entendimento dessa questão envolve três aspectos: a evolução das linguagens de programação, a difusão de ferramentas automatizadas de auxílio à

construção de softwares (CASE) e o surgimento de técnicas e métodos para a gestão de projetos de software.

As linguagens de programação, em especial aquelas baseadas em técnicas de orientação a objetos, tendem a evoluir para formas mais complexas de desenvolvimento, constituindo os *Frameworks* que exercem um significativo efeito na concepção e construção dos programas de computador, com incrementos consideráveis de qualidade e produtividade.

A vantagem de se utilizar esta tecnologia está na questão da reusabilidade dos códigos e ratificação de padrões, que proporciona ganhos de escala na construção do software. Os resultados podem ser medidos pela melhoria da qualidade, pela redução dos *bugs* (utilização de códigos pré-testados e estáveis) e pelo aumento da produtividade na programação.

Em relação à difusão de ferramentas CASE, verifica-se a automação parcial de diferentes atividades relacionadas ao ciclo de vida do projeto, com impactos positivos sobre a produtividade e qualidade. Deve-se destacar, contudo, que os ganhos potenciais oriundos da utilização destas ferramentas estão condicionados em grande parte à capacidade das empresas em aplicá-las corretamente nas etapas produtivas do software, de modo que sua difusão exige esforços de treinamento de pessoal e um aprendizado que leva tempo, além de envolver alto custo na aquisição de licenças de utilização.

O terceiro fator decorre de melhoramentos na forma de organizar e gerir as atividades de desenvolvimento de software, estabelecendo mecanismos mais eficazes para controlar, avaliar e dar suporte às atividades de projeto. Atualmente, a

sistematização do processo de produção de software obteve ganhos significativos, principalmente pela aplicação de técnicas como FPA e CMM que, entre vários objetivos, se propõem a promover melhorias da qualidade e produtividade no processo de construção de softwares. Entretanto a implantação do CMM exige não só conhecimento técnico e treinamento exaustivo, como um alinhamento estratégico com as metas da organização, sem o que sua aplicação fica inviabilizada, pois a adoção desse modelo, na maioria das vezes, implica uma ampla mudança no processo produtivo, exigindo esforços e investimentos elevados.

Outra questão importante para se mensurar a dinâmica desse setor gira em torno da composição e evolução da base instalada de hardware, que constitui um importante parâmetro para a atuação das empresas de software, pois delimita as perspectivas de crescimento da demanda por tipos específicos de programa. A difusão dos microcomputadores nos anos oitenta e a criação de mercados bilionários para diversos tipos de pacotes exemplificam esta questão.

Dessa forma, novidade e heterogeneidades são características do jogo concorrencial das empresas estabelecidas no setor, que possuem produtos divididos em diversos segmentos.

Um exemplo concreto da heterogeneidade do setor de software consiste na evidência de que vários fabricantes de hardware elegeram o mercado de software como uma forma de diversificação, constituindo-se num foco destacado das suas estratégias de negócios. Este movimento tem origem na crescente tendência à padronização dos equipamentos de processamento de dados, reduzindo o espaço para a diferenciação de produto e intensificando a concorrência via preço, que acabou também impactando os mercados de sistemas de arquiteturas proprietárias, pois os usuários preferem produtos com padrões mais difundidos, de forma a reduzir

a dependência tecnológica de um único fabricante. Conseqüentemente, as margens de lucro caíram e as empresas buscaram ofertar software e serviços diversos junto com seus equipamentos, o que constitui uma forma de elevar o valor agregado dos sistemas comercializados e um meio de diferenciação de produto (PONDÉ, 1993). Além disso, à medida que a biblioteca disponível de aplicativos constituiu um elemento fundamental nas decisões dos usuários referentes a aquisições de plataformas de hardware, os fabricantes destas procuraram firmar parcerias com as *softwarehouses* de modo a garantir uma oferta adequada de software compatível com os seus sistemas.

Destaca-se ainda a perspectivas dos retornos crescentes obtidos a partir da difusão de softwares específicos, cuja ampliação da base instalada proporciona externalidades de rede providenciando um importante ganho de competitividade (GRAEML, 2003).

