Implications of Nearshore Development in the Mexican Software Industry

by

Alexandro Artola

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and Master of Engineering in Electrical Engineering and Computer Science at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Author____

Department of Electrical Engineering and Computer Science
May 21, 2003

Certified by________

Michael J. Piore
Thesis Supervisor

Accepted by________

Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Implications of Nearshore Development in the Mexican Software Industry

by

Alexandro Artola

Submitted to the
Department of Electrical Engineering and Computer Science

May 21, 2003

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Over the past ten years, many US companies have began to outsource software projects abroad, offering new employment and income opportunities to developing countries. This thesis examines in detail the organization of work at a company providing offshore services. This thesis looks at two types of projects and for each project shows the distribution of employees by occupational level and how this level varies as the project evolves over time. This thesis links the occupations to the processes of education and training through which the labor force required to fill the jobs is produced. Finally, this thesis uses these results to estimate a growth scenario. This thesis concludes that the growth of the offshore industry in Mexico is unlikely to be constrained by the availability of qualified labor. Moreover, the quality of the jobs (and their wage rates) will be significantly better than that of jobs in declining industries, but the number of jobs created will be unable to match the number that have been lost in declining industries.

Thesis Supervisor: Michael J. Piore
Title: David W. Skinner Professor of Political Economy

Acknowledgements

First, I would like to thank Professor Michael Piore, my thesis advisor, for his guidance and support for the past two years.

I would also like to thank my thesis readers, Professor Clemente Ruiz and Professor Andrew Shrank, for their interest and on-going research in the field.

Moreover, I would like to thank all the people at "Azteca Software," without whom this thesis would not be possible.

Finally, I would like to thank my family for their care and support.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The *maquila* industries in garments and electronics in Mexico are collapsing (Sargent and Matthews, 2002; Smith, 2002). The plants that once went from the United States to Mexico are now leaving Mexico to Central America and Asia in search of higher wage differentials. Mexico needs new industries to replace them. One possibility is the software industry. The basis for this possibility is that other developing countries have managed to launch successful software industries (Arora et al., 2001).

Similar to the maquila industries, the Mexican Software Industry can develop by offering offshore development services to US companies. Mexican companies can take advantage of their geographical proximity to US customers and engage in offshore services often referred to as "nearshore development." The offshore industry, however, does not depend exclusively on the wage differentials, but on the change on the organization of work required to take advantage of these wage differentials.

This thesis examines the organization of work at a nearshore development center. This center is part of a Mexican software company, which I call "Azteca Software." To perform this study, I visited Azteca Software for a week during January 2003 and conducted twelve formal interviews with ten employees across different roles (see Appendix A). To support the interviews, I make use of internal documents, company news, and informal conversations with employees at Azteca Software. Based on this study, this thesis then presents a growth scenario for nearshore development and its implications in the Mexican software industry.

The research done in this thesis is part of a larger project on globalization being conducted by a group of faculty from the Economics, Political Science, and Urban Studies departments at the Massachusetts Institute of Technology. In addition, there was collaboration with other faculty and students from the *Universidad Nacional Autónoma de México* and Yale University.

The interest in the software industry is that it embodies technologies that can be applied to many economic sectors. In addition, the software industry represents the advanced technology of the "new economy," which offers an apparent contrast to labor-intensive industries (i.e., maquila) that have moved from the US to Mexico. This project, however, calls into question the conventional view of globalization, in which mature industries move to the developing world, while the advanced industrial countries concentrate on high technologies. The reason behind this is that developing countries, as mentioned above, have been able to launch successful software industries.

This thesis is structured as follows: In the next chapter, offshore development is introduced. In addition, the demand for offshore development in the US market and the position of Mexico in offering offshore services are described. Chapter 3 details the organization of work at Azteca Software for two types of projects: Applications Development and Applications Maintenance & Support. Chapter 4 builds on the previous chapter and presents a growth scenario for nearshore development in Mexico. Moreover, this chapter describes the implications of this growth scenario in the Mexican Software Industry. Finally, Chapter 5 presents the main conclusions of this thesis and some future work suggestions.

Chapter 2: Background

This chapter lays out what offshore software development is, and the position of Mexico in the offshore development market. Before discussing offshore development, the first section describes two general approaches to software development. The origins and reasons for offshore development are illustrated in the second section. The third section summarizes the offshore development market in the US. Finally, the fourth section points out Mexico's position in offshore development, and overviews the Mexican government's effort for the development of the software industry.

2.1 Software Development

There are two main views on how to approach software development: software craftsmanship and software engineering (Eischen, 2003). The craft-based approach has been used since the start of software development. This approach has been addressed by Frederick Brooks in *The Mythical Man-Month,* where he states the efforts for rationalizing the approach to increase productivity – higher quality software with lower costs (Brooks, 1995). Some argue that software engineering is such outcome.

Other authors, such as Peter McBreen, state that this rationalization of software development is not even possible (McBren, 2002). McBreen's main argument is that the automation of processes is geared towards mechanical tasks and not creative tasks, such as software development. Nevertheless, McBreen supports the use of processes, as used in the engineering approach, for large-scale defense projects; and the craft-based approach for commercial development. In addition, other efforts have supported the craft-based approach by stating the importance of developers over processes (Agile Manifesto, 2003).

The engineering approach, on the other hand, was introduced in a North Atlantic Treaty Organization (NATO) conference to tackle a software crisis; the term "software engineering" was born (Nauer and Randell, 1969). This software crisis was caused by the complexity of the software being built and the need for skilled developers in developing large-scale defense systems.

This approach to software development was then supported by the creation of the Software Engineering Institute (SEI) in 1984 by the US Department of Defense. The reason for creating the institute was so that the US government could acquire quality "software-intensive systems" from a broader contractor base. This goal has been shifting over the years towards making improvements in overall software engineering capabilities (SEI, 2003). One example is the Software Capability Maturity Model (SW-CMM), where the SEI has outlined the 'best' software management practices (Paulk et al., 2003). In this model, software companies can be assessed in five different levels according to their processes maturity: initial, repeatable, defined, managed, and optimizing. Used in over 5,000 software organizations, this model is considered as the norm among the industry for measuring a company's quality in software development practices.

Although there are differences between the two approaches, the engineering approach has been widely adopted for commercial software development – especially offshore development. For example, most of the software companies that are assessed SW-CMM Level 5 are from India. To understand why this has been the case, the next sections describe these two approaches.

2.1.1 Software Craftsmanship[1]

In this approach, software development depends on high-skilled developers, which are referred to as software craftsmen. To become a software craftsman, experience in developing successful software is required. Search of these craftsmen are based on their reputation, and the best examples of this approach are the Open Source and Linux communities.

McBreen argues that the software craftsmanship model should be used in short and medium projects with development time of fewer than 100 man-years. These projects are for commercial development and should not be critical systems; in other words, projects that are under a tight budget and schedule. Moreover, the hardware infrastructure should be known and not developed at the same time as the software.

The three roles present in software craftsmanship are software craftsmen, journeymen developers, and apprentice developers. In this framework, a small team of craftsmen tackles the whole project, including the architecture, analysis, design, coding, and testing. While working on this project, craftsmen get support from a few journeymen and apprentices who are mainly being trained and mentored to become craftsman. An important requirement for this team is that they have worked together in the past, and that

[1] Most of the arguments presented in this section are based on Brooks (1995) and McBreen (2002).

there is respect towards master craftsmen. The time for an apprentice to master the craftsmanship can last up to twenty years.

In software craftsmanship, the project depends on the craftsman. As a result, developers are more valuable than their managers, and should be paid from $150,000 to $250,000, whereas college graduates are currently overpaid with starting salaries at $40,000. That is, instead of having 30 average programmers, a project may have three master craftsmen who are paid ten times as much as average programmers.

2.1.2 Software Engineering

In the software engineering approach, the point of view is transferred from the developers to the use processes. This use of processes enables a division of labor, which claims to make software development more efficient. Moreover, the use of processes makes managers, and not developers, control the software being built (Eischen, 2002).

Software engineering was introduced for large-scale defense projects. McBreen suggests that these projects should at least be 100 man-years long and that there should not be budget constraints because of their criticality. Moreover, the hardware infrastructure might change in these projects. This change causes the software team to wait and design the system in detail until the hardware is ready.

The four roles present in software engineering are analysts, designers, programmers, and testers. These roles correspond to the four main stages that are usually present in the engineering model: analysis, design, programming, and testing. Moreover, this approach enables specialization and the interchangeability of developers within their roles.

Since software development depends on the processes, there is the idea that skilled developers can be replaced with average developers. This idea has led some authors to use the notion of "Software Factories" (Cusumano, 1991). Because companies in the software engineering approach depend on processes, the SW-CMM has been used to differentiate high quality companies from the rest.

2.2 Offshore Development

As multinational companies have begun to cut costs in software development, these companies have outsourced these projects to software companies. The division of labor achieved with these two types of companies has lead to more productivity. Continuing with this trend of outsourcing, in the past ten years these multinational companies have began to outsource their software projects to companies abroad – offshore (software) development. These types of offshore services have their roots in the beginning of the 1970's, where manufacturing companies around the globe began to move parts of their operations to places with lower labor costs such as Mexico, the Philippines, and Puerto Rico (Amoribieta et al., 2001).

Offshore development is mostly visible in two segments of the software industry: applications and maintenance & support. However, offshore development has also showed up in other segments of the industry such as development tools (i.e., programming languages), infrastructure software (i.e., operating systems), and embedded software (i.e., software in cell phones and other appliances). Regardless of the software development approach that these software companies use, these companies have been measured by their use of processes as outlined in the SW-CMM.

There are two main reasons for offshore development. First, similar to decrease costs as manufacturing companies did in the 1970's, companies have outsourced their software projects abroad to save time and money; These projects are usually sent to developing countries. This increase in productivity is due to the wage differentials between the acquiring country (i.e., US) and the offshore locations. For example, software developers rates in India can be 40 to 60 percent below those in the US. Another important factor to consider is that labor accounts for more than 75 percent of the cost of developing software, which makes offshore development an attractive option within countries with low labor costs (Amoribieta et al., 2001). As a result, after taking into account management, travel, telecommunications, and other costs, companies can achieve savings of around 50 to 60 percent by outsourcing their software projects to India (Sharma et al., 2000).

The second reason for offshore development has been the shortage of skilled software developers in the US. One indication of this shortage has been the increase in use of temporary visas for US companies employing foreign workers in computer-related occupations. For example, more than 50 percent of the 257,640 H-1B visas approved in the year 2000 were for workers in computer occupations (INS, 2002). Moreover, according to International Data Corporation (IDC), the required workforce for Information Technology (IT) workers, including software developers, in the US surpassed the available workforce in 1996 and has been increasing ever since. Figure 1 shows this hiring gap for IT workers.

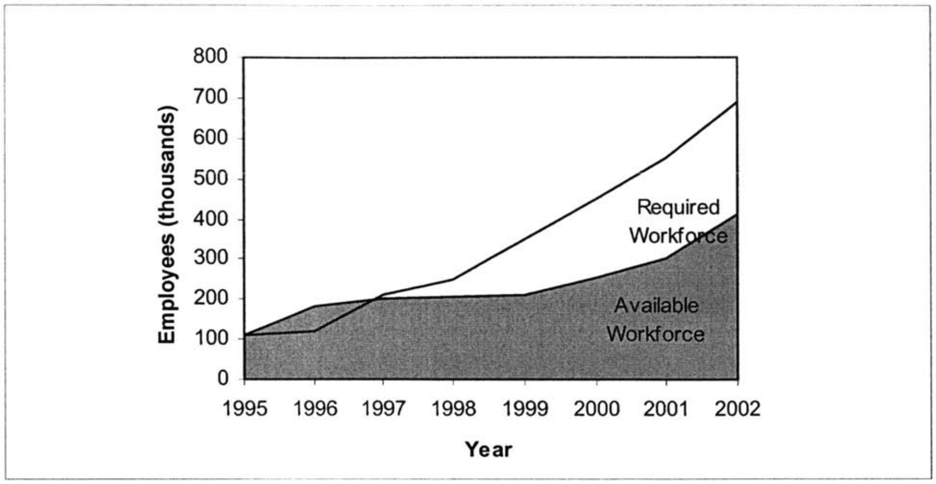

Figure 1: Hiring gap for information technology workers in the US.[2]

2.2.1 Nearshore Development

In recent years, there have been efforts by Mexican software companies in offshore development to distinguish the services done in countries that are geographically close to the acquiring country. The name for this specialized type of offshore development is "nearshore development." Companies that engage in nearshore development projects are usually from bordering countries or from the same region. For instance: Canada or Mexico to the US, Ireland to the United Kingdom, or Russia to Germany (Sinha and Scholl, 2002).

Apart from geographical proximity, which makes it easier to manage scope changes in a project, other factors distinguish nearshore development from traditional offshore development, which make nearshore development an attractive option to the acquiring country. These factors include cultural affinity, same time zone, short traveling

---

[2] This figure was taken Amoribieta et al. (2001) according to IDC projections. Workforce for years 2001 and 2002 are forecasted.

time, geopolitical similarity (associated with stability in the acquiring country), trade agreements, and in some cases, the same language.

As mentioned by Sinha and Terdiman (2002), cultural affinity is important because different cultures have different value systems and different approaches to solving problems. Same time zone benefits arise mainly in the area of communications and maintenance of systems. In the case of communications, the fact that the two countries have the same time zone lets partners in different countries have meetings, whenever these are necessary, at the same time. In the case of maintenance of systems, team members that are located in the providing country can see how the system behaves, and solve issues as they arise, as the system is used in the acquiring country. Short traveling time is important for critical projects, especially when problems arise at any stage of the project and team members from the providing country are under time pressure to arrive to the acquiring country. Geopolitical stability is important for various reasons. For example, regions that are prone to violence can be disruptive for business activities, or countries with geopolitical risks may have restrictions on travel by the acquiring country (Sinha and Terdiman, 2002). Finally, trade agreements between the two countries are also an important factor to consider. For example, several benefits from trade agreements include special types of visas for traveling to the acquiring country.

2.2.2 Offerings Spectrum[3]

Within the segments of applications and maintenance & support, there are different offerings in offshore development. The applications segment includes development of legacy and 'new' applications. In legacy applications, customize application development is done in older programming languages such as COBOL, VSAM, Mainframe and DB2. In new applications, newer technologies are used to build e-commerce and functional applications.

The maintenance & support segment includes Applications Maintenance & Support, applications redesign and migration, package applications, and call centers. Applications Maintenance & Support provides ongoing maintenance, management, enhancement, and support to an existing application. Applications redesign and migration provides an upgrade of the tool or the platform. An example of this offering is the 'Y2K bug.' The package application offering provides implementation support in the form of modifications, extensions, and user support for applications from different vendors.

Figure 2 ranks these offerings within the applications and maintenance & support segments according to their maturity. Maturity is defined according three main points: maturity of the offering, end-user buyer "mainstream" acceptance, and volume of business.

---

[3] The information presented in this section, regarding the different offerings in offshore development, is from Karamouzis (2002).

