Transformation from the Information Age to the Conceptual Age:

Impact on Outsourcing

A.B. Patki
Department of Information Technology, Government of India, New Delhi, India

Tapasya Patki
Nexus of Entrepreneurship and Technology, University of Arizona, Tucson, USA

Mahesh Kulkarni
Center for Development of Advanced Computing, Pune, India
Abstract

The previous decades have seen the emergence of the Information Age, where the key focus was on knowledge acquisition and application. With the emergence of cross-domain disciplines like outsourcing, we are witnessing a trend towards creative knowledge, rational application, and innovation. We are now progressing from an era that was information-dependent towards the era that revolves around “concept” development. This age, referred to as the Conceptual Age, will be dominated by six new senses -- Design, Story, Symphony, Empathy, Play and Meaning--. creating a need to diverge from the current reliance on linear and sequential algorithmic practices in outsourcing and to adopt cognition based engineering and management approaches. This paper lays the foundation for Offshore Engineering and Management (OEM) and discusses estimation issues in OEM that have their roots in software engineering. Also, this paper identifies the limitations of the current methodologies from an outsourcing point of view, and delineates how they can be deployed effectively for an outsourced environment.

Keywords: Information Age, Conceptual Age, Outsourcing, Outsourcing Engineering and Management (OEM), Cost Estimation, Evidence Based Software Engineering,

INTRODUCTION

Professional outsourcing relies greatly on the attitude, innovation, and creative instincts of the taskforce involved. Pink (Pink, 2005) highlights the transformation of the society from the Information Age to the Conceptual Age from the psychological point of view, and emphasizes that six new senses will play an important role in the Conceptual Age. While Design, Play, and Meaning will be the prime senses for corporate outsourcing, Story, Symphony, and Empathy will be the potential senses for personal offshoring
(Gamerman, 2007). The development of the working components of the six parameters of Conceptual Age will occur by cultivating the skill set and education (Johnson, 2006). Corporate outsourcing will be driven by these conceptual components. The need for re-orienting education has been justified for future success (Greeespan, 2004). The criteria for economic success and increased productivity have been broadly classified as creativity, artistry cultural diversity, and technical experience. Oversupply, outsourcing, and automation are perceived to be the defining characteristics of the evolving state of economy (Wikipedia).

From the outsourcing perspective, software-intensive systems will play a significant role in a variety of projects, creating the need for a new strategy to improve the dependability and trustworthiness of the software. This motivates the creation of Offshore Engineering and Management (OEM) as an emerging discipline. The primary forces driving the emerging OEM trends are as follows:

(i) The competition between the left-brain and the right brain will pose new problems of demands, supply, and satisfaction in the conceptual age (Pink, 2005). This will lead to greater emphasis on cognitive aspects of information processing and less dependence on routine conventional data processing areas as manifested in the outsourcing activities of the current decade, like call centers, medical transcriptions, and claims processing.

(ii) Personal offshoring will boost the e-Service sector to meet the demand for new variants of existing products and services. Corporate outsourcing will be directed towards mass-scale and bulk capacity products through reorientation
of knowledge workers (Lumb, 2007). This trend is imposed by the era of abundance and is mandated for the co-existence and survival of companies.

(iii) The phenomenon of software aging is important in the context of cognitive support for outsourcing in the conceptual age. The detection of the onset of software aging can help to prevent dynamic failure events. Multivariate State Estimation Techniques (MSET) have been investigated for real-time proactive detection of software engineering mechanisms in operating environments involving multiple CPU servers (Gross et al, 2002)

OFFSHORE ENGINEERING AND MANAGEMENT (OEM)

OEM proposes the systematic and structured application of scientific, engineering, and management principles, through the use of proactive software engineering and information technology approaches, in the business process outsourcing arena. Proactive software engineering can be defined as a framework that extends the scope of conventional software engineering by incorporating additional concepts of fault tolerance, graceful degradation, software aging, adaptability, usefulness of software (pre/post-development) documentation, user manuals, and measure of module level Machine Intelligence Quotient (MIQ) (Patki 2006; Patki and Patki, 2007). MIQ is a measure of autonomy and performance for unanticipated events and links the infrastructural needs of an outsourcing institution with its throughput. MIQ differs significantly from other indices like control performance, reliability, and fault-diagnosis (Park et al, 2001).
The use of a rule oriented approach for workflow control and design consistency checking has been illustrated in DPSSEE (Deng et al, 2003). The approach of semantics of software project for perception and cognition is broadened to introduce logic rules to all levels of the software life cycle. DPSSEE has limitations while addressing outsourcing projects. In the past, software engineering concentrated on the analytical philosophy and rarely addressed the issue from the viewpoint of design synthesis. Typically, problems were framed in terms of being algorithmically solvable instead of being intuitionally developed. For example, we have been using man-months as a measure to estimate the software development effort, neglecting to look at issues of manpower strength and profile of a software development house. It is still difficult to do systematic analysis to determine which software development project can execute effectively and which cannot, even with deployment of additional manpower.