As *softwarehouses*, por sua vez, conseguiram evoluir para outras formas de comercialização de serviços que a tradicional produção de software, percebendo-se uma nova divisão de segmento estabelecido sobre o tipo de atividade desempenhada na cadeia de valor, como a elaboração e aplicação de metodologias e a consultoria em etapas de projetos. Estas são atividades que envolvem competências acumuladas por algumas empresas ao longo de seu funcionamento e foram reformuladas para atender à demanda proveniente de outras empresas do setor ou de usuários finais.

A característica mais importante dessas atividades é que além do uso intensivo de conceitos de software, é necessária a utilização de capacitações prévias oriundas do conhecimento acumulado ao longo da vida da organização, combinada a uma forte articulação com os demandantes de serviços, que muitas vezes são

empresas que também atuam no setor de software.

O que se verifica é o deslocamento das competências ao longo da cadeia produtiva do software, na qual a fragmentação em etapas permite a atuação especializada de algumas empresas com considerável sucesso. Uma das grandes limitações dessa estratégia é a distância geográfica dos centros consumidores de serviços. Embora algumas metodologias de gerenciamento de projetos de software contemplem a coordenação à distância, verificam-se aí alguns elementos que dificultam o processo: a questão da capacidade gerencial dos projetos, incrementada pelo tempo de resposta à demanda dos clientes, que se acentua à medida que não é possível estabelecer toda a infra-estrutura necessária para a execução do serviço no local demandante; e o incremento dos custos de transação, haja vista a necessidade de deslocamento de equipes que trabalham com dedicação exclusiva e a manutenção de estrutura de apoio à execução dos serviços solicitados.

Dessa forma, como “[...] mobilidade e organização tornam-se ativos cruciais para que um ofertante de serviço possa estabelecer uma relação bem-sucedida com clientes; e a qualidade e a eficiência na oferta de serviços é altamente afetada por barreiras territoriais” (TEIXEIRA; GUERRA, 2002, p.195), esses fatores se mostram prejudiciais ao desenvolvimento desse tipo de estratégia.

Ademais, com o aumento da complexidade e do tamanho dos sistemas, a ampliação das possibilidades de se utilizar conjuntamente produtos de distintos fabricantes, aliada à necessidade de se desenvolver soluções customizadas, faz com que cada vez mais soluções completas e totalmente configuradas sejam encomendadas aos fabricantes de software com a tarefa de combinar componentes de hardware e software em sistemas apropriados às suas necessidades, envolvendo esforços e competências em diversas áreas da computação. Segundo Pondé (1993),

tem-se nesta atividade um mercado no qual a diversificação em direção à integração de sistemas aparece como um caminho que provoca uma mudança na forma de comercialização dos programas, pois, na medida em que o contato direto com o usuário passa a ser feito pelo integrador, este último pode constituir um canal importante para a venda de softwares.

Por outro lado, analisando a divisão de segmentos segundo a forma como os produtos chegam ao mercado, percebe-se que o software de pacote atinge um amplo número de clientes, sendo normalmente padrão e tem a característica de que, durante sua produção, não existe interação direta entre o usuário e o fornecedor. Assim, não existe cliente exclusivo que imprima modificações no software de acordo com sua necessidade, o que torna possível o produto existir por si só, independentemente de requisitos de usuários.

A comercialização desse tipo de software se dá através de vendas em prateleiras e a estratégia de marketing e vendas é semelhante à utilizada para equipamentos de hardware. A competição se concretiza com a distribuição em massa dos produtos, envolvendo altos custos para criação e lançamento. Dessa forma, companhias líderes investem pesadamente na estratégia de vendas e divulgação da marca.

Entretanto, algumas empresas avançaram no conceito de software de pacote e começaram a fornecer personalização de funções aos seus clientes, o que constitui um diferencial competitivo muito importante para a expansão do *market share*. Contudo, essa estratégia envolve riscos inerentes aos custos e prazos de desenvolvimento das personalizações.