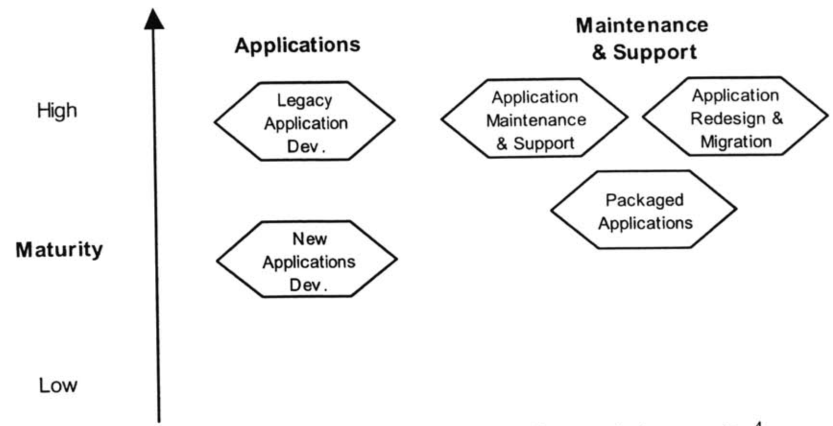

Figure 2: Offshore offerings spectrum according to their maturity.[4]

Other offerings within both the software segments and the offshore development spectrum include business intelligence, business process outsourcing, enterprise application integration, and infrastructure. Most of these offerings, however, are not considered 'mature enough' or are currently unpopular offshore attractions.

2.3 The US Offshore Development Market

In 2000, the US was the dominant leader in the world's software spending, accounting for around 50 percent (WITSA, 2002). Of this market, 45 percent was spent in the segments of applications and maintenance & support. The offshore development market represented 6 percent of software spending in the US. However, due to the recent economic slowdown, many companies are expected to go offshore to lower costs. As a result, the offshore development market is expected to increase significantly in the near future. Table 1 summarizes the software market in the US, including the markets for

---

[4] This figure was adapted from Karamouzis (2002).

outsourcing and offshore development, for 2000 and 2005. Software spending for 2000 was taken from WITSA (2002); for 2005, software spending was estimated by the author according to the compound annual growth rate (CAGR) from 2000 to 2001 (see Appendix B). For outsourcing and offshore development, the data was taken from Maroto and Zavala (2003) according to IDC projections.

Table 1: **Software market in the US in 2000 and 2005 (US$ million).**

	2000	2005	CAGR
Software Spending	90,969	122,543	6.14%
Outsourcing	56,000	100,000	12.30%
Offshore Development	5,500	17,600	26.19%

Source: WITSA 2002; Maroto and Zavala 2003; and estimations by author.

The most popular destination in offshore development has been India. In 2000, India accounted for a little more than 70 percent of the market with exports sales of $3,900 million (Arora et al., 2001). The remaining (non-Indian) offshore market share of $1,600 million was scattered among other countries including Mexico, Russia, China, the Philippines, Malaysia, and countries within Eastern Europe.

2.4 Mexico's Position in Offshore Development

Mexico occupies a unique position with respect to offshore development. Bordering the US, Mexico is able to offer its offshore services under the nearshore development scheme, giving Mexico an advantage over other countries. In addition, the North American Free Trade Agreement (NAFTA) with the US and Canada gives Mexico an advantage over other countries. These advantages include legal and contractual issues, which assures US companies of the services they subscribe with Mexican providers;

security issues and intellectual property rights, which protect the company's systems that are being coded or maintained across the border; and temporary work visas, including the "NAFTA visa" (type TN), which is easier to get for Mexican workers who need to travel to the US at any stage of an project.

To see where Mexico stands in the offshore development market, Figure 3 illustrates the dominant players in the market against two orthogonal axes for considering going offshore: quality of supply and cost.

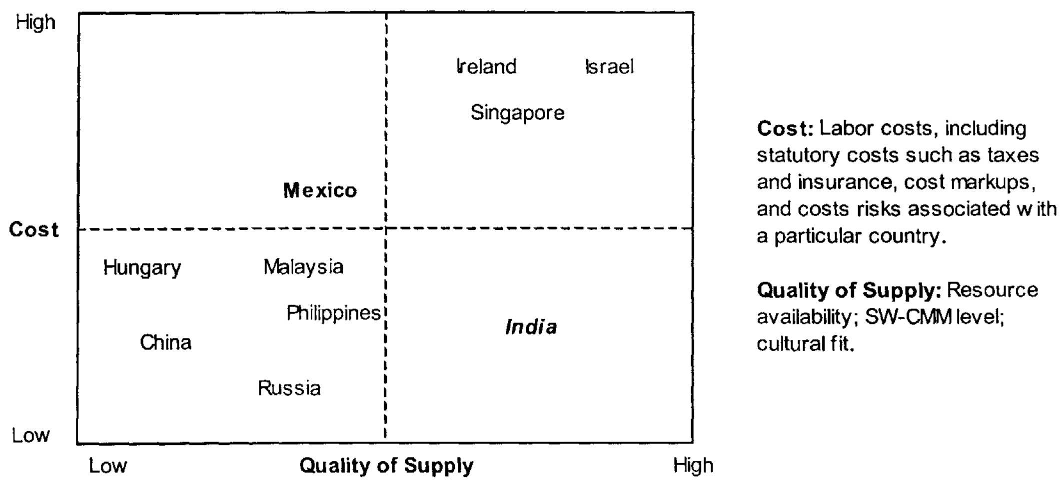

Figure 3: Mexico's position in offshore development.[5]

As seen in Figure 3, Mexico lags other countries in offering offshore development services, and has not been able to capitalize the advantage of bordering the US and NAFTA. To improve the quality of services and the supply of labor, the Mexican IT Industry Association (AMITI) and the Mexican government have been making efforts by outlining several points for developing the Mexican software industry (AMITI and

---

[5] This figure was taken from Amoribieta et al. (2001).

Bancomext, 2001). Figure 4 shows the revenues of offshore development companies in Mexico. In 2001, offshore development revenues amounted to $134.8 (Select, 2000).

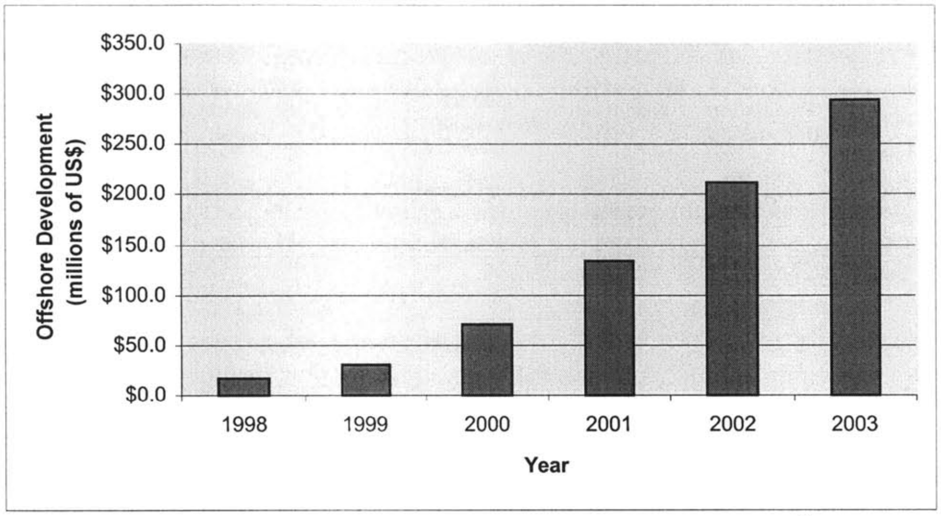

Figure 4: Offshore development revenues for Mexican providers.[6]

2.4.1 Country Evaluation Model Criteria

Labor, infrastructure, and capital are other important criteria when evaluating countries for going offshore. This section provides a brief analysis of Mexico regarding these three criteria. This analysis is done by presenting macroeconomic variables of the US, Mexico, and India (the leader in offshore development). Appendix B.2 provides a description of the measured information and the rationale for these variables.

2.4.1.1 Labor

Labor is probably the most important component of software companies. Regardless of the approach to software development, these companies still depend on

[6] This figure was taken from Select (2000). Revenues for years 2001, 2002 and 2003 are forecasted.

highly qualified software developers. Table 2 shows several variables measuring the level of IT sophistication and the cost of labor across the US, Mexico, and India.

Table 2: **Labor criterion comparing the US, Mexico, and India.**

	Total Population (thousands)	Share of H1-B Workforce (%)	Minimum Wage (US$)	Scientists and Engineers	High-Tech Share of Manufactured Exports (%)	Personal Computers
US	278,230	N/A	$8,056	1,022,773	35%	142,036
Mexico	96,586	13.0%	$768	20,669	21%	4,269
India	997,515	42.6%	$408	148,630	6%	3,292

Source: Scholl et al. 2001. N/A: Not applicable.

As shown in Table 2, Mexico falls behind India in offering qualified labor. To generate this qualified labor pool, there is the need for a good formal educational system; India seems to have this system along with a large population that can join it. An advantage that Mexico has over India is cultural affinity within this pool, though its importance is unclear. A related aspect to consider is the number of students in computer science programs. In Mexico, this pool has been increasing considerably in past years (see Appendix C.1). Most of these students, however, do not graduate with the desired level of English that is needed for offshore development projects.

India also offers lower labor costs than Mexico. However, since Mexico borders the US, low travel costs (and short traveling time) can substitute these labor costs when offering offshore development services. For example, the average cost of a round-trip flight from the capital of Mexico (Mexico City) to New York City is around $400 and lasts approximately 7 hours, while the average cost of a round-trip flight from the capital of India (New Delhi) to New York City is around $1,400 and lasts approximately 20 hours (Expedia, 2003). Telecommunication costs may also be considered when team members communicate using traditional services such as telephone lines, but nowadays the Internet may used as form for communicating.

2.4.1.2 Infrastructure

Infrastructure is also an important aspect to consider when going offshore. The level of infrastructure determines the ability of companies within a particular country to conduct transactions within this country, and most importantly, with their customers in the US. Table 3 shows several variables that assess the overall level of information technology (IT) infrastructure across the US, Mexico, and India.

Table 3: **Infrastructure criterion comparing the US, Mexico, and India.**

	Internet Users (thousands)	Secure Servers (units)	GDP (US$ billion)	ICT Expenditures (US$ billion)
US	74,100	73,386	$9,255	$823.7
Mexico	1,822	218	$865.5	$77.0
India	2,800	85	$1,805	$160.6

Source: Scholl et al. 2001.

From the table above, we see that the differences we saw before in the labor criterion are starting to narrow. In other words, Mexico is in better shape against India with respect to the infrastructure criterion. The liberalization of the computer market in Mexico, which started in the early 1990s, has been one of the reasons for the improvement in the level of infrastructure in Mexico (Dedrick et al., 2001). Another aspect to consider is the efforts of the Mexican government to increase the country's infrastructure and connectivity to the Internet (see Section 2.4.2.2).

2.4.1.3 Capital

The third criterion considered in this section is capital. Access to capital is important to determine the overall level of investment within a country across various industries. Table 4 shows several variables measuring this level of investment across the US, Mexico, and India.

Table 4: **Capital criterion comparing the US, Mexico, and India.**

	Foreign Direct Investment's Share of Gross Capital Formation (%)	ICRG Risk Rating	Gross Capital Formation, 1990-1999 CAGR (%)
US	10.6	82.0	6.7
Mexico	10.5	73.0	3.8
India	2.1	61.8	6.5

Source: Scholl et al. 2001.

From the above table, we see that Mexico has an advantage over India with respect to capital. One of the reasons for this may be NAFTA, however, it is unclear how these variables affect offshore development. An important aspect to consider is the role of the Mexican government in promoting the software industry. For example, national development banks in Mexico, such as Bancomext and Nacional Financiera, have programs for granting credits to software companies.

As for country risk, we see that Mexico has a lower risk than India. A reason for this is the stability of countries within their regions. For example, the India-Pakistani conflict may be seen as a threat for companies considering offshore development in India (Rosenthal, 2001).

2.4.2 Government Programs

Since 2000, the Mexican government has been making efforts to promote the software industry, and has launched a project to connect the country to the Internet (improving the telecommunications infrastructure in Mexico). These two efforts are the Software Industry Development Program (SIDP), developed by the Mexican Ministry of Economics with aid of the industry and academic sectors; and the e-Mexico project, developed by the Mexican Ministry of Communications and Transportation.

2.4.2.1 Software Industry Development Program

The two main goals outlined in the SIDP for 2010 are: reaching $5,000 million in software revenues (including revenues from offshore development), and reaching the world average information technology divided by gross domestic product (IT/GDP ratio). The first goal of achieving $5,000 million in software revenues is based on what has been accomplished by India in past years: in 1994, India had software revenues of $485 million, and achieved $5,000 million six years later (Arora et al., 2001). To have software revenues of $5,000 million in 2010, the Mexican government has set a description of the number of companies that would have to exist according to their SW-CMM level (SE, 2002). Table 5 summarizes the situation in 2001, the goals set for 2010, and the necessary annual growth rate to achieve the SIDP 2010 goals. For 2001, the data was taken from WITSA (2002); for 2010, the data was taken from SE (2002).

Table 5: Mexican software revenues and IT/GDP ratio in 2001 and 2010.

	2001	2010	CAGR
Software Revenues (US$ million)	$597	$5,000	26.64%
IT/GNP	1.40%	4.30%	13.28%

Source: WITSA 2002; and SE 2002.

2.4.2.2 E-Mexico Project

Another related effort by the government is the e-Mexico project. This initiative is designed to connect 85% of the country's population to the Internet by the year 2006, which will partly be done by installing 10,000 digital community centers (DCC) and by connecting around 145,000 public schools to the Internet. This system will allow communities the access to services such as education, health services, business, government and science, technology and industry (E-Mexico, 2003). The main partners so far in the e-Mexico project are Microsoft and Intel. In 2002, Microsoft committed to contribute $58 million over the next five years. This deal includes software, technical support, and training for 20,000 technicians and engineers to set up the DCC's. Intel also agreed to donate $17 million to train 17,000 elementary-school teachers in computer technology. Other deals with computer makers IBM and Hewlett-Packard are planned by the government (Gori, 2002).

Several factors such as the current economic slowdown, however, have indicated that the objectives of the project will not be realized on time. For example, even though the program was announced in 2000, the first phase of the project did not begin until the beginning of 2003 (Vizcaino, 2003). Moreover, there are questions to why the Mexican government did not choose to adopt an Open Source software approach for this project. As mentioned by de Icaza (2003), an Open Source approach has many advantages over a proprietary approach; most importantly, an Open Source approach would create a base of open source software developers, and would enable significant savings in royalties for proprietary software.

Chapter 3:
Organization of Work at a Nearshore Development Center

This chapter focuses on the organization of work at Azteca Software. The chapter is structured into four sections. The first section gives a brief overview of Azteca Software. The second section presents the organization of work for two types of projects: Applications Development and Applications Maintenance & Support. The third section describes the training and career development options available for employees at Azteca Software. Finally, this chapter ends with an analytical summary section.