First generation outsourcing efforts, such as ones related to call centers and medical transcriptions, belonged to the automation class (algorithmic, semi-structured or reasonably structured); they are primarily right brain oriented - logical, algorithmic, and analytical. These tasks were adaptable and could be effectively handled by web-based approach. The e-service oriented outsourcing in the conceptual age will be heavily cognitive informatics loaded. Cognitive Informatics (Wang 2007, Pascal et al, 2005) is motivating the move from the physical layer to the abstract level along the lines of the artificial mind system (Hoya, 2005). This leads to a corresponding shift in computer science with greater emphasis being placed on logical, data centric, and algorithmic styles, as well as on language and thinking modules. Non-linear, intuitive, and holistic
activities will gain importance, in place of sequential, logical, and analytical activities (Pink, 2005; Lumb, 2007).

In the information age, the prevailing culture was to evolve business avenues by deploying computer systems and establishing Information Technology (IT) practices. Carroll (Carroll, 2007) identifies the concerns of engineering executives in outsourcing decisions and discusses how costs can be driven out of engineering analysis and decisions. After the board of directors opts for outsourcing, it is the engineering executives who have to ensure that the outsourcing partner will actually deliver what is promised. The situation is complicated, as engineering executives must control work quality, cost, and schedule in a situation that they cannot directly see. In the absence of adequate theory and practice of OEM, engineering executives rely on trust. Such issues cannot be satisfactorily handled with conventional software engineering practices or thru legal procedures. The current methodology of directly extending traditional software engineering techniques, such as the Capability Maturity Model, to outsourced information technology projects, suffers from several drawbacks, as highlighted later in this paper.

**EXTENDING SOFTWARE ENGINEERING PRINCIPLES TO OEM**

Offshore development has, so far, received inadequate attention in the software engineering literature (Meyer, 2006). The establishment of a large, complex BPO system that relies on a multiplexed architecture (Patki and Patki, 2007) calls for expertise from many different domains of engineering and requires a diverse, distributed team of people
working simultaneously to ensure a stable operation. Such a team needs to be configured to support two distinct types of activities: (i) sequential, logical, and analytical activities; and (ii) non-linear, intuitive, and holistic activities. The existing workforce must be reoriented to prepare for the cognitive challenges of the conceptual age outsourcing, in place of the watertight compartment philosophy of the past (Robert, 2006, Pascal et al, 2005). In the subsequent sections, we discuss the limitations of existing software effort approaches in the context of outsourced projects.

**Software Effort Estimation**

The estimation of resources, cost, and schedule for a project requires experience, access to good historical information, and the courage to make quantitative predictions when qualitative information is all that exists. Cost models have a primary cost factor, such as size, and a number of secondary adjustment factors or *cost drivers*. Cost drivers embody characteristics of the project, process, products, and resources that influence effort. Boehm derived a cost model called COCOMO (Constructive Cost Model) using data from a large set of projects at TRW. COCOMO is based on inputs relating to the size of the system and a number of cost drivers that affect productivity (Pressman, 2001; Sommerville, 2000); the model takes the form:

$$E = aS^b \times EAF$$

where $E$ is effort in person months, $S$ is size measured in thousands of lines of code (KLOC), and $EAF$ is an effort adjustment factor (equal to 1 in the Basic model). The factors $a$ and $b$ depend on the development mode.

The Basic COCOMO model computes effort ($E$) as a function of program size.

$$E = a \times (KLOC)^b$$
The Intermediate COCOMO model computes **effort (E)** as a function of program size and a set of cost drivers.

\[ E = a (KLOC)^b \times EAF \]

The effort adjustment factor (EAF) is calculated using 15 cost drivers. The cost drivers are grouped into four categories: *product, computer, personnel, and project*. The product of all effort multipliers is the EAF. The factors \( a \) and \( b \) used in Basic and Intermediate versions of COCOMO are numeric quantities (Aggarwal and Singh, 2005).

The Advanced COCOMO model computes effort as a function of program size and a set of cost drivers weighted according to each phase of the software lifecycle. The advanced model applies the intermediate model at the component level, and then a phase-based approach is used to consolidate the estimate.