Softwares customizados são programas feitos sob encomenda, normalmente resultam de programas solicitados por usuários. Os usuários

especificam antecipadamente os requisitos gerais e específicos, aparentando mais a execução de serviços que a construção de um produto. Deste modo, a interação entre usuário e fornecedor é intrínseco à fabricação do produto, diferente do que acontece em software de pacote. Dois fatores competitivos importantes nesse segmento são a capacitação da empresa para executar o serviço e um bom relacionamento com os clientes. Os riscos de mercado são menores porque as vendas são realizadas anteriormente. Porém os custos de desenvolvimento são mais significativos.

Em relação aos softwares embarcados, sua característica básica é o fato de que este é comercializado dentro de um equipamento. Atualmente, qualquer equipamento automatizado traz seu software embutido, algo que faz deste segmento um dos mais dinâmicos. Embora o software embarcado tenha sido utilizado inicialmente em equipamentos industriais, atualmente – com o desenvolvimento da microeletrônica e com o advento dos equipamentos automatizados em diferentes áreas (contabilidade, educação, saúde, geofísica e outros) – o número de funções executadas por estes aumentou consideravelmente (RIBEIRO,1998). Existe uma dificuldade de se estimar o valor total da produção deste tipo de software, tendo em vista que seu valor é agregado ao custo do produto manufaturado como um todo.

Agora, considerando-se o tipo de domínio de aplicação, podem-se distinguir duas grandes categorias: horizontal e vertical.

1. Segmento horizontal: Seu conteúdo normalmente é da área de computação, com pequeno contexto específico de outra área de informação. É vendido por meio de pacotes, deve ser flexível, desde que seu objetivo seja resolver problemas básicos de informação nas áreas mais diversas. São exemplos de produtos de segmento horizontais, sistemas operacionais, planilhas eletrônicas, bancos de dados e processadores de textos, entre outros.

Freqüentemente, eles se tornam ferramentas de desenvolvimento de software no segmento vertical. Este segmento apresenta a característica de possuir um ciclo de vida curto devido à dinâmica da informática;

2. Segmento vertical: Os programas são desenvolvidos para uma atividade econômica específica (saúde, educação, geofísica, contabilidade, etc.) e pode ser vendido em pacotes (prateleiras de lojas especializadas) ou, e especialmente, feito sob encomenda. Assim, para o desenvolvimento deste tipo de software é necessária, além do conhecimento dos conceitos computacionais, a utilização de conhecimentos específicos de cada domínio de aplicação. Essa característica determina um ciclo de vida mais longo que o segmento horizontal.

A respeito dessa segmentação, Pondé (1993, p.24) observa:

A proliferação de padrões e a crescente compatibilidade entre sistemas de diferentes fabricantes têm feito com que, no mercado de aplicativos (segmento horizontal), as empresas deixem de competir em segmentos restritos a usuários que utilizam plataformas de hardware específicas. Isto gera um incremento do número de empresas concorrendo em um mesmo espaço de atuação, na medida em que desaparecem algumas fronteiras entre mercados, mas pode levar também a uma maior concentração, pois as possibilidades de conquistar posições de liderança a partir do usufruto de economias de escala se ampliam – especialmente no mercado de pacotes. Na área de software de sistemas (segmento vertical) a tendência é que o mercado se organize em torno de soluções padronizadas dominantes, gerando posições quase monopolistas para as empresas que as ofertam.

Assim, a importância de se construir tipologias para segmentação da indústria se verifica à medida que se consegue identificar padrões de concorrência característicos de cada segmento e também pelo fato de serem percebidos fatores específicos para determinação da competitividade em cada um deles.

Dessa forma, todas estas definições e caracterizações são pertinentes porque, além de facilitar a compreensão da dinâmica tecnológica desta indústria,

elas servem como subsídio instrumental às análises de competição dentro dos diversos segmentos apresentados. Desta maneira, em setores onde mudanças tecnológicas são muito rápidas, existe uma forte relação entre o desempenho da companhia e sua capacidade tecnológica. Como exemplo, considerando o mercado de destino, a empresa poderia seguir dois caminhos para a obtenção de sucesso do seu produto no mercado:

Se quiser ir para o mercado horizontal, deveria entrar com inovação radical ou inovações com incremento, no caso de companhias com monopólio forte no segmento; se quiser entrar no mercado vertical, pode fazer assim com inovações incrementais, mas esses devem deter conhecimentos sólidos da área de aplicação (FRICK; NUNES, 1996).