3.1 Company Overview

Azteca Software is one of the leading providers of software development and maintenance & support services in Latin America. The company, founded in Mexico in the early 1980's, has presence in the US, Spain and most of Latin America.

Azteca Software reached the US market by means of an important US services company. This company had started outsourcing software development projects to India in the early 1990's, and then focused on Latin America in 1997. After becoming a provider to the US services company, Azteca Software targeted other corporate customer in the US. Today, Azteca Software focuses on Fortune 500 companies.

There are around 450 employees at Azteca Software's nearshore development center. Of these 450 employees, 300 are in software development and the rest in operational work. In addition, around 30 employees work in the US as account managers, project leaders, analysts, designers, and programmers.

Revenues for Azteca Software in 2002 were $80 million, with the US representing around 35 to 40 percent. In terms of quality assessment, Azteca Software is assessed SW-CMM Level 3, and has been practicing the Six Sigma Initiative for four years.

3.2 Organization of Work

The types of services that Azteca Software offers are: Applications Development (AD), Applications Maintenance & Support (AMS), Latin America Localization, Enterprise Application Integration, and Testing. The organization of work in AD and AMS projects are covered in this section. These projects are the most typical at Azteca Software, and consequently, the source of a large percentage of revenues.

3.2.1 Applications Development

Azteca Software offers AD services to customers who are interested in tailored applications. Azteca offers these development services within a variety of industries and applications. These applications include Internet banking, supply chain management, telecommunications billing, customer service, distribution systems, decision support, data warehousing, human resources, sales support, and service tracking among others. AD projects are usually under-defined and vary from project to project in terms of type of application, technology, customer and knowledge domain.

AD projects are composed of three main stages: definition, development and conclusion. The development stage consists of four phases: inception, elaboration, construction and transition (see Figure 5).

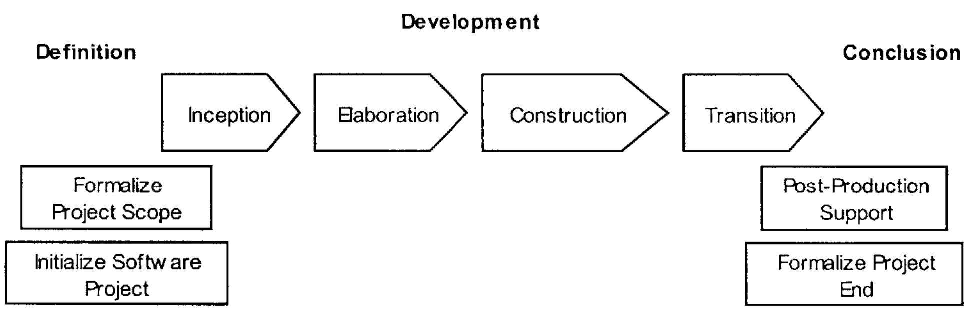

Figure 5: Main AD project stages, including the four project development phases.

During the definition stage, a project is formalized with a potential customer. The project is then designed and constructed during the development stage, and reaches its end during the conclusion stage. These stages are explained in detail in the following subsections, paying particular attention to the organization of work during the development stage.

3.2.1.1 Definition Stage

During the definition stage, the goal is to initialize a project with a potential customer. This customer may auction the project to several outsourcing companies. At Azteca Software, the pursuit team formalizes the project scope by submitting a project proposal. If the bid is won, the project is initialized.

Role Descriptions

There are six roles present during the definition stage: account manager, regional manager, operations leader, technical leader, marketing assessor, and project leader.

The account manager is responsible for selling the Azteca's services to potential customers. The account manager is based in the US according to a particular geographical location, and may be appointed to a specific set of companies.

The pursuit team, based in Mexico, is composed of a regional manager, an operations leader, and a technical leader. The regional manager and the operations leader, who specialize on a particular set of customers, are responsible for developing the proposal, estimating the cost of the project, and setting up the project team. The technical leader, who specializes in a particular technology, is responsible for all the technical details of the project proposal such as the methodologies and platforms to be used.

The marketing assessor is responsible for working on the presentation aspects of the proposal once it is ready to be presented to the customer. The project leader, who is chosen by the regional manager if the bid is won, is responsible for defining and completing the project.

Definition Phases

a. Formalize Project Scope

The objective of the formalization project phase is to present a project proposal to a customer. For this to happen, the customer auctions the project to several companies.

At Azteca Software, the pursuit team performs a project cost estimation and submits the project proposal.

Project Auctioning[7]

During the project auctioning, the potential customer provides a request for proposal (RFP) to several companies. In the RFP, the customer specifies the general job description of the project.

In small projects, the customer picks the company (two weeks after they issued the RFP) based on the proposal that meets the specifications; subsequently, the project starts. In large projects, the process is longer. After two months of working in the project proposal, the customer hosts a question/answer conference call session with all the companies bidding for the project. In this session, each company poses questions regarding the project specifications in a round robin format. This session lasts until there are no more questions.

During this phase, the objective of the pursuit team is to formalize the project scope to develop the proposal. For this, the account manager meets with the pursuit team: a regional manager, usually the nearshore development center director; an operations leader, chosen according to the customer; and a technical leader, chosen according to the technology for building the system. In this phase, the objectives of the pursuit team are the following:

- Define solutions to the business requirements
- Define acceptance criteria for the proposed solution
- Develop an estimate to complete the solution

[7] This auctioning process is particular to an important financial company that was interviewed for this thesis. However, this process can be generalized to cover a broad number of cases.

The pursuit team defines the business requirements and its acceptance criteria by determining the technology of the system, and by using experience from other projects.

Project Cost Estimation

To develop an estimate of the cost of the project, the *project effort* (in hours) is calculated – the number of hours required to complete the project. This effort is calculated by determining the functional reach of the system, which measures the complexity of the system. This functional reach is then used to determine the individual efforts required for the system; which is then summed to come up with the total project effort.

The project effort can be used to classify the size of projects as described in Table 6. The project effort is then used to calculate the calendar of the project and the number of resources to assign to a project.

Table 6: **AD project sizes according to project effort.**

Size	**Effort (hrs)**
Small	0 - 600 hrs
Medium	600 - 1,000 hrs
Large	1,000 - 6,000 hrs
Macro	More than 6,000 hrs

A special consideration is given to risks when calculating the project effort. Several factors that that may come into play are: familiarity with the technology; knowledge domain; customer maturity, determined by previous experience with Azteca Software or with other offshore development companies; infrastructure stability with the customer; and availability and training of resources to be assigned to the project.

Knowing the project effort leads to calculating the *man-months* required to build the system. The man-months are calculated using equation below. This equation takes the effort required for a project and divides it by the average number of hours (162) that a person works in a month.

$$Man - Months = \frac{Effort}{162}$$

Then, the *project duration* (in weeks) is calculated using the equation below. The basis of this equation is experience estimating past projects.

$$Project\ Duration = \sqrt{\frac{Effort * 2.5}{40}}$$

The *number of resources* (persons) required for a project is then calculated using the equation below. After the number of resources is calculated, the pursuit team makes sure that this estimate is within the bounds of the project effort and the project duration. Then, the pursuit team sets up a project development calendar.

$$Number\ of\ Resources = \frac{Effort}{40} \Big/ Project\ Duration$$

Assigning resources to the four project development phases (inception, elaboration, construction, and transition) is the next step. This is done by taking into account the project effort, project duration, and the project development calendar.

To determine which specific resource will be assigned, consideration is given to the role of the project member (project leader, analyst, designer, senior programmer, junior programmer, and tester), to the project member's experience, to the technology at hand, and to the geographical location of the resource.

In general, the cost for each resource to the customer is $20 if based in Mexico, and $40 if based in the US. However, there are other indirect costs to consider such as traveling and communication expenses.

Table 7 summarizes the calculations that have been mentioned for the different classifications of projects:

Table 7: **AD project estimations according to size.**

	Small	Medium	Large	Macro
Effort (hrs)	600	1,000	6,000	10,000
Man-months (months)	3.7	6.2	37	61.7
Project Duration (weeks)	6.1	7.9	19.4	25
Average Number of Resources (persons)	2.4	3.2	7.7	10
Cost Estimate for Customer (US$)	$12,000	$20,000	$160,000	$260,000

Note that these calculations are generalizations and depend mostly on the nature of each particular project. As mentioned above, the main determinants of this cost are project effort, resources, the place where the resource is based, and indirect costs. In an interview, an operations leader stated,

> *"Once you have developed an estimate, you then make sure that the estimation makes sense... This is done by experience, consensus with a group of people, and by making comparisons to similar projects... This estimating process is automated from 80-90 percent."*

The cost estimate for the customer is presented in Table 7. This estimate is approximately narrowed down in the following manner:

- Labor costs: 40%
- Sales expenses: 15%
- Management expenses: 10%
- Utilities: 35%

Calculating these percentages leads the team to decide if the project is worth pursuing. Another consideration is the nature of the customer, which may be new or consolidated; the pursuit team considers the work potential that the project may give the company.

Note that through the development stage of the project, these cost percentages may change. For example, if there is a delay in a project, then the utilities on the project will be affected. The other costs are direct and cannot be avoided.

Proposal Submission

After two months, the companies competing for the project submit their final proposals containing technical information along with information regarding the cost, delivery time, and resources. Then, customer managers spend around two weeks reviewing all the proposals.

Then, the customer holds a presentation session. In this session, each bidding company individually presents its project for one hour, followed by a 30-minute question and answer session.

With the particular customer interviewed for this thesis, two companies are then chosen. The projects of these two companies are reviewed in detail so that the customer can then make its final decision.

b. Initialize Project

If the bid is won for the project, the software project is initialized. The account manager develops a statement of work (a technical document based on the proposal).

The regional manager selects the project leader for the project, who may be the technical leader who was part of the pursuit team. The main objectives of this phase are the following:

- Identify the deliverables that were agreed upon
- Prepare the development strategy
- Provide overall planning for the project by identifying risks

In some cases, the project leader chosen for a project may not be the same as the technical leader who estimated the initial cost of the project. In an interview, a project/technical leader stated,

> *"If I am not the technical leader for the project, I like to revalidate the estimate by performing my own estimates based on past experiences (without paying much attention to the estimating tool)."*

He also went on to comment about discrepancies that may arise between his estimations and the ones that were done before.

> *"Sometimes there are discrepancies between the estimates that are presented to me and the estimates that I do. In most of these cases, the customer may be strategic, and the company (regional manager) is usually willing to lower their prices in order to build a long-term relationship with the customer (that may benefit the company later on)."*

Usually, project leaders have experience in different knowledge domains. However, when project leaders are presented with a new domain, they research the literature and ask people in (or outside) the company to learn about the new domain.

After the project leader has evaluated the cost estimate of the project, a general work plan is developed, including overall project management and quality plans, work guidelines, risk plans, and a general work plan. Then, the project leader does a requisition of resources (persons and computers), which are based on the proposal. At this point, the project leader may be under pressure to lower the project's cost. However, the project leader is always careful and even battles for more people in order to deliver a quality product to the customer.

A kick-off meeting, where a formal presentation takes place with the account manager, operations leader, project leader and the customer representative (contract manager), is then held. If the customer is mature in offshore projects, the meeting is usually done via telephone. Otherwise, the contract manager (customer) might go to the nearshore development center to meet the team members and visit Azteca's installations. During this kick-off meeting, the general work plan is presented as well as details on trips, project features, weekly status meetings that are to be held between the project leader and the contract manager, and the details on the next analysis meeting. In general, subsequent meetings with the project leader and contract manager are done via telephone or e-mail.

3.2.1.2 Development Stage

Once the project has been formalized and initialized, the development stage begins. In this stage, the project members build the system starting with a prototype and ending with a production period going through the different phases shown in Figure 6.

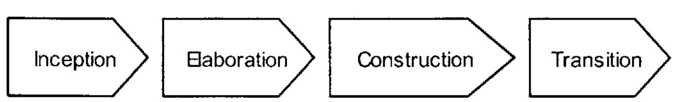

Figure 6: AD project development phases.

Team Composition

Figure 7 shows the conceptual team composition during the development stage. The location of a team member (US vs. Mexico) mostly depends on the size of the project. In general, small projects are entirely based in Mexico through all the phases; medium projects are also based in Mexico, but during the inception phase, team members go to the US and visit the customer. For this reason, the role of the nearshore coordinator is eliminated in small and medium projects. In the case of large projects, the development team is separated between the US and Mexico.

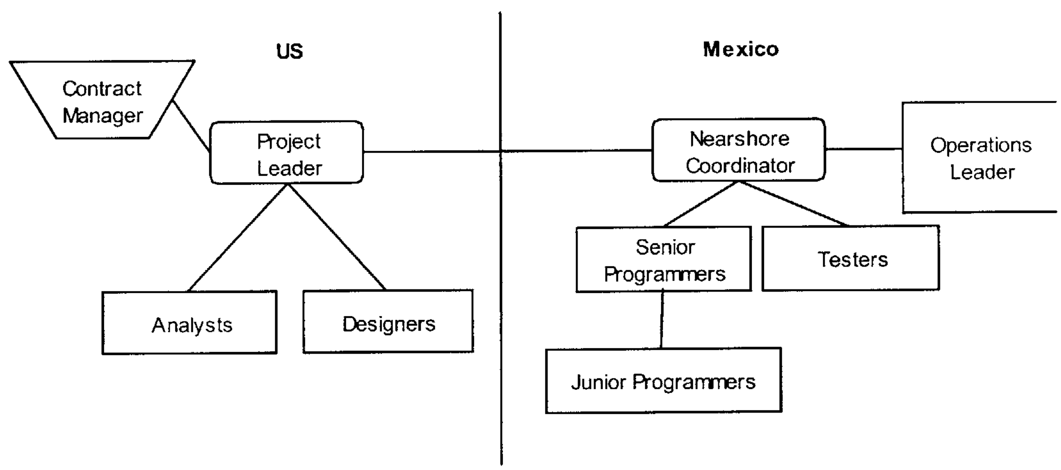

Figure 7: Team composition for large AD projects.

The division between small, medium, and large projects and where the team members are based is not set in stone. For example, there may be cases where small or medium projects may be separated, in terms of location (US vs. Mexico), as large projects. For this, special consideration is given to the effort and functional reach (complexity) of the project.

Role Descriptions

The different roles present during the development stage are the following: project leader, operations leader, nearshore coordinator, analyst, designer, senior programmer, junior programmer, and tester. Project members can play more than one role, and a role can be integrated by more than one project member.

The job of project leaders is to take the project vision of the customer to its completion. Project leaders are in close contact with the contract manager, who may raise any issues with the system being developed. Project leaders are the main architects of the system as well as the technical leaders. In large projects, project leaders personally supervise all the work done in the US, and supervise, through nearshore coordinators, the work done in Mexico. In medium projects, project leaders need to keep in mind the project's budget to decide the amount of time and the number of team members that will visit the customer. Finally, project leaders are also responsible for assigning tasks to the development team.