The four phases used in the detailed COCOMO model are: requirements planning and product design (RPD); detailed design (DD); code and unit test (CUT); and integration and test (IT). Each cost driver is broken down by phase as in the example shown in Table 1.

<table>
<thead>
<tr>
<th>Cost Driver</th>
<th>Rating</th>
<th>RPD</th>
<th>DD</th>
<th>CUT</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACAP</td>
<td>Very Low</td>
<td>1.80</td>
<td>1.35</td>
<td>1.35</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td>Nominal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>0.75</td>
<td>0.90</td>
<td>0.90</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>Very High</td>
<td>0.55</td>
<td>0.75</td>
<td>0.75</td>
<td>0.70</td>
</tr>
</tbody>
</table>

**Table 1 Analyst capability effort multiplier for Detailed COCOMO**

The use of COCOMO for analyzing outsourcing projects is hampered by two factors:

(i) As opposed to conventional software development, the BPO environment calls for distribution of work force and infrastructure;

(ii) Some of the characteristics of the outsourced environment and the BPO methodology are inherently non-linear in nature.
Since the current set of cost drivers of COCOMO model does not cater to these non-linear factors, one approach would be to add a new set of parameters that are related to offshore development and deployment of software, including infrastructure and country specific parameters like Cyber Crime Index (CCI), Customer Satisfaction Rate (CSR) and Civil Infrastructure Status (CIS) (Patki and Patki, 2007). Table 2 characterizes and compares existing COCOMO model and the additional features needed for the OEM environment.

<table>
<thead>
<tr>
<th>Discerning Factor</th>
<th>Information Centric (COCOMO)</th>
<th>Concept Centric (PROPOSED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modus Operandi</td>
<td>Sequential, Linear</td>
<td>Simultaneous, Non-Linear</td>
</tr>
<tr>
<td>Parameters Considered</td>
<td>Cost Drivers, which are primarily static (numeric)</td>
<td>CIS, CSR, and CCI which are dynamic (vector)</td>
</tr>
<tr>
<td>Concluding Methodology</td>
<td>Computational</td>
<td>Inferential</td>
</tr>
<tr>
<td>Resources Utilized</td>
<td>Databases, Files</td>
<td>Knowledge Base</td>
</tr>
<tr>
<td>Retrieval Algorithm</td>
<td>User Query Oriented</td>
<td>Pattern Oriented</td>
</tr>
</tbody>
</table>

**Table 2. Comparison of Information Centric and Concept Centric Approach**

One study shows that linear scaling of existing COCOMO for outsourced effort estimation models may not be an effective solution (Patki 2006).

**Capability Maturity Model**

Some observers attribute the success of outsourcing projects to the Capability Maturity Model (CMM). CMM is a mixture of process and organization assessment schemes. The Software Engineering Institute (SEI) at Carnegie Mellon University conceived the CMM certification methodology, under the aegis of funding from the US government. However, recent studies reported by Kitchenham (Kitchenham et al, 2007) have questioned the myth of the direct relationship between CMM and the success of software projects. The Capability Maturity Level is a measure of the quality software development ability as per time schedule. However, it is erroneous to assume that all Level 2 organizations develop
better software than Level 1 organizations. Using corporate project data, it has been
demonstrated that software engineering metrics can be misleading and that software measures
are often not as useful as practitioners hope. The cited study concludes that it is
inadvisable to characterize productivity, either for monitoring or prediction purposes,
using mean and standard deviation techniques; further, one should avoid making any
estimates.

Pfleeger (Pfleeger et al, 1997) reports problems from the use of software metrics without
keeping the development goals in mind; they also discuss the applicability of CMM. The
role of third party (ESSI consortium) assessment is considered in (Kitchenham et al,
1997b). The applicability of CMM in outsourced environment is further eroded when one
analyzes issues related to the product line concept for outsourcing (Patki et al, 2006). A
shared set of software assets, including the software architecture, is the basis for the
product line. The potential customer association for product line could be along the lines
of Microsoft’s Government Security Program where trust is established as a policy and
principle of business. So far, such an integrated approach of customer association and
product line methodology has not been effectively deployed in the context of outsourcing.
Software product families can be considered to be related products that make extensive
reuse of available components. The dynamic object roles are useful, both for conceptual
modeling and implementation. The links among the objects and roles are used for
conceptual modeling of business applications (Subieta et al, 2003). While extending the
scope of outsourcing to production of such product family, the need has arisen to
augment the existing software engineering theory (Desouza, 2003; Keung et al, 2004).
EVIDENCE BASED SOFTWARE ENGINEERING