Atualmente se verifica a atuação de um número pequeno de grandes empresas no segmento horizontal e um elevado número de pequenas empresas no segmento vertical.

Uma análise mais detalhada dessas classificações evidencia que, embora não existam elevadas barreiras para a entrada, com permissão de proliferação de companhias menores, as dificuldades para o crescimento são significativas, uma vez que as corporações de grande porte ocupam segmentos estratégicos e lucrativos. Quer dizer, na organização da indústria de software, ocorre uma co-existência natural entre fragmentação e concentração. Este fato decorre da incapacidade de as grandes empresas aumentarem suas metas de desempenho para tirar proveito das oportunidades existentes, ou pela natureza fragmentada da demanda de uma quantidade enorme de produtos. Há uma configuração dinâmica em que a estrutura da indústria constantemente é remodelada pelo aparecimento de produtos e segmentos novos. Uma intensa concentração é verificada em produtos do segmento horizontal, como planilhas eletrônicas, processadores de textos, bancos de dados e

sistemas operacionais, permitindo a sobrevivência e a reprodução de companhias pequenas e de médio porte no segmento vertical.

Assim, uma forte dinâmica tecnológica é o grande termômetro desta indústria, porque modifica os ciclos de vida dos softwares existentes, ampliando para os usuários a oferta de produtos.

Entretanto, como os produtos de software apresentam um dinamismo tecnológico intenso, em que se combinam reduções nos seus ciclos de vida e a ampliação do leque de alternativas disponíveis para os usuários, verifica-se que as empresas – especialmente as de maior porte – iniciaram movimentos de diversificação e esforços no sentido da ampliação da sua base de capacitações tecnológicas e da busca de canais de distribuição que garantam seu acesso nos mercados emergentes. Neste contexto, são instrumentos destacados de expansão as parcerias tecnológicas, os acordos de *marketing* e distribuição, as fusões e as aquisições.

Pondé (1993) observa que as estratégias das empresas de software são fortemente influenciadas pelo seu porte e o tipo de mercado em que atuam. Percebe-se, por exemplo, que as empresas que detêm a liderança do segmento de software de pacote, atuam explorando de forma agressiva as vantagens proporcionadas pelas economias de escala, rede de vendas, estrutura de suporte abrangente e marca reconhecida, configurando um padrão de concorrência onde o *marketing* assume uma dimensão decisiva. Ademais, as capacitações acumuladas por essas empresas, aliadas ao seu poder financeiro, permitem a diversificação dos serviços em direção à execução de atividades em outros segmentos, proporcionando a entrada em mercados que atingem dimensões econômicas razoáveis e apresentam taxas de crescimento bastante promissoras, utilizando para

isto mecanismos de associações e aquisições de empresas menores. Essa diversificação também inclui a busca de mercados em outros países, conduzindo as empresas à trajetória de um crescente grau de internacionalização.

Ainda segundo esse autor, é importante ressaltar que as características intrínsecas do software enquanto produto, destacadamente o alto peso dos custos fixos na sua produção, combinado com baixos custos marginais, proporcionam à firma que consegue expandir suas vendas um grande potencial de crescimento. Assim, os incrementos nas margens de lucro com custos marginais de produção insignificantes são uma combinação que rapidamente leva à oligopolização dos mercados mais dinâmicos, que acabam evoluindo para um padrão mais concentrado em torno de poucas corporações com sólidas posições tecnológicas, financeiras e de *marketing*, que levaria, em último caso, à saída das empresas marginais do mercado (STEINDL,1983).

Entretanto as empresas que atuam no mercado de software por encomenda, competem segundo suas capacitações e habilidades, o que lhes permite não só chegar a soluções customizadas para resolver problemas específicos dos clientes, como também agregar um montante significativo de serviços – consultoria, treinamento, etc – aos sistemas oferecidos. O projeto de grandes sistemas requer ainda recursos financeiros consideráveis e uma imagem de confiabilidade consolidada, nos quais muitas vezes recorre-se ao deslocamento do risco do projeto para os clientes, uma vez que muitas dessas empresas são de pequeno e médio porte e não dispõem de muitos recursos financeiros.