Operations leaders supervise projects so that they are built according to schedule. Operations leaders may solve problems that arise during a project. These problems include scalability issues, need of new resources, and training of project members.

Nearshore coordinators are responsible for managing the development team based in Mexico. Nearshore coordinators stay in close contact with the project leader, share their vision throughout the project, and transmit the vision to the AD team based in Mexico.

Analysts and designers help the project leader in defining all the technical issues in a system. They are involved in the early stages of a projects helping in the analysis and design respectively.

Senior and junior programmers code the system. Senior programmers are in close contact with the project leader or the nearshore coordinator, and usually share the vision of the system. In early project phases, senior programmers may have participated as analysts and designers. Junior programmers help to code individual modules that are assigned by the nearshore coordinator or senior programmers. Senior programmers, who are not involved in the early phases of a project, and junior programmers are usually handed pseudo-code[8] to code their assigned modules. Full-time employees or interns at Azteca Software, who are referred to as "trainees," may perform the roles of junior programmers.

Testers are in close contact with senior and junior programmers and help them in testing the modules that are being coded. The tester's main responsibility is to ensure that the system meets the customer's specifications.

Role Distribution

Roles are distributed through the project development phases (inception, elaboration, construction, and transition) as shown in the Figure 8.

[8] A notation with the structure of a programming language combined with comments of the computations to be performed.

	Inception	Elaboration	Construction	Transition
Project Leader	X	X	X	X
Nearshore Coordinator	X	X	X	X
Analyst	X			
Designer		X		
Senior Programmer			X	X
Junior Programmer			X	X
Tester			X	X

Account Manager
Operations Leader

Figure 8: Role distribution through AD project phases.

This role distribution does not mean that the marked roles are the only roles present at the specific project phase. However, these marked roles are responsible for the different activities that take place during the project phase. For example, senior programmers may help build a prototype of the system during the elaboration phase, but designers have to guarantee that the prototype is built.

Resource Assignation

Project members are assigned to the different phases by considering estimates that include effort, number of resources, project duration, and cost (see Table 8). Moreover, the following metrics are taken as model for the time (in terms of effort and project duration) that each phase should take:

Table 8: **Distribution metrics for assigning resources in AD projects.**

	Inception	Elaboration	Construction	Transition
Effort	5%	20%	65%	10%
Project Duration	10%	30%	50%	10%

Small Projects

Small projects are classified to take up to 600 hours in effort. They are calculated to take 6.1 weeks in duration, with an average of 2.5 project members through the development phases. In general, all project members are based in Mexico. To cut costs, the analyst takes the role of the project leader through all phases.

Table 9: **Occupational distribution for small AD projects.**

	Inception	**Elaboration**	**Construction**	**Transition**
Effort	30 hours	120 hours	390 hours	60 hours
Duration	4 days	12 days	3 weeks	4 days
Project Leader	x	x	x	x
Analyst	1			
Designer		1		
Senior Programmer/Tester			2	2
Junior Programmer/Tester			2	2
Total	1	1	4	4

As shown in Table 9, the analyst changes his/her role to a designer in the elaboration phase; and a senior programmer/tester in the construction and transition phases. In addition, new project members join the development team when the construction phase begins.

Medium Projects

Medium projects are classified to take from 600 hours to 1,000 hours in effort. They last for 7.9 weeks (2 months) with an average of 3.1 project members through the development phases. In general, medium projects are based in Mexico except for the inception phase, where the project leader and analysts visit the customer in the US.

Table 10: Occupational distribution for medium AD projects.				
	Inception	**Elaboration**	**Construction**	**Transition**
Effort	50 hours	200 hours	650 hours	100 hours
Duration	6 days	16 days	3 weeks	5 days
Project Leader	1	1	1	1
Analyst	1			
Designer		1		
Senior Programmer/Tester			2	1
Junior Programmer/Tester			2	2
Total	2	2	5	4

Similar to small projects, the analyst changes his/her role to a designer in the elaboration phase; and to senior programmer/tester in the construction and transition phases. New project members may join the development team in the construction phase, and other project members may leave during the transition phase to cut costs.

Large Projects

Large projects take from 1,000 hours to 6,000 hours in effort. These projects last around 19.4 weeks (5 months) with an average of 7.7 project members through the development phases. Project members are separated into two teams based in the US and Mexico (see Table 11).

Table 11: Occupational distribution for large AD projects according to location.								
	Inception		**Elaboration**		**Construction**		**Transition**	
	US	Mexico	US	Mexico	US	Mexico	US	Mexico
Effort	300 hours		1,200 hours		3,900 hours		600 hours	
Duration	2 weeks		6 weeks		10 weeks		2 weeks	
Project Leader	1		1		1		1	
Nearshore Coordinator		1		1		1		1
Analyst	3	1						
Designer			2	2				
Senior Programmer					2	2	2	2
Junior Programmer						3		3
Tester						1		1
Total	4	2	3	3	3	7	3	7
	6		6		10		10	

Similar to small and medium projects, analysts change their role to designers in the elaboration phase; and to senior programmers in the construction and transition phase. However, project members who are based in the US do not come to Mexico between phases, but instead change projects.

Most of the project members that are permanently based in the US are Mexican. These project members, and the project members who visit the customer in medium projects have type I-99 visas. Americans, who are based in the US, also work for Azteca Software since there is the need for people with the knowledge of the American culture.

Project members are paid in the respective currency (US Dollars or Mexican Pesos) of where they work: the US or Mexico. Moreover, Azteca Software pays travel expenses for people visiting the customer, and pays for relocation costs if the assignment is long.

Development Phases

This section presents a study on the organization of work during the different project development phases. To have a clear separation between the project phases, Azteca Software uses gates between each development phase (see Figure 9).

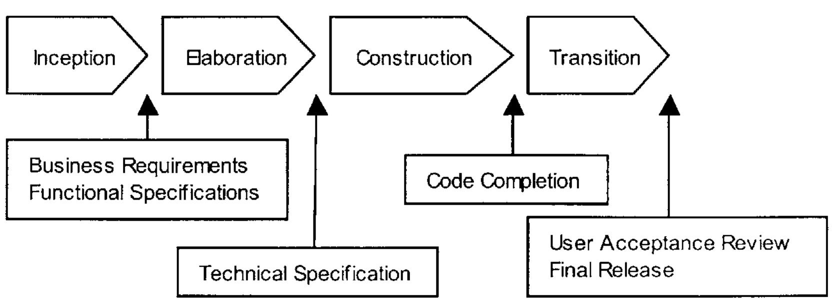

Figure 9: AD project development phase gates.

a. Inception

The inception phase begins with an analysis meeting between the project leader and the contract manager. The main objectives of the inception phase are as follows:

- Create a vision of the functionality of the system
- Identify the scope of the system
- Design and approve the user-system interaction

Figure 10 shows the interaction between project members during the inception phase.

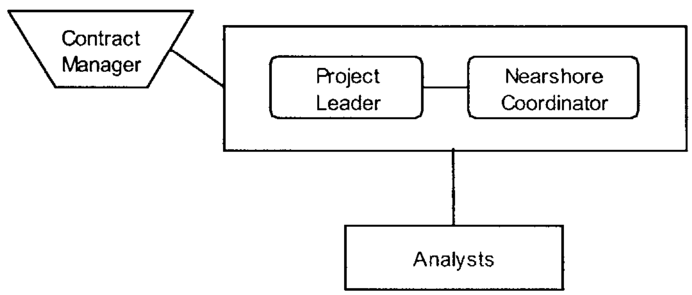

Figure 10: Interaction of the AD team during the inception phase.

After the analysis meeting, the business requirements are documented, which is the first gate of the process. For small projects, the business requirements are prepared by the project leader alone. In larger projects, analysts help the project leader to prepare the business requirements.

Subsequently, the project leader takes on the role of a system's architect, and proceeds to build a prototype (user-interface design) of the system. This prototype is designed to have the look and feel of the actual system, and it is shown to the contract manager once it is built, who provides constant feedback. This process can last from two to three iterations until the customer is satisfied with the prototype

In an interview, a project leader stated, *"Most of the time, we exceed the customer's expectations on the quality of the systems we design and build."*

Finally, the project leader and analysts document the functional specifications by analyzing the different variables used by the system. The functional specifications document, the second gate in the process, determines the functionality of the system. In addition, the contract manager reads these specifications to approve the proposed system.

b. Elaboration

The elaboration phase starts seamlessly at the time the functional specifications are being built. The main objectives of the elaboration phase are as follows:

- Establish the technical feasibility of the solution
- Design and approve the data architecture model
- Transform the functional specifications into software component specifications

Figure 11 shows the interaction between project members during the elaboration phase. As shown in the figure, project members who took the role of analysts during the inception phase change their role to designers in the elaboration phase.

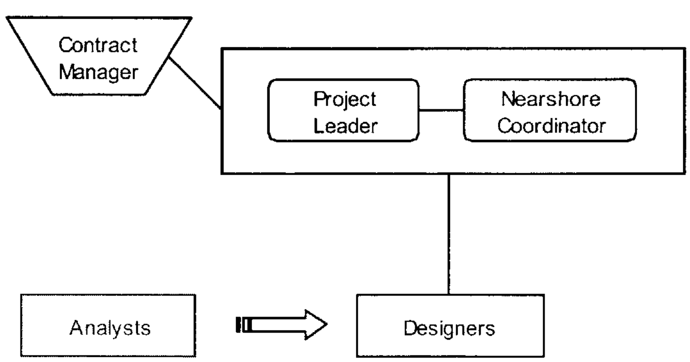

Figure 11: Interaction of the AD team during the elaboration phase.

During the elaboration phase, the project leader analyzes the technical decisions and defines the architecture. The project leader then documents the technical specifications (the third gate in the process), where more information about the inner works of the system is specified. This specification includes technical information such as the use of tables, databases, etc. Moreover, the project leader defines the test case scenarios to be used at a later point in the development process. Designers help the project leader in all aspects.

c. Construction

Coding of the system is done during the construction phase. The main objectives of the construction phase are as follows:

- Transform the component specifications into component code
- Perform unit-level testing to each component
- Perform system-level testing

Figure 12 shows the interaction between project members in this phase. Note that designers change their role to senior programmers in the construction phase. Moreover, junior programmers and testers join the team, and when the project requires, other senior programmers join the team as well.

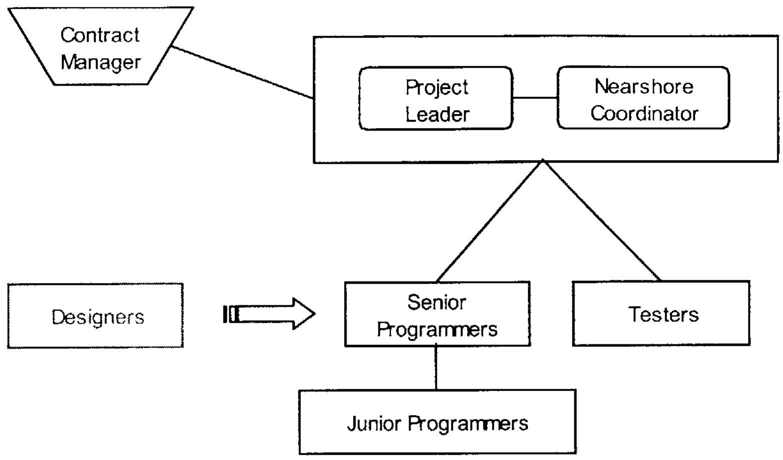

Figure 12: Interaction of the AD team during the construction phase.

When new project members join the project, they study the functional and technical specifications to learn about the system. Senior programmers who have been part of the project since the inception phase share the vision of the project leader in a deeper way than new programmers who just joined the project.

In an interview, a senior programmer emphasized that these specifications need to be well written so that new project members can have a full understanding of the system. In another interview, an analyst stated,

> *"A difficult part during projects is when new programmers are added to the team. The problem with this is the varying knowledge of people who are involved in a project."*

To start the coding of the system, the project leader meets with his chosen "star (senior) programmer" to work on the details of the work plan. They also set the standards of the particular system, and the project leader assigns different modules to senior programmers. The same project leader who was interviewed stated, *"I, with my star programmer, build the central part of the system and get the message across to the programmers to build the system I have in mind."*

The project leader tackles the most complex part of the system for one reason: programmers can use his code as a form of design and have an easier time in coding what they are asked. In addition, when the project leader assigns work to different programmers, he also hands them pseudo-code of their specific assignment along with a checklist of information regarding error handling, look and feel of the system, access of objects, documentation guidelines, etc. The senior programmers then use the pseudo-code and the checklist to start coding.

When senior programmers code their assigned modules, they assign submodules to junior programmers, who are only focused on their specific submodules and are usually not aware of all the other modules in the system. Senior programmers help junior programmers in case the latter have questions.

It is important to notice that the project leader is the only one who needs to know about the domain knowledge of the system. By making pseudo-code and the central part of the system available to programmers, the project leader specifies exactly what needs to be coded.

Throughout the coding of modules, the project leader assists senior programmers with specific questions. If the project leader cannot answer these questions, he or she may then contact the contract manager via instant messaging, e-mail, or telephone. In an interview, a junior programmer stated,

"I never talk to the contract manager. I only have contact with the project leader and analysts."

As a result, the project leader also needs to be proficient in English, while senior and junior programmers need not. In an interview, an operations leader stated,

> *"English is not a barrier for programmers. All they have to do is include low-level comments in their code, which are based on the function that they are coding."*

Moreover, the same junior programmer stated,

> *"The only places where I use English are in documenting my code, in testing cases, and in error handling."*

After senior programmers are done with coding their assigned modules, they proceed to do unitary tests. These tests are supervised by the project leader, and both check information regarding validation and correct use of database and objects. When the unitary tests are done, the project leader may then assign other modules to the senior programmer. This process continues until the coding of the system is complete.

Next, there is an integral test of the system where all modules interact with each other. In this stage, the project leader, senior programmers, and testers check the senior programmers' code and the system's code. If there are any 'bugs' in the system, the original programmer fixes and documents them. In an interview, a project leader mentioned he could break code developed by programmers easily because programmers have difficulty considering the scenarios their code is going to handle.

This process continues until the system meets the functional and technical specifications. Moreover, the project leader assures that the system code is bug-free. The gate of the construction phase is the completion of the code.

d. Transition

After the code of the system is completed, the transition phase begins. The main objectives of the transition phase are as follows:

- Develop documentation and installation procedures
- Support user acceptance test
- Train users on software usage
- Deliver the developed software and documentation

Figure 13 shows the interaction between project members in this phase. Note that the same roles are maintained from the construction phase. New project members may not join the team at this point, but they may leave. In addition, programmers focus most of their attention on testing their code and getting the final details ready to deploy the system.