Evidence Based Software Engineering (EBSE) is based on the premise that software engineering will advance by moving away from its current dependence on advocacy and analysis. The scope of the current cost models that primarily focus on activities of a single system has been expanded in the context of System-of-Systems (SoS) development by researchers like (Lane and Valerdi, 2007). The limitations of using axioms in software metrics are discussed in the context of incorrect assessment of the validity of some software measures by (Kitchenham and Stell, 1997a). EBSE lays emphasis on empirical based approaches for decision making and for evolving new practices (Kitchenham et al, 2004). The five steps involved in EBSE (Dyba et al, 2005) approach utilize a continuously updated knowledge based mechanism. EBSE techniques for identifying, producing, combining and accumulating evidence, including metadata, are directly relevant for outsourcing (Jorgensen et al, 2005). In order to increase its relevance further, EBSE needs to be expanded to incorporate fuzzy logic inference techniques. Parameters like Cyber Crime Index (CCI), Customer Satisfaction Rate (CSR) and Civil Infrastructure Status (CIS) help to determine the trustworthiness of a particular unit, and the progress it has made towards attaining its targets. Independent agencies that employ EBSE principles should be setup to provide portals for assessment and periodic updates for these parameters, and the workforce should be retrained for such new jobs (Huit, 2007; Pascal et al, 2005).
Unlike Boehm’s cost driver multipliers for product, computer, personnel, and other project attributes that are more static in nature, the EBSE methodology must provide for continuously updated values for CCI, CIS, and CSR, using inference-oriented models. The five-step approach of EBSE can be applied here (Dyba et al. 2005, Patki 2006). Overall, we propose the use of a EBSE approach for CCI, CIS, CSR, and other evolving parameters, in place of COCOMO, CMM, and other conventional models. The factors like software aging mechanism (Gross et al, 2002) must be integral part of the software effort estimation process.

**EBSE APPROACH FOR OUTSOURCING CONTRACTS**

Outsourcing contracts are developed for legal assurance and litigation based settlements. Reifer (Reifer, 2004) highlights that it is necessary to avoid legal *conflict* through contract administrations. The contractual relationship is a subjective interpretation of individual contract clauses, as text comprising of vocabulary and grammar in the psychological sense. In this context, fulfilling contractual obligations is a necessary but not sufficient condition for outsourcing projects, especially in mission-critical situations using COTS methodology with MIQ features (Patki and Patki, 2007). In the past, the study of outsourcing legal contracts had been focused on the customer perspective (Koh, Ang & Straub, 2004). We need to extend the scope of these contracts to include design, empathy, and meaning (Pink, 2005) as psychological dimensions, in order to establish context and trust in the entire outsourcing system. Table 3 highlights the major differences.

<table>
<thead>
<tr>
<th>Description</th>
<th>Conventional Contract</th>
<th>EBSE Based Contract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Text, using standard vocabulary a</td>
<td>Context and Trust, using mapping</td>
</tr>
<tr>
<td>Basic Units</td>
<td>nd grammar</td>
<td>techniques after evidence collection</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Review</td>
<td>Clauses (for analysis)</td>
<td>Mid Term Review in fault-detection mode, projects unacceptability</td>
</tr>
<tr>
<td></td>
<td>Information (for synthesis)</td>
<td>Mid Term Review in fault-correction mode, projects acceptability and supports furtherance</td>
</tr>
</tbody>
</table>

**Table 3. Significant Factors for EBSE based contract**

The principles of Kansei engineering can be extended to drafting contractual documents for outsourcing applications. Kansei engineering refers to the translation of consumers' psychological feeling about a product (like a digital contract) into perceptual design elements (Coleman et al, 2005). Kansei engineering is a step towards supporting the six-sense conceptual age philosophy in the outsourcing domain; significant scope exists for integrating the Kansai techniques with EBSE methods.

**CONCLUSIONS**

In the near future, outsourcing will promote innovation, shorten the research and development cycle, and prevent the deployment of over-engineered architectures. The growing use of proactive software engineering techniques in the outsourcing domain will lead to greater use of Evidence Based Software Techniques (EBST) and the new discipline of Offshore Engineering and Management (OEM), signifying the move from the existing information centric approach to a new concept-centric methodology that emphasizes cognitive content. These are in the areas of providing cognitive support using proactive software engineering around EBSE methodology to integrate seamlessly the operations of outsourcing industry.

**REFERENCES**