Em relação às empresas de menor porte, suas estratégias se baseiam em duas vertentes. A primeira consiste na especialização em determinado domínio de aplicação, procurando atender às demandas de um grupo de clientes. Este tipo de

relação tem a característica da manutenção de uma certa dependência mútua: de um lado, o cliente como demandante de serviços e, de outro, o fornecedor de software como conhecedor dos processos da empresa. Para este tipo de estratégia, é necessário que se estabeleçam relações baseadas em vínculos de confiança mútua, ricas em trocas de informações, cooperação e aprendizado interativo. O segundo tipo recorre ao caráter multidimensional dos produtos de software, que é aproveitado para a implementação de uma diferenciação de produto voltada para a ocupação de pequenos espaços deixados pelas empresas líderes. Isso decorre da incapacidade das grandes empresas em atender a todos os segmentos de mercado. Assim, essas estratégias sobrevivem devido à fragmentação característica da indústria de software, bem como pelo fato de que os conhecimentos, técnicas e ferramentas para o desenvolvimento de programas serem relativamente acessíveis para os novos entrantes.

Uma discussão atual sobre as estratégias empresariais é a questão do software livre, que promete efetuar uma reviravolta no jogo competitivo de alguns segmentos. Empresas como a Microsoft, que obteve vantagens competitivas oriundas de informações privilegiadas de seus sistemas operacionais, se vêem agora em uma situação crítica, haja vista a preferência dos usuários pela redução de custo com a aquisição do software livre (é bom lembrar que estes softwares possuem requisitos de qualidade iguais ou melhores que outros de arquiteturas proprietárias).

A Microsoft considerou, em um relatório trimestral entregue à Comissão de Valores Mobiliários (SEC – *Securities and Exchange Commission*) dos Estados Unidos, o movimento de software de código aberto como uma ameaça a seu modelo de negócios. A popularização do movimento de software livre mostra um desafio

significativo ao modelo de negócios de algumas companhias, ante as recentes tentativas dos representantes do modelo de código aberto a convencer governos de todo o mundo a escolher o software livre em suas compras e desenvolvimento de novos aplicativos.

Entretanto, atualmente, as empresas líderes em âmbito internacional estão adotando estratégias bastante agressivas de ocupação do mercado e expansão da base instalada, atuando em três frentes:

1. ampliação dos esforços de *marketing*: efetuando reforço da marca perante os clientes locais e reestruturando as redes de distribuição e suporte;
2. reduções de preço, visando ampliar o *market share*;
3. estabelecimento de parcerias e aquisições de outras empresas do setor (COMPUTERWORLD, 2003).

Analisando-se as informações apresentadas até este ponto e confrontando com a metodologia de estudo de competitividade aplicada ao Estudo de Competitividade da Indústria Brasileira (COUTINHO; FERRAZ, 1994), percebem-se quatro estratégias competitivas distintas na indústria de software:

- ▶ Custos: verificam-se economias de escopo provenientes de utilização dos mesmos canais de distribuição e reestruturação de processos organizacionais, além da obtenção de economia de escala e redução do preço final ao cliente com a utilização de ferramentas automatizadas. Este padrão, embora característico de segmentos mais comoditizados (pacotes), consegue ser percebido nos diversos segmentos da indústria de software, inclusive naqueles que se mostram mais diferenciáveis.
- ▶ Diferenciação: caracterizada pelo fornecimento de serviços personalizados e pela disponibilidade de funções específicas, bem como pela construção de rede de suporte a usuários e pelo aumento dos padrões de qualidade. É normalmente verificado no

segmento de softwares diferenciados, podendo também ser praticado por pequenas empresas no que tange à utilização de capacitações acumuladas para atendimento às demandas específicas de um determinado grupo de clientes;

- ▶ Inovação: o potencial financeiro se faz essencial para este padrão de concorrência. As empresas necessitam despende esforços para capacitação em P&D, tanto sobre o aspecto físico, quanto humano, sendo então característico de empresas de maior porte;
- ▶ *Responsiviness*: este padrão possui maior aderência às estratégias praticadas por empresas de menor porte do segmento de encomenda, haja vista a especificidade de demanda de alguns clientes, que exige uma ampla flexibilidade produtiva das empresas.