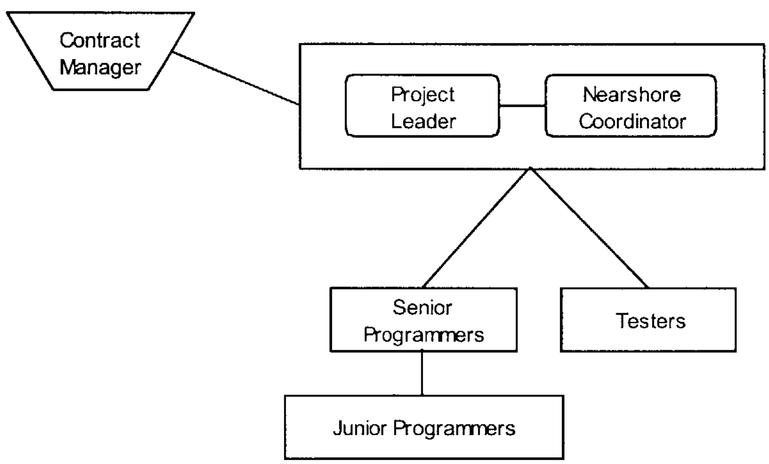

Figure 13: Interaction of the AD team during the transition phase.

In this phase, project members prepare an installation guide. Apart from this guide, there are times when the AD team prepares a configuration and a deployment guide. The configuration guide contains all the hardware and software requirements of the system; the deployment guide is a technical and detailed document containing the technical implications of the system, as well as enhancements and maintenance issues.

Once these guides are ready, the AD team replicates the customer's hardware and software environment by using machines sent by the customer. With these machines, the project leader completes performance tests, which serves as an internal user's acceptance test.

There are times when the customer requests a user's manual. However, as an operations leader pointed out, this is seldom the case, and the programmer's ability to be proficient in the English language is not as crucial as it would seem. In an interview, an operations leader stated,

> *"...As for the documentation of the system, the customer or external companies usually work on it. They are the ones who know the business domain and know what the system is doing for them. They are more qualified to write the documentation..."*

The project team then delivers the code and the guides to the customer, who performs a user's acceptance test. After this delivery, the project leader, the star programmer, and other project members (if the budget allows) have a week to fix any bugs that the customer finds. This is probably the most stressful period during the project phases, since the project members need to be working extra hours to have the system in optimal condition for its final release.

The gates that mark the end of the transition phase are the user's acceptance review and the final release of the system. This final release is usually done on a Friday, so that the system is ready to operate in the customer's installations by the following week.

3.2.1.3 Conclusion Stage

The conclusion stage begins after the delivery of the system. During this stage, the development team offers support for the system they have built.

Role Descriptions

The roles that are present in the conclusion stage vary according to each phase. During the post-production support phase, project leaders, nearshore coordinators, senior programmers, junior programmers, and testers take part. These roles perform activities similar to those performed during the development stage. During this phase, however, part of the development team provides support for the system they have built.

All the project members that took part on the project are present during the formal project end phase, where they are given feedback on their job performance.

Conclusion Phases

a. Post-Production Support

During the conclusion stage, the team members may offer post-production support to the application, which depends on the agreements of the original proposal. The support level that Azteca Software offers are: resolving issues by telephone, or creating system patches. The main objectives of the post-production support phase are as follows:

- Keep track of defects detected by the customer
- Conduct new enhancements
- Provide support

b. Formal Project End

Once the support phase is over, whether or not there was post-production support, the contract manager and the project leader formalize the end of the project in a meeting. In this meeting, the customer provides feedback on the development of the project. The main objectives of this phase are as follows:

- Provide feedback to team members, identifying strengths and areas of opportunity
- Capitalize customer's feedback to improve future performance
- Leverage the knowledge acquired during the project to improve future experiences

3.2.2 Applications Maintenance & Support

Azteca Software offers AMS services to customers that are interested in getting support for a set of applications. The types of services that Azteca offers include business support, maintenance & enhancement, and production support. These services are offered across different technologies such as object oriented software, client/server applications, mainframe applications, internet/intranet applications, and enterprise resource planning (ERP) applications.

AMS projects are composed of three main stages: definition, maintenance & support, and conclusion. The maintenance & support stage consists of two phases: implementation and operation (see Figure 14).

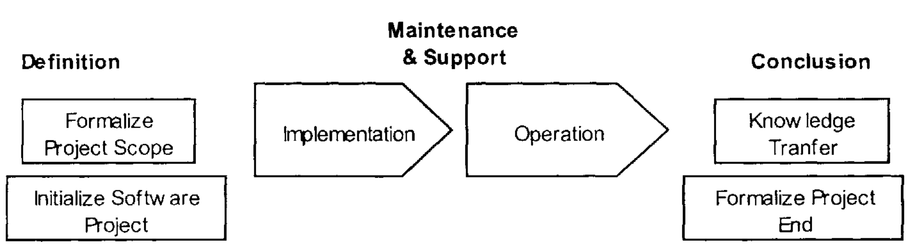

Figure 14: AMS project stages, including the maintenance & support phases.

Similar to AD projects, a project is formalized with a customer during the definition stage. The project is then maintained and supported, and reaches its end during the conclusion stage. These stages are explained in detail in the following subsections, paying special attention to the organization of work during the maintenance & support stage. Table 12 shows the time frame for AMS projects.

Table 12: Time frame for AMS projects.

Definition		Maintenance & Support		Conclusion	
Formalize Project	Initialize Project	Implementation	Operation	Transition	Project End
1-2 weeks	1-2 months	1-4 months	At least 4 months	1-6 months	

3.2.2.1 Definition Stage

During the definition stage, the goal is to formalize a maintenance & support project with a customer. These support services initialize from a general scenario: the original developer of the system, who may be offering post-production support, or the current support team is transferring the support of the system to a new provider.

Similar to AD projects, a customer may auction an AMS project to several software companies. At Azteca Software, the pursuit team formalizes the project scope by submitting a project proposal. If the bid is won, the project starts.

Role Descriptions

There are six roles present during the definition stage: account manager, regional manager, operations leader, technical leader, marketing assessor, and service manager (which is the equivalent of an AD project leader).

The responsibilities of these roles are similar to the ones outlined in the role descriptions for AD projects; the pursuit team's job is to win a bid for a project by presenting a project proposal. If the bid is won, a nearshore coordinator joins the project, and helps the service manager in planning and coordinating the transfer of on-going requirements from the current support team

Definition Phases

a. Formalize Project Scope

During the formalization project phase, the pursuit team elaborates a project proposal from the information in the RFP, following a similar auctioning phase to that in AD projects. At the end of this phase (in around 1-2 weeks), the pursuit team presents the project proposal to the customer. The main tasks during this phase are as follows:

- Gather information required from RFP
- Analyze information to elaborate a project proposal

The information required to elaborate a project proposal includes the following: number, size, and complexity of the applications to be maintained and supported; technical environment; documentation quality; AMS requests history; and current organization structure. Once the pursuit team has analyzed this information, the pursuit team presents the project proposal with solutions to the RFP requirements. This proposal includes the applications that will be maintained and supported along with the agreed services, a general work plan, and the costs of the services that will be provided. At the end of this phase, the team gets ready to start the project.

Project Cost Estimation

To develop an AMS project cost estimate, the customer usually states the *project duration* (in months) and suggests the *number of resources* (persons) that are needed to maintain the application. In some cases, the customer may ask the pursuit team for help on estimating the number of resources that are needed. If this is the case, the pursuit team bases these estimates on their experience with the customer or with similar applications. Table 13 shows the classification of AMS projects.

Table 13: **AMS project sizes according to project duration.**

Size	Project Duration (months)
Small	1 - 6 months
Medium	6 - 12 months
Large	12 - 24 months
Macro	More than 24 months

Once the project duration and the number of resources have been estimated, the *effort* (in hours) is calculated according to the formula below. The effort is the number of hours that the support team will be at the customer's request for maintaining and supporting the application. The equation below takes the project duration and multiplies it by the average number of persons in a project and by the average number of hours (162) that a person works in a month.

$$Effort = 162 * Project\ Duration * Number\ of\ Resources$$

Finally, the project cost is estimated. In this estimate, the effort of the support team is considered without taking into account management costs. The reason for this is that the customer usually manages the project by assigning tasks and supervising them. Table 14 summarizes the overall estimations for AMS projects.

Table 14: **AMS project estimations according to size.**			
	Small	**Medium**	**Large**
Project Duration (months)	6	12	24
Average Number of Resources (persons)	5	10	20
Effort (hrs)	4,860	19,440	77,760
Cost Estimate for Customer (US$)	$100,000	$500,000	$2,000,000

If Azteca Software takes the role of the service manager, additional management costs are considered. The decision of who manages the support team is based on the level of confidence with the customer and on the complexity of the application. In an interview, an operations leader stated,

> *"The customer does not usually give us the entire responsibility on the maintenance and support of applications... "*

Otherwise, if the customer gives Azteca Software the responsibility of managing the support team, the planning of the operational model is still performed by the customer.

b. Initialize Project

The project starts once the customer has accepted the project proposal. During this phase, which usually lasts around 1-2 months, a detailed project plan is issued and the support team is selected. The main tasks during this phase are as follows:

- Gather existing documentation
- Define operational model
- Define and configure infrastructure
- Define knowledge transfer plan

During this phase, the lead team (service manager and nearshore coordinator) identifies the current processes by gathering existing documentation from the leaving support team. This documentation includes the technical and functional specifications of the system to be supported, as well as a database of the issues related to the system. Then, the lead team documents and defines the operation of processes according to the services that will be delivered. These processes are then reviewed and approved by the customer.

Finally, the lead team designs the implementation of the support services that will be offered. This lead team works together for one week developing a transition plan, where they specify how the project is going to be managed. In this plan, they include information such as the use of documents, the processes to follow along with their requirements, the use of documentation, and specification.

The rest of the new support team is also selected during this phase. Prospective team members are required to have an interview with the service manager (customer) and the nearshore coordinator. Finally, the nearshore coordinator sets up the infrastructure and facilities in Mexico.

3.2.2.2 Maintenance & Support Stage

Once the project has been formalized and initialized, the maintenance & support stage begins. In this stage, the support team enters a knowledge transfer period and tests the operational model during the implementation phase. Then, the support team provides the maintenance & support service during the operations phase (see Figure 15).

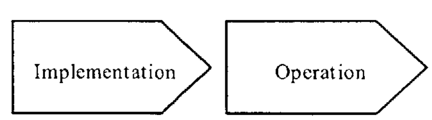

Figure 15: AMS project maintenance & support phases.

Team Composition

Figure 16 shows the team composition for typical AMS projects during the maintenance & support stage. Notice that the customer usually takes the role of the service manager. In general, project members are separated into two support teams: one based in the US and the other based in Mexico. Ideally, the distribution of support analysts is 30 percent in the US and 70 percent in Mexico. However, the customer determines how project members are allocated by considering the following aspects: telecommunications infrastructure, customer maturity with offshore development, project cost, project risks, information security, type of business, and type of technology.

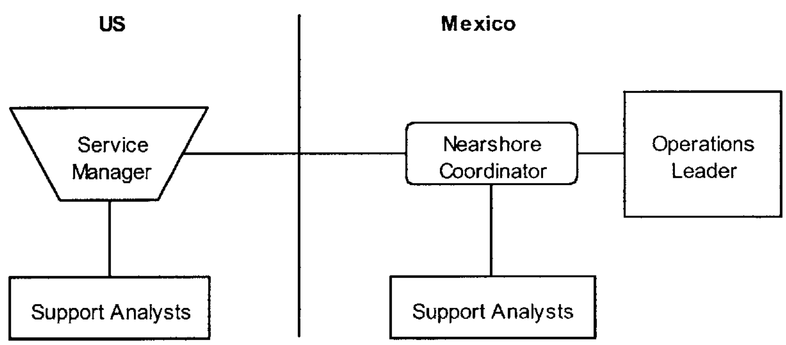

Figure 16: Team composition for AMS projects.

The service manager and the nearshore coordinator are in constant communication throughout the maintenance & support stage. The best way of communicating between project members based in the US and Mexico is by Internet instant messaging, e-mail, and telephone. Support team members are required to report time spent on support activities on a weekly basis. The support team also holds weekly meetings.

Role Descriptions

The roles that are present during the maintenance & support stage are the following: service manager, operations leader, nearshore coordinator, and support analyst. Similar to AD projects, an AMS project member can play more than one role, and a role can be integrated by more than one project member.

The service manager, a role generally taken by the customer and based in the US, performs all the management activities throughout the project. The service manager provides periodic status reviews to his/her managers. The service manager also assigns tasks to support analysts, and supervises them. In some cases, the service manager may also provide support for specific issues.

Nearshore coordinators are responsible for managing the activities based in Mexico. Nearshore coordinators monitor and assign tasks to support analysts in Mexico, keep records of these tasks, and report their status to the service manager. In some cases, nearshore coordinators may provide technical guidance for solving these requests. In addition, nearshore coordinators may provide support for some non-critical systems.

Support analysts are responsible for solving any requests that the customer may have. They also document activities performed, and report the status to their assigned requests to their direct manager (the service manager for support analysts based in the US, and the nearshore coordinator for support analysts based in Mexico).

Maintenance & Support Phases

a. Implementation Phase

During the implementation phase, the support team follows a knowledge transfer period, and the operational model is tested. At the end of this phase (in 1-4 months), the new support team is responsible for maintaining and supporting the system according to the operational model. The main tasks of this phase are the following:

- Perform knowledge transfer
- Deploy defined processes
- Test operation model

After the support team has been composed (at the end of the definition stage), the team is distributed between the US and Mexico. Then, the knowledge transfer process begins and part of the support team based in Mexico may visit the customer.

The first part of the knowledge transfer process is to train the new support team in the tools and in the applications that they will be supporting. For this, the support team analyzes the system's functional and technical specifications. The support team also reviews the applications and studies the system's user manuals. In some cases, however, the functional and technical specifications may not specify the system in detail, and the new support team has to analyze the code in detail to see how the system works. The service manager then coordinates the documentation of the system's specifications. In addition, the support team receives formal training in the processes that were outlined as part of the operational model. Coordination of this training process for the support team is done by the service manager in the US, and by the nearshore coordinator in Mexico.

After the support team has been trained, the leaving support team trains the new support team on how to support the system. This knowledge transfer is performed according to a plan developed during the previous stage of the project.

This second part of the knowledge transfer process usually lasts around three weeks, in which both the leaving and new support teams work together in the US. During the first week, the leaving support team solves all the customer's requests, while the new support team overviews the system and receives on-the-job training from the leaving support team. During the second week, the new support team starts solving a few requests, but the system is still under the responsibility of the leaving support team. Finally, during the third week, the new support team solves all the customer's requests, and the leaving support team overviews their work.

In some cases, however, the customer may not pay for these extra weeks of support to the leaving provider, and the new support team puts additional effort to prepare to support the system.

The final part of the implementation phase is to test the operational model that was defined in the previous stage of the project. For this, the service manager and the nearshore coordinator coordinate and verify a pilot test to validate the effectiveness of the operational model. This test is reviewed and approved by the customer. In addition, the support team and the customer agree on the service level and metrics definitions

b. Operation Phase

During the operation phase, the support team is responsible for providing AMS services that were agreed with the customer. The main tasks of this phase are as follows:

- Monitor service level agreements
- Deliver solutions to requests
- Document and improve process

There are three levels of support offered in AMS projects: help desk; production support, and maintenance & enhancements; and business support (see Figure 17). More knowledge in the application is required as the level of support increases. Currently, Azteca Software offers second and third level support; first level support is usually offered by the customer or an external provider. Third-level support is always based in the US since these requests are the most complex.

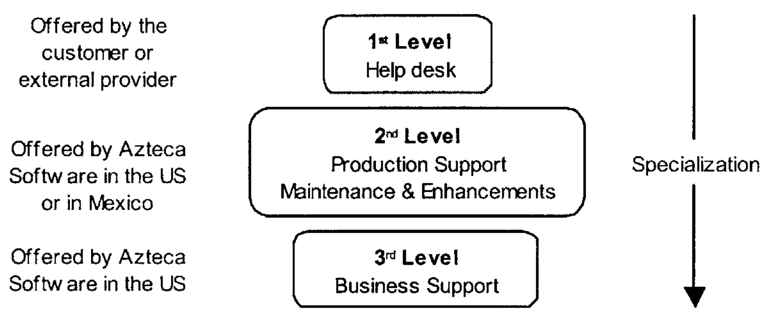

Figure 17: AMS levels of support.

The four kinds of severities that a request may have are business critical, high, medium and low. The solution time to these request priorities are: from 30 minutes to two hours for business critical requests; up to two hours for high requests; up to four hours for medium requests; and up to ten hours for low requests.

Managing Requests

When a user (the customer) has an issue with the system being supported, the user attends the first-level support – help desk. There, the user files the request, and the request is recorded in the Requirement Management Database (RMD). This database is accessible to the customer and the support team (see Figure 18).

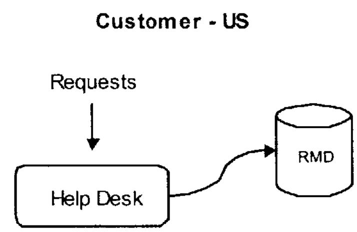

Figure 18: Filing of AMS requests by users at the help desk.

The help desk offers limited support, and if the help desk cannot solve the request, the request is sent to the second-level support – production support and maintenance & enhancements. Production support services provide continuous maintenance of applications. These types of support include emergency fixes, applications monitoring, and 24/7-call support.

Maintenance & enhancement services provide minor enhancements to applications that need to be updated to handle new features. These types of support include regulatory changes, incremental enhancements, and corrective and preventive maintenance. For example, the support team may adapt applications to new business or environments, or it may add functions to applications as requested by the customer.

If the request cannot be solved in second-level support, the request is sent to third-level support – business support. Business support services provide support to users of a set of business applications that were agreed with the customer. These types of support include business, functional, and operational user support. In this level of support, the support analysts must have knowledge in the application, platform, and in fine-tuning.

Solving Requests

To offer second and third-level support, the service manager continuously monitors new requests through the RMD and tracks the progress of other requests. When new requests arrive, the service manager determines the severity of the request and estimates the time to fix it.

The site of where the request is going to be served (US vs. Mexico) depends on the level of support; second-level support is offered in the US and Mexico, and third-level support is only offered in the US. For second-level support, the service manager or nearshore coordinator chooses the support analyst according to the location where there is an available support analyst specialized in the type of application that is being requested (see Figure 19).

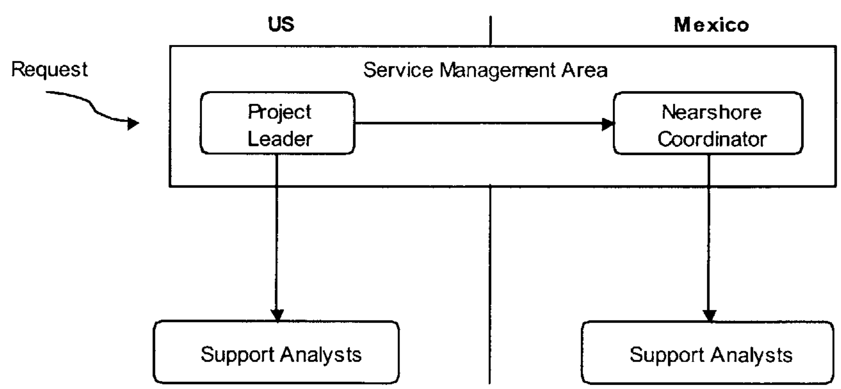

Figure 19: Assignation of AMS requests to a support analyst.

As for the differences between support analysts based in the US and in Mexico, an operations leader stated, *"In general terms, the only difference between support analysts (based in the US and Mexico) is their location. Apart from that, they have the same role, profile and capabilities..."*

After the request has been assigned, the support analyst is responsible for solving it. In some cases, only analysis of the request is required, and the customer may use this analysis to solve the issue internally or through an external provider. In other cases, the support analyst develops a solution for the request; this solution includes analyzing, designing, programming, and testing. If the support analyst has questions on the request, he or she may contact the service manager for assistance.

Once the request is served, the support analyst fills an active statement (written in English) including information of what was done, the tests performed, and the modifications made to the system. This statement is then recorded in the RMD. In an interview, a nearshore coordinator emphasized the importance of communicating in English.

"English is enforced in our internal documentation. If we communicated amongst ourselves in Spanish, we would be doing doubling our work in translating every document."

Finally, the service manager or nearshore coordinator evaluates the solution to make sure the request has been solved appropriately. In some cases, the service manager may correct the solution, perform validation tests, and modify the active statement in the RMD.

3.2.2.3 Conclusion Stage

During the conclusion stage, the support responsibilities are transferred to a new support team and the project reaches its end. This stage starts one to six months prior to the end of the project.

Role Descriptions

All the project members who worked on the project are present during the conclusion stage. In this stage, project members are given feedback on their job performance.

Conclusion Phases

a. Knowledge Transfer

The knowledge transfer phase consists of transferring the operations support activities to a new support team. As part of this process, the support team first transfers their knowledge of the business and applications that were serviced. For this, all documentation on the support services, such as the functional & technical specifications and the RMD is delivered. The new support team is trained on the operations model, and is assessed on their readiness to perform the support activities.

b. Formal Project End

In the final phase of the project, the overall results of the project are presented to the customer. These results include efficiency metrics on the services that were delivered. In addition, the customer provides feedback on all issues regarding the project.

3.3 Training and Career Development

In this section, a focus on the training programs available to Azteca Software employees for their career development is given. First, the different role categories are be presented according to the roles described in the previous section. Then, there is a description on how new employees join the company's labor force through different internship programs. Finally, two training programs for career development are described in relation to how employees advance through the role categories.

3.3.1 Role Categories[9]

Four role categories describe the different roles that are present throughout nearshore projects: managers, project supervisors, conceptualizers, and programmers. The definitions of these role categories are described below, along with their roles.

Managers: those who engage, supervise and support a project from the outside in scalability issues, training of project members, and with any other complications.

- Regional managers
- Account managers
- Operations leaders

Project Supervisors: those who supervise and manage other people in building a software system from its conception.

- Project leaders
- Nearshore coordinators
- Service managers

---

[9] These role categories were defined with partial help from Freeman and Aspray (1999).

Conceptualizers: those who conceive and sketch out the basic nature of a software system, along with the specifications and design.

- System architects (project leader)
- Analysts
- Designers
- Support analysts

Programmers: those who work on constructing and testing a software system by following specifications.

- Senior programmers
- Junior programmers
- Testers

3.3.2 Career Advancement

Figure 20 shows how Azteca Software employees may advance through the different role categories.

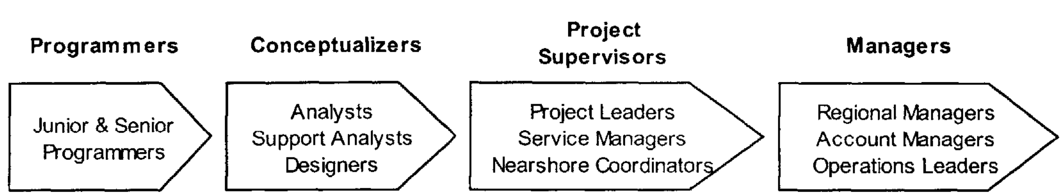

Figure 20: Career advancement through the different roles categories.

Several factors determine how an employee advances through these different categories. These factors include specialization in a particular technological profile, experience gained by taking part on projects, development of managerial skills, and time.

3.3.3 Labor Force Recruiting

New employees may join the company in two ways: year-round recruiting and internship programs. Internship programs are available to students from two-year and four-year colleges (see Figure 21).

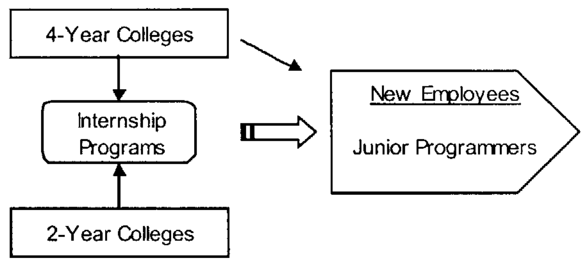

Figure 21: Labor force recruiting from two-year and four-year colleges.

When a new employee is hired, he or she has to register in a career path according to a particular technology profile. The available technological profiles at Azteca Software are: object-oriented programming languages, legacy applications, database applications, and testing.

3.3.3.1 Year-Round Recruiting

Azteca Software recruits new employees on a year-round basis. Forms of hiring at Azteca include career fairs at local universities, newspaper job ads, electronic job posts on the company's web site, and résumé collection at the company's headquarters. The majority of new employees come from local private and state universities.

The "Hiring Division" group focuses on the prospective employee's personality and aptitudes. This group also pays attention to the prospective employee's technical experience from previous course projects, as well as the employee's experience with particular computer tools, platforms, and programming languages.

The recruitment process lasts around three weeks. This process consists of a personal profile interview, exams (psychometric and English), background checks, and interviews with project leaders, operations leaders, and in 70 percent of the cases with customers.

3.3.3.2 Internship Programs

The internship programs are designed to recruit new employees before college students enter the outside workforce. The idea within the Hiring Division group is to have as much as 20 percent of the company-wide software developers composed of these students, who are considered as trainees within the company.

In these programs that last six months, trainees take the role of junior programmer, and learn the company's methodologies and technologies before becoming full-time employees. When trainees become full-time employees, they maintain their role of junior programmers.

Two-Year College Program

The two-year college program was established in 2000 as a joint venture with a local two-year state technical college. This program lasts around two years, and the successful completion of the program leads to an associate's degree in Computer Science validated by the Mexican Ministry of Education. In an interview, the director of the training and internship programs at Azteca Software stated, *"It is really rewarding watching these kids (from this program) develop their potential as employees at the company."*

Students from the local two-year state technical college apply by taking the following exams: a psychometric exam, which serves as a diagnosis to determine the student's ability to succeed in the program; a general exam administered by the local two-year college; and an English exam where applicants are expected to have around 10-20 percent of proficiency in the language. Admission to the program is heavily influenced by the attitude of the student. When a student is admitted to the program, the student takes the role of a trainee and he/she is awarded $200 per month.

The program consists of three parts. In the first part, students take English courses for two months. At the end of this session, students are expected to achieve levels of 50 percent of proficiency in the language. In the second part of the program, students take technical and laboratories courses that last a year. Students take these courses at both Azteca Software (spending five hours per day) and at the local technical college. During this part, students specialize in a particular technological profile. In the final part of the program, students join a project within the company.

During the school year of 2001, 26 students entered the program. Of these 26 students, 11 finished the program and subsequently were offered jobs and joined the company. When these students join Azteca Software, the student's salary increases to $400 per month.

Four-Year College Program

The four-year college program started in 1998 offering internships to students graduating with bachelor's degrees in Computer Science related fields from local private and state universities. This program is designed for students in their last term of their senior year, and lasts for a term. Students work as trainees for seven hours per day with a salary of $400 per month, and usually join Azteca Software upon the completion of the program

To enter the program, students have to submit their resume to the Hiring Division group. In addition, students have to take psychometric and English exams. Applicants to this program are expected to have a level of 50-60 percent in English proficiency.

During the school year of 2001, 21 students entered the program, and finished the program successfully. All of these students were offered employment and 18 accepted the offer to work at Azteca Software.

Retention Rates

In an interview, the sub director of the Hiring Division group mentioned that four-year college students were seen as more independent, more demanding towards challenges and salaries, more prepared, and with more leadership and negotiation skills. On the other hand, she mentioned that two-year college students were seen as more passive and seemed to have slower career advancement within the company. She also mentioned that average employees at Azteca Software stay from three to four years; employees from state universities stay for up to five years, whereas employees from private universities tend to stay from two to three years.

3.3.4 Training Programs

Within one year of entering the company, through year-round recruiting or by an internship program, junior programmers may then become senior programmers. These senior programmers may then become conceptualizers by working on projects and by taking technical courses at the "Corporate University." Moreover, conceptualizers can then become project supervisors and managers by developing their managerial skills. Conceptualizers can learn these skills by taking internal managerial courses at the Corporate University and by taking part in the "Project Management Program." The transitions of the role categories, along with the available training programs, are shown in Figure 22.

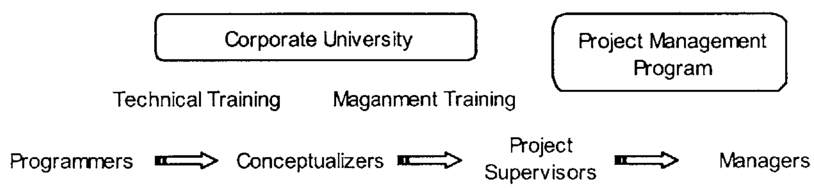

Figure 22: Career advancement through Azteca Software's training programs.

3.3.4.1 Corporate University

The Corporate University was created in 2001. In this program, employees have access to online learning materials, attend internal courses on software engineering practices, and attend daily one-hour English courses.

Technical Training

The Corporate University offers technical training to programmers. These technical courses allow programmers to become more competent in a particular technology, and prepare programmers to advance to the next role category, conceptualizers (designers and analysts).

The Corporate University facilitates employees specializing in a technological profile. Once an employee has taken online courses and tests, he or she may then complete an internal certification. In some cases, employees may pursue external certifications from leading software companies. For these certifications, the director of the Corporate University aids employees in gathering details and materials to complete the external certification.

Managerial Training

Separate from the technical courses available at the Corporate University, employees also have access to occasional internal management courses. These courses are intended for programmers who want to become part of the Conceptualizers role category, and for conceptualizers who want to transition to the Project Supervisors role category. The focus of these management courses is on software engineering practices. These practices include project management and software processes, as well as skills in team management, negotiation skills and communication.

3.3.4.2 Project Management Program

Azteca Software started the Project Management Program in 2000. This program is offered to the best group of conceptualizers who want to become project supervisors, and at some point, managers. Upon the completion of this one-year program, employees receive a certification in "Project Management in New Technologies."