Assim, a análise das informações apresentadas até este ponto permite construir uma breve conclusão sobre alguns conceitos apresentados nesta seção, na qual se abordou a concepção das estratégias competitivas, as implicações dos padrões de concorrência prevalentes na indústria e os fatores determinantes da competitividade.

Em relação às estratégias competitivas, observa-se que elas representam as ações das empresas na busca de um posicionamento que lhes forneça a vantagem competitiva necessária a seu crescimento ou a sua manutenção de forma sustentável. Para tanto, a estratégia competitiva é influenciada por uma série de fatores internos e externos à empresa, que abrangem o volume de capacitações acumuladas, o potencial financeiro, a estrutura da indústria, o posicionamento da empresa no mercado, os concorrentes, as condições da demanda, entre outros.

Já o padrão de concorrência envolveria o conjunto de estratégias aplicadas em cada espaço de competição (mercado ou indústria, região, nação) e se revela através de um aglomerado de formas de concorrência que são dominantes

nesses espaços, variando conforme as transformações tecnológicas, organizacionais, industriais e econômicas.

4 CONCLUSÕES

Atualmente, coexiste no mercado uma grande diversidade de estratégias em meio a um número relativamente pequeno de firmas de grande porte que prestam serviços de informática a grandes clientes e a uma maior quantidade de médias e pequenas empresas que seguem, predominantemente, estratégias de sobrevivência, buscando nichos relativamente protegidos da competência de seus concorrentes de maior porte e, em geral, atendem a usuários locais de todos os setores e tamanhos.

As maiores empresas produtoras de software são as que têm menor dificuldade de acesso aos profissionais qualificados, possuem mais recursos para implementar os sistemas de qualidade, utilizam ferramentas mais complexas e têm as menores dificuldades para a obtenção de financiamento. Além disso, ainda existe o caso de firmas estrangeiras que contam com o apoio e o respaldo de suas matrizes em termos não só financeiros, como tecnológicos e comerciais.

De acordo com os padrões de concorrência predominantes na indústria de software, é possível identificar alguns aspectos que materializam ou dificultam a adoção das estratégias competitivas.

A estratégia de inovação é uma importante fonte de vantagem competitiva que pode ser adotada em todos os segmentos da indústria de software.

Já a estratégia de *responsiviness*, como envolve aspectos relacionados à flexibilidade produtiva, teria maior aderência ao segmento de software por

encomenda, pois neste as questões relacionadas à rapidez e a habilidade de servir o mercado se mostram essenciais. Essa estratégia é pouco aderente ao segmento de pacotes, pelas características da padronização de serviços e da forma de comercialização.

Em relação à estratégia de diferenciação, geralmente levam-se em consideração os aspectos referentes à qualidade, à imagem e à confiabilidade das empresas. Dessa forma, esta estratégia parece perfeitamente aplicável a qualquer segmento da indústria, através da qual poderiam ser alcançados tanto benefícios tangíveis, como manutenção de padrões, quanto intangíveis, relacionados à consolidação da marca e satisfação dos clientes, entre outros.

A estratégia de custos é outra fonte de vantagem competitiva que pode ser empregada pelas empresas em todos os segmentos da indústria de software. Entretanto, como será discutido adiante, alguns cuidados devem ser considerados para a sua adoção.

Agregam-se nessa estratégia os indicadores relativos à estrutura de custos, à produtividade, à contratação de mão-de-obra e às práticas de engenharia de software para a obtenção de economias de escala.

Neste ponto, cabe uma breve discussão sobre a adoção concomitante das estratégias de custo e diferenciação.

É interessante observar que a competição por custo na indústria de software se mostra um pouco paradoxal quanto à estratégia de diferenciação, pois se a atividade de se construir software é bastante intensiva em conhecimento e aspectos ligados à qualidade do software são fatores essenciais para o sucesso dos projetos, então, seguindo o pensamento de Porter (1989) sobre a extrema dificuldade em se manter estratégias concomitantes de baixo custo e diferenciação

por qualidade, seria possível utilizar estratégias de baixo custo na indústria de software?