To enter this program, employees have to take psychometric and English exams. Applicants are expected to have a level of around 70 percent in English proficiency. When employees attend this program, they are seen as collaborators. Moreover, the program offers the following areas of specialization: project management, web technologies, and e-business.

While enrolled in the program, collaborators take courses at a local private university meeting twice a week for four hours. In these courses, collaborators are taught the methodologies and practices of the SW-CMM. Moreover, collaborators take courses on process management. In a particular course, students provide feedback on the methodologies implemented at Azteca Software.

3.4 Analytical Summary

In this section, an analytical summary will be given on central points that have surfaced regarding AD and AMS projects and the role of its project members. These findings are posed as questions and are answered to the best of my knowledge visiting Azteca Software.

1. Where does Azteca Software get the qualified labor?

Azteca Software has two internship programs to recruit qualified labor. These two programs are aimed to people with different skills, and possibly, different career advancement opportunities. Azteca Software takes advantage of these programs in at least four ways:

- Recruit new employees before they enter the outside workforce
- Teach trainees software practices related to the company's methodologies before trainees become full-time employees
- Pay trainees lower salaries; thus, decrease the costs of projects
- Filter trainees by assessing their skills from their participation in projects

2. Which project members need to understand the system's ethos?

In both AD and AMS projects, we saw that project supervisors were the main architects of the system. They need to understand the knowledge domain and ethos of the system, and transmit this ethos to the rest of the project members. Other project members that may also understand the system's ethos are conceptualizers.

3. How much English is required for project members?

The amount of English that project members require depends on their interaction with the customer. This interaction varies across the types of projects and the different role categories.

In AD projects, English is essential for managers and project supervisors to communicate with US customers. On the other hand, conceptualizers and programmers are not required to be proficient in the language, since they mostly interact between each other and with their managers (project supervisors and managers).

In AMS projects, all project members need to be proficient in the language. The reason for this is that they need to provide support for users in the US, and constantly fill reports written in English.

4. How much formal training do project members need?

Formal training depends on the role category. Managers need to have formal training in project management. Project supervisors may be able to learn most project management techniques through internal courses and experience. Good conceptualizers need to be up to date with current technologies, which can be achieved by getting certifications. Programmers learn most their coding skills in school; however, on-the-job experience can make programmers more efficient time by learning the company's methodologies.

5. What are the differences in skills for AD and AMS project members?

In AD projects, project members need to be creative and to have the ability to apply their knowledge depending on the type of application. These project members need to understand the domain knowledge of the application so that they can design the system they are building.

In AMS projects, project members need to have a higher capacity for understanding an application in short time. Once a project member has understood the application, he or she needs to be more methodic by following specific steps to solve the problems required by the customer. In addition, project members need to be able to communicate clearly to both users and other project members.

6. What are the necessary skills for different AD project members?

The most important skill for project supervisors is to have management skills. Conceptualizers need to master the technology that is being used to build the system.

Finally, programmers need to have good software practices so that they can follow instructions from their managers.

7. How can an AD project be separated between the US and Mexico?

AD projects are separated into four phases in the development stage. Of these four phases, the inception phase needs to be done in the US. The reason for this is to have a higher understanding of what the customer wants (knowledge transfer). Once conceptualizers have understood the functional specifications of the system, the project can be brought to Mexico during the elaboration phase. Construction and transition can also be done in Mexico where project members at lower wages are available. Moreover, to mitigate communication risks, project members that were part in the inception phase need to stay until the project has been completed.

8. To what extent can AD projects depend on talented project members?

At Azteca Software, the most important element of a software development project is the quality of its project members – mostly conceptualizers and programmers. To compensate this dependency, Azteca Software created the "Software Development Process" – a reference guide for building and deploying software systems. In this guide, project members can find the steps and time frames of their responsibilities.

9. Can a project member work for more than one project?

It is not advised within the company to work for more than one project. However, this can happen in two cases. The first case arises from the notion of "core teams," which is a team of project members who are assigned to a particular customer. Within this customer, project supervisors may assign tasks to the development or support team across different projects.

The second case arises from the fact of having most of the resources at the nearshore development center, where experts (highly qualified conceptualizers) support different projects. Because of this interaction, project members are trained faster.

10. Are there teams that generally work together from project to project?

Yes. Teams composed of managers, project supervisors, conceptualizers, and programmers may form groups that work together from project to project. The fact that project members within the group already know each other and have experience in building successful applications makes this team more productive, as suggested by the Software Craftsmanship approach.

However, project members may join or leave these groups to foster communication within the company. In this manner, isolated groups are broken, and new groups are generated.

11. What are the engagement models for AMS projects?

In AMS projects, there are two engagement models for charging customers for Azteca Software services: times & materials and fixed rate. In times & materials, the customer is charged for the hours in which project members provide support.

In fixed rate, the notion of "core teams" is revisited, and the customer is charged for this group as a whole, whether the support time is used or not. When there is more workload, additional project members join the team and the customer is charged for these extra hours of support.

12. What determines a projects member's career advancement?

Technical career advancement is mainly determined by the person's aptitudes and experience. External factors that may also affect are new projects openings and the company growth rate.

Managerial career advancement, for project members wanting to become project supervisors and managers, is mainly determined a variety of different factors including: technical experience; capacity to manage large teams; negotiation, communication, and leadership abilities; and recognition within the company.

It is important to notice that there may be people within the company that can last their entire career in the same role. For example, experienced and technical capable senior programmers may have a higher salary than their project leader. However, the productivity and goals of this person is expected to be greater with time.

Chapter 4: Implications of Nearshore Development

Nearshore development represents an important opportunity for expanding the Mexican Software Industry. The demand for offshore services is conveniently located in the US. The question now is whether Mexico has the capacity to establish and maintain a software industry based on selling services to US companies. This industry would have to compete with other countries in offering high-quality offshore development services, which would include AD and AMS projects as presented in the last chapter.

This chapter outlines a possible growth scenario for nearshore development in Mexico given its current workforce, and highlights the implications derived from this growth. The first section discusses the inputs and the limiting factors for the human capital offering offshore development services. The second section describes a model for estimating the growth of the Mexican workforce in offshore development. Based on this model, the third section outlines a possible growth scenario for nearshore development in Mexico. Finally, the fourth section describes the implications of this scenario for human resources, employment, and income.

4.1 Qualified Labor and Current Workforce

The previous chapter showed that the main requirement for offering offshore services is qualified labor. The source of this qualified labor is the formal educational system composed of two-year and four-year colleges. College graduates become trainees, and then transition to different roles by taking part in formal programs and by gaining experience working on projects. With time, nearshore software developers become skilled and are able to offer high-quality offshore development services. The number of these new developers, however, depends on the current workforce, which will train new employees in offering offshore development services.

4.1.1 Sources of Qualified Labor

To compute the expected number of college graduates with degrees from computer science programs, the compound annual growth rate for students in four-year colleges from 1995-1999 was calculated. The number of college graduates at a growth rate of 10.5 percent was then calculated for each year between 2000 and 2012 (see Appendix C). Two-year college graduates were not considered for two reasons. First, data was difficult to obtain. Secondly, two-year college graduates represent a small fraction of the workforce at Azteca Software. According to the sub director of the Hiring Division group at Azteca Software, this fraction ranges from three to five percent. The calculations of the number of college graduates might be an overestimate, however, as shown later, the number of college graduates is not a limiting factor in the growth of the software industry based on selling offshore services.

4.1.2 Current Nearshore Development Workforce

Currently, there is no reliable data regarding the current workforce in the Mexican software industry. Unfortunately, the same situation applies to the workforce selling offshore development services. To estimate this workforce, we first consider expected revenues from Mexican offshore development services in 2002 – $213 million (Select, 2000). Then, we calculate the rate at which nearshore software developers are sold to customers using the Azteca Software mode for estimating the cost of projects: resources based in Mexico are sold to the customer at a rate of $20 per hour, and resources based in the US are sold at $40 per hour. Moreover, we use Azteca Software claim that in general nearshore projects, 70% of the resources are based in Mexico and 30% of the resources are based in the US. These two facts lead us to conclude that nearshore resources are sold to the customer at a rate of $26 per hour, or at around $52,000 per year. Finally, we divide 75 percent of the expected revenues from Mexican offshore services in 2002 by $52,000, which leads to a workforce estimate of 3,068 employees. The reason for taking 75 percent of the revenues is that according to Amoribieta et al. (2001), labor accounts for around 75 percent of the cot of developing software.

It is important to notice that a pay level according to role was not used in this workforce estimate. The reason for not using a pay level is that Azteca Software does not distinguishes roles when performing project costs estimates. Moreover, to have a better idea on the size of 2002 workforce, other estimates are needed. However, as shown later, the conclusions of this chapter will be mostly qualitative and will not depend on the precise workforce size.

Table 15 presents the distribution for this workforce according to its role category. This distribution was estimated by the author using an approximated individual role distribution at Azteca Software based on interviews with an operations leader and with the sub director of the Hiring Division group. Note that although trainees are presented in a different role category, they perform the roles of junior programmers. As mentioned in Section 3.3.1, project leaders are also members of the Conceptualizers role category, but will only be considered as members of the Project Supervisors role category for the remainder of this chapter.

Table 15: **Estimated 2002 workforce distribution according to role category.**

Trainees	**Programmers**	**Conceptualizers**	**Project Supervisors**	**Managers**
	Senior Programmers	Analysts	Project Leaders	Account Managers
	Junior Programmers	Designers	Nearshore Coordinators	Regional Managers
	Testers	Support Analysts	Service Managers	Operations Leaders
15%	45%	25%	13%	2%

4.2 A Nearshore Development Workforce Growth Model

The model presented in this section focuses on how employees within a role category transition to other categories. Employees make these transitions by taking part in formal training programs and by gaining experience working on projects. To build this model, transition times for each role category (i.e., the time required for programmers to become conceptualizers) and transition probabilities for these role categories (i.e., the number of programmers that become conceptualizers) are estimated according to several observations.

4.2.1 Transition Times

Table 16 shows the transition times for trainees and role categories considered in the model. These transition times were provided by an operations leader at Azteca Software according to the individual roles. As noted by the operations leader, the transition times presented in the table below are average times; some employees may take shorter or longer to transition to the next role.

Table 16: Transition times for trainees and role categories.		
	Time Spent	**Time required for a trainee to attain role**
Trainees	6 months	Not Applicable
Programmers	3.5 years	6 months
Conceptualizers	2 years	4 years
Project Supervisors	3 years	6 years
Managers	6 years	9 years

As shown in Table 16, the model presented in this section assumes a sequential transition for employees (i.e., programmers become conceptualizers before becoming project supervisors). Whether this is true or not to a certain degree is questionable. This model, however, does not suggest that every trainee becomes part of the Manager role category over time. These two observations will be taken into account in the transition probabilities.

4.2.2 Transition Probabilities

In this model, each employee within a given role category has three options: to stay within the role category, to transition to the next role category, or to leave nearshore development to other sectors of the software industry or to other industries. To estimate these transition probabilities, we need to consider the nature of the job (technical vs. managerial) required by each role category. Moreover, we need to take into account the average leaving probability for full-time employees at Azteca Software – this percentage ranges from eight to nine percent. Table 17 shows the transition matrix that captures these transition probabilities.

Table 17: **Transition probabilities for trainees and role categories.**

	Stay	Transition	Leave
Trainees	0%	85%	15%
Programmers	40%	50%	10%
Conceptualizers	62%	30%	8%
Project Supervisors	72%	20%	8%
Managers	80%	12%	8%

In the case of trainees that become full-time employees (Programmers role category) at nearshore development companies, we see a transition probability of 85 percent. This probability is based on statistics at Azteca Software for trainees that enter the company's internship program for students from four-year colleges (see Section 3.3.3.2).

For full-time employees, programmers are expected to have the highest transition and leaving probabilities. The reason for a high leaving probability is that employees first explore software development when they become programmers, and as a result, the transition from programmers to conceptualizers is the first cut for full-time employees. The reason for a high transition probability is based on the pay levels at Azteca Software: programmers receive lower salaries than employees within other role categories. Therefore, programmers will more likely transition to the next role category for higher salaries and responsibilities.

In the case of conceptualizers transitioning to the Project Supervisors role category, we expect a low transition probability. The reason for this is the shift of skills that are required: technical to managerial. When project supervisors transition to the Managers role category, more managerial skills are needed, and we expect a low transition probability for project supervisors as well. Finally, managers may transition to become high-level executives, or leave to sectors of the software industry, or to other industries.

4.3 A Nearshore Development Growth Scenario

This section presents a growth scenario for nearshore development in Mexico. In this scenario, the number of hired college graduates determines the growth rate of the nearshore development. This scenario is based on the variables presented in Section 1: the number of four-year college graduates and the current workforce along with their role distribution; and on the model presented in Section 2: the transition times and probabilities for each role category. In addition, this growth scenario takes into account the following assumptions:

1. College graduates are hired as trainees on a yearly basis.
2. Trainees last one year in their category.
3. The hiring percentage of college graduates is determined by each year's workforce.
4. Employees at nearshore development companies can only be in one role category a year.
5. Each year's workforces evolve according to the model presented in the previous section. Specifically, employees at different role categories advance to the next role category according to the transition times and probabilities specified in Table 16 and Table 17.

In 2002, 15% of the total workforce were trainees. For nearshore development to grow at a higher rate than currently, more trainees are required as inputs into labor force.

Hence, in subsequent years, I increase the percentage of trainees above the current (15%) level. This increment assumes that more trainees can be recruited from the educational system, and the supply of graduates is sufficient to support this. This increase is arbitrary, but the percentage of trainees of the total workforce never exceeds 20 percent of each year's workforce. Moreover, except for the first three years, the percentage of trainees and programmers of the total workforce does not exceed 60 percent of each year's workforce. This percentage is consistent with the distribution of trainees and programmers in the 2002 workforce. This percentage exceeds 60 percent in the first three years because there is no information on the 2002 workforce's experience. As a result, programmers that should be considered as conceptualizers are accounted in this percentage. Appendix D provides a full description of the variables, constraints, equations, and estimations for the nearshore development workforce growth model presented in this chapter.

4.3.1 Preliminary Results

The figure below shows the nearshore development workforce for the next ten years according to their role category. One point to notice in this figure is that the distribution of role categories in the first few years is ambiguous. The reason for this is the same as before; there is no information regarding the 2002 workforce's experience. In other words, the model assumes that all members of a role category in 2002 have no experience in their respective role category. However, after a few years, the impact of the 2002 workforce fades away as college graduates join the nearshore development workforce and advance through the different role categories.