Possível é, no entanto só através de soluções que contemplem a obtenção de economias de produção, seja com inovações tecnológicas, seja através da reutilização de códigos, mas qualquer outra solução que venha a prejudicar a qualidade dos projetos parece desastrosa para esta questão. Na prática, a indústria local já acumula alguns casos de fracasso justamente quando se tentou reduzir a capacitação da mão-de-obra para a obtenção de vantagem de custo.

Convém citar, que a perspectiva para o desenvolvimento do setor abrange uma série de fatores relacionados à competitividade sistêmica que exercem interferência direta na perspectiva de elaboração e execução das estratégias. Portanto, o desenvolvimento da indústria, é marcado e dificultado enormemente pelas limitações impostas pela situação do conjunto de fatores de competitividade do setor de software.

Enfim, a competitividade é, então, o resultado da adequação das estratégias competitivas adotadas pela empresa ao padrão de concorrência prevalecente em cada mercado ou segmento. O padrão de concorrência é a variável independente enquanto a competitividade é a variável dependente. O elemento básico de análise da competitividade é a empresa, considerada como um espaço de planejamento e organização da produção que se estrutura em torno de suas capacitações. Contudo a competitividade está relacionada não somente às questões internas à firma, como também à capacidade dos governos, ao comportamento da sociedade e aos recursos disponíveis, exigindo, portanto, uma abordagem sistêmica para o seu completo entendimento. Assim, num âmbito mais amplo, a competitividade de indústrias ou firmas depende de um ambiente social, política e

economicamente estável, de instituições eficientes, de sistemas educacionais adequados e de políticas industriais que favoreçam o desenvolvimento de forma sustentável.

REFERÊNCIAS

- BRANCO, C. E. C.; MELO P. R. de S. **Proposta de atuação para o BNDES no setor de software**. Disponível em: <<http://www.bndes.gov.br/conhecimento/bnset/sofset02.pdf>>. Acesso em: 10 fev 2003.
- COMPUTERWORLD. **Oracle permanece na liderança de mercado de DBMS**. Disponível em: <http://computerworld.terra.com.br/templ_textos/noticias.asp?id=11636>. Acesso em: 19 fev. 2003.
- FRICK, S.; NUNES, R. **Produtos, estruturas de mercado e estratégias competitivas no setor de software**. *Economia & Empresa*, v.3, n.1, p.34-44, 1996.
- GAIO, F. **The development of computer software technological capabilities in developing countries: a case study of Brazil**. 1990. Thesis (Doctor of Philosophy)-University of Sussex, April 1990.
- Graeml, Alexandre R. **Sistemas de informação: O alinhamento da estratégia de TI com a estratégia competitiva**. São Paulo: Atlas, 2003.
- HEEKS, R. **India's software industry: state policy, liberalisation and industrial development**. New Delhi: Sage Publications, 1996.
- KUPFER, D. **Made in Mercosur: avaliação de competitividade visando à promoção da integração regional**. Rio de Janeiro: GIC-IE/UFRJ, 2000.
- PONDÉ, J.L. **Competitividade da Indústria de Software: nota técnica setorial do complexo eletrônico – ECIB**. São Paulo: Ministério da Ciência e Tecnologia, 1993.
- PORTER, M.E. **A vantagem competitiva das nações**. Rio de Janeiro: Campus, 1989.
- RIBEIRO, A. **A indústria brasileira de software: qualidade como um fator de competitividade**. 1998. Dissertação (Mestrado em Economia) – IE-Unicamp, Campinas, 1998.
- STEINDL, J. **Maturidade e estagnação do capitalismo americano**. São Paulo: Abril Cultural, 1983.
- TEIXEIRA, F.; GUERRA, O. **Estratégia para o desenvolvimento da indústria na Região Metropolitana de Salvador**. In: AVENA, Armando (Org.). *Bahia Século XXI*. Salvador: SEPLANTEC, 2002. p. 149-204.