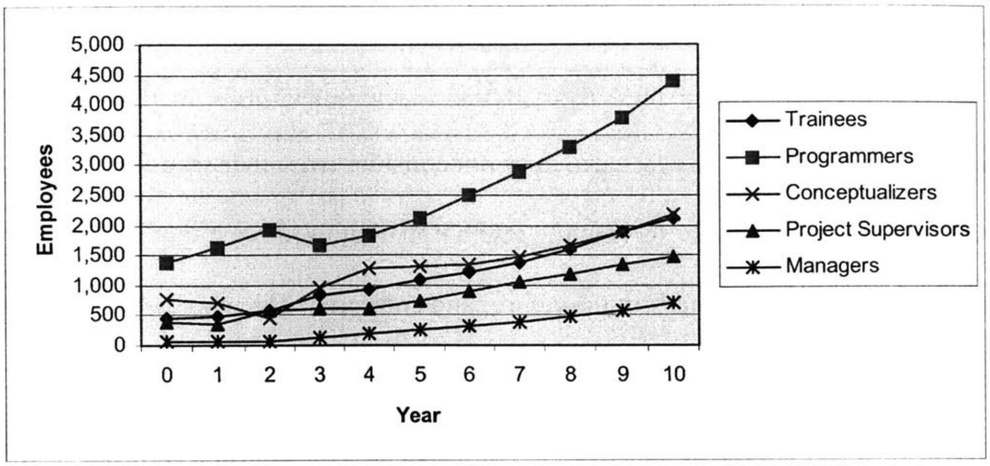

Figure 23: Nearshore development workforce for the next ten years.

Table 18 presents some important variables in the growth scenario. Not surprisingly, the number of college graduates from computer science programs increases over time. The percentage of college graduates who become trainees also increases. The expected revenues for nearshore development were calculated according to the same rationale used in Section 1: nearshore software developers are sold to the customer at $26 per hour. In addition, corrections are made to these revenues to take into account Amoribieta et al.'s (2001) claim that in software development, labor costs represent 75 percent and extra costs, such as telecommunications and traveling expenses, represent 25 percent.

Table 18: Important variables in growth model for various years.

	2002	2007	2012
College graduates	10,435	17,199	21,004
College graduates to Trainees percentage	N/A	8.0%	10.0%
Trainees	460	1,089	2,100
Workforce	3,068	5,501	10,862
Revenues (US$ million)	$213	$381	$753

As show in Figure 23 and Table 18, the workforce and revenues for nearshore development increase over time. This growth scenario, however, only considers the current workforce's ability given college graduates and other variables already mentioned. In particular, this scenario does not consider two other sources of growth: software companies currently serving the internal market in Mexico that may offer nearshore development services in the future; and entrepreneurs, who may start their own software companies offering offshore services and boost the growth in nearshore development. In this latter option, Figure 23 suggests a source of these entrepreneurs from within the workforce in this scenario: employees from the Managers role category. These managers have the business expertise and experience required to set up and run new nearshore development centers. However, it is important to notice that companies run by these entrepreneurs need to be portrayed by a different model than the one presented in the last section. In this entrepreneurship model, nearshore software companies will grow at a higher rate: employees will have to transition faster to other role categories, and with a different transition probability matrix.

4.3.2 Mexico's Percentage of the US Offshore Development Market

Table 19 presents the estimated growth of the US offshore development market, along with the percentages of Mexico and India. For 2000, revenues for Mexico were taken from Select (2000); revenues for India were taken from Arora et al. (2001). For 2000 and 2005, revenues of the US offshore development market were taken from Maroto and Zavala (2003) according to IDC projections. In later years for Mexico, revenues were taken from the model presented in the last section. For 2005, revenues for

India were estimated according to a 23 percent annual growth rate since 2000. This annual growth rate is slightly below the annual growth for the US offshore development market from 2000 to 2005: 26.19 percent. The rationale behind this estimate is that other countries will be competing for providing offshore services; thus, the India's market share will fall in future years. However, the percentage by which India's market share will fall remains questionable and other estimates for Indian revenues from offshore development are needed. From 2005 to 2010, revenues for India and the US offshore development market were estimated according to half their CAGR from 2000 to 2005: 25.19 percent and 23 percent respectively. From 2010 to 2012, revenues for India and the US offshore development market were estimated according to half their CAGR from 2005 to 2010: 12.59 percent and 11.5 percent respectively. The rationale behind halving the annual growth rate every five years is that offshore development cannot grow a high rate for longs periods. In other words, there are limiting factors to which Indian revenues and the US offshore development market are susceptible. As before, the estimates presented here are arbitrary and other forms for estimating these growth rates are needed.

Table 19: **Estimated Mexican and Indian offshore revenues.**

	2000	2005	2010	2012
Revenues (US$ million)				
Mexico	$71	$290	$571	$753
India	$3,900	$10,980	$18,922	$21,160
Percentage				
Mexico	1.3%	1.6%	1.8%	2.1%
India	70.9%	62.4%	59.4%	58.8%
US Offshore Market (US$ million)	$5,500	$17,600	$31,850	$35,988

Source: Arora et al. 2001; Maroto and Zavala 2003; Select 2000; and estimations by author.

From the table above, we observe that India will still be the dominant player in the future offshore development market, while Mexico's share will increase slightly over time. This view, however, does not consider the two extra sources of growth to offshore development in Mexico already discussed. For the rest of the market, we have to consider other countries, such as Russia and China, that are making efforts to establish software industries selling offshore services. Moreover, it is important to notice that the estimations on Indian revenues and on the US offshore development market do not affect the main conclusions of next section, and of the next chapter.

4.4 Implications for Human Capital, Employment, and Income

The growth scenario presented in the previous section draws several implications for human capital, employment, and income. Before outlining what these implications are, we observe that sources of qualified labor in Mexico are not a limiting factor in the growth of nearshore development. On the contrary, qualified labor is available for companies selling offshore services, and for software companies serving the Mexican market.

Because of the increment in the nearshore development workforce, direct jobs will be created for college graduates in computer science programs, and indirect jobs for other groups of people. The jobs created in nearshore development will be significantly better to other jobs in declining industries. First, nearshore development jobs are knowledge based, as opposed to jobs in the maquila industries. Secondly, nearshore development wages are much higher than wages from the maquila industries. For example, average wages at nearshore development centers are $26 per hour, while wages at maquilas are $3.52 per hour[10].

In addition, the jobs created from nearshore development and their generated income, would serve to create new markets for the software industry. For example, software developers will have the opportunity to become entrepreneurs in nearshore development, or explore other options that will generate a higher growth rate for the Mexican Software Industry.

Other observations worth pointing out are the implications of Mexican software companies shifting from serving the internal market to offer offshore development services to US companies. As observed in the last section, we saw that this shift is necessary for the nearshore development sector in Mexico to grow at a higher rate. Software companies serving the internal market would make this shift to generate higher revenues and take advantage of the US offshore development market. When these companies make the shift, skilled developers would be needed to compete with other high-quality companies offering offshore services. As a result, we would expect the quality of services offered in the internal market to decline, but these companies can still offer their services to both markets depending on their demand and on the capabilities that are required to offer these services.

[10] According to Smith (2002).

Chapter 5: Conclusions

This thesis studied the organization of work at a nearshore development center. The types of projects that were analyzed are Applications Development and Applications Maintenance & Support. This study was important for analyzing how these projects are organized. In particular, the objectives at different project phases, the responsibilities of project members, and the interactions of project members among themselves and with the customer as a project evolves over time. The analytical summary section in Chapter 3 showed some important aspects from the case of Azteca Software. The organization of work in offshore projects and the wage differentials then determine the advantages of going offshore.

Based on the organization of work at Azteca Software, this thesis also presented a growth scenario for nearshore development in Mexico. From this scenario, we observed that the growth of nearshore development depends on the level of entrepreneurship of Mexican software developers. That is, although there is a high demand for offshore services in the US, human capital will be the constraint for the growth of nearshore development, and entrepreneurship is needed for nearshore development to grow at a higher rate. The implications of this growth scenario were presented at the end of Chapter 4.

Most importantly, if nearshore development grows according to the model presented in Chapter 4, the Mexican Software Industry based on selling services to the US will no be able to replace the jobs left by maquilas that are leaving Mexico. From

January 2001 until April 2002, maquila employment fell by 240,000 according to Smith (2002),[11] while the growth scenario presented in the previous chapter showed that 10,862 nearshore development direct jobs would be created within ten years. Nevertheless, the quality of jobs created in the Mexican offshore industry will be significantly better than the jobs left vacant by maquilas. Software development jobs are knowledge-based and their wages are as much as seven times higher than those at maquilas.

The rest of this chapter is divided into two sections. The first section presents some lessons on offshore development from the case of Azteca Software; and the second section presents some future work suggestions for studies in the field of offshore development and in the development of the Mexican Software Industry.

5.1 Other Lessons

The lessons presented in this section are structured into three main categories: software development, offshore development, and human capital.

Chapter 2 presented two approaches to software development: software craftsmanship and software engineering. These two approaches seemed to take opposing views on how to view software development. The case of Azteca Software revealed that software firms use a mix of the two approaches. For example, similar to the craftsmanship approach, the AD project leader was the main architect of the system and

[11] The garments and electronics sector represented 48 percent of the industry's 2001 workforce – nearly 1.3 million (SE, 2001).

did most of the work. Project members, however, still followed company processes as suggested by the software engineering approach.

An important lesson in the study of offshore development projects was the need of constant communication between providers and customers. Most importantly, from the organization of work at Azteca Software, it is clear that customers in the US cannot throw medium and large projects "over the wall" to foreign locations. In other words, project members need to go to the customer's site in the US during the analysis phases and knowledge transfer periods.

In the case of human capital, we saw the need for high-skilled labor in offshore development projects. At Azteca Software, most of the new workforce came from internal programs for two and four-year college students. That is, the formal school system required on-the-job training to supplement it. Nevertheless, it would be interesting to study if informal on-the-job training could substitute the formal school system. Finally, the observation regarding the source of developers at Azteca Software raises the question of other possible sources of qualified labor. In particular, the informal educational system and other levels of the formal educational system (i.e., high schools).

5.2 Future Work

The importance of offshore development is observable from the revenues and workforce it generates. This field, however, has just started to be the topic of research studies. As a result, there are many future work possibilities that intend to have a better understanding of the field. The main studies that can be derived from this thesis are divided into four categories: organization of work at offshore development centers; models for the growth of nearshore development in Mexico; the role of the government, industry, and the formal education system in promoting the software industry; and entrepreneurship in the Mexican Software Industry.

This thesis showed the organization of work at a particular software company in Mexico; however, to have a better understanding on how offshore development projects are managed, similar studies at other leading offshore development companies are necessary. These studies should not be limited to software companies in Mexico, since most successful offshore development companies are based in other countries such as India. Moreover, studies with companies in other countries would also contribute to a comparison in offshore development practices across countries. For example, these specific studies can be used to explore the advantages, if any, of nearshore development. In addition, these studies can explore each country's potential niche market, which may be defined according to their capability and other factors including geographical proximity.

Other studies are necessary to improve the model of increasing the nearshore development workforce in Mexico as presented in Chapter 4. These studies include surveys within offshore development companies. These types of surveys would include information to have better estimates on the transition times of each role category, the probabilities in the transition matrix, the role distribution, and the pay level for different roles. Furthermore, researchers can also come up with better models to capture the growth of the nearshore development workforce.

The role of the government, industry, and the formal education system in promoting the growth of offshore development is a topic that also needs to be studied. In the case of Mexico, these studies need to focus on how these institutions can put Mexico in a better position against other countries when offering offshore development services

Finally, a topic of great interest is entrepreneurship within the Mexican software industry. This topic was derived from the conclusion of Chapter 4, which stated that entrepreneurship was the key to increase the growth rate in nearshore development. In this study, researches need to consider how current software companies serving the internal market, or new companies, can sell their services to US companies. Moreover, an entrepreneurship model needs to be analyzed to illustrate how these new companies will help nearshore development in Mexico grow at a higher rate.

Appendices

Appendix A Profiles of Employees Interviewed at "Azteca Software"

This appendix presents the profiles of the people interviewed at Azteca Software during January 2003 for the purpose of this thesis.

Person A is Vice-President of Azteca Software and director of operations for nearshore development services, where he performs the role of a regional manager. Person A has a Bachelor of Science Degree in Computer Systems Engineering and has been around in the company for almost twenty years.

Person B performs the role of an operations leader. He supervises projects in nearshore development and is responsible for strategic customer's accounts. Person B also helps outlining the company's software development processes. Person B has a Bachelor of Science Degree in Computer Systems Engineering and has been working at Azteca Software for about ten years.

Person C is a member of the "Software Engineering" group. In this position, Person C is in charge of specifying and supervising the company's practices in software development processes. Person C also teaches Six Sigma practices to software developers within the company. Person C has a Bachelor of Science Degree in Industrial Physics Engineering.

Person D is the sub director of the "Hiring Division" group. In this position, she helps regional managers and operations leaders find the human capital, internally or externally, necessary to work on projects. Person E has a Bachelor of Arts Degree in Business Administration and has been working at Azteca Software for five years.

Person E is a member of the "Training & Development" group. In this position, she outlines the training courses for software developers and aids them in finding material for certification. In addition, she supervises the career plans of these software developers. Prior to joining Azteca Software, Person E was a teacher for five years at a technical college in the north of Mexico. Person E has a Bachelor of Science Degree in Computer Systems Engineering.

Person F performs the role of project leader for Applications Development projects. He has worked in projects across different industries and knowledge domains. Prior to joining Azteca Software, Person F worked at a major software development company in the US. Person F has a Bachelor of Science Degree in Computer Systems Engineering.

Person G performs the role of a nearshore coordinator for Applications Maintenance & Support projects. He has only been around at Azteca Software for three years, but has around 10-15 years of experience in software development. Person G has a Bachelor of Science Degree in Computer Systems Engineering.

Person H performs the role of an analyst for both Applications Development and Applications Maintenance & Support projects. Person H has a Bachelor of Science Degree in Computer Systems Engineering and has worked at Azteca Software for three years. Prior to joining the company, Person H worked at other software development companies.

Person I performs the role of a designer for Applications Development projects. In this position, Person I mostly works on the graphics design of the system. Person I has a Bachelor of Arts Degree in Graphic Design and has been working at Azteca Software for two years.

Person J is a junior programmer for Applications Development projects. He has been working at Azteca Software for the past two years. Person J has a Bachelor of Science Degree in Computer Science.

Appendix B Country Evaluation Model and Compound Annual Growth Rate

This appendix presents two sources of information referred in several chapters of the thesis: Scholl et al.'s (2001) country evaluation model, and the equation for the compound annual growth rate (CAGR).

B.1 Country Evaluation Model

Table 20 shows a summarized description of the variables, measured data, and rationale used in Scholl et al.'s (2001) country evaluation model.

Table 20: **Country evaluation model criteria.**

	Variable	Measure	Rationale
Labor			
	Total population	Measured in 1999	
	H1-B visas	Offered by the US between October 1999 and February 2000	Good indicator of the supply of highly skilled individuals
	Average minimum wage	Paid per year between 1995 and 1999	Reflects the cost of labor
	Scientists and engineers in research and development	Number for the latest year available between 1987 and 1997	Serves as a measure of the overall quality of the labor pool
	Number of personal computers	Measured in 1999	Indicates the population's experience with IT
	High-technology exports	As a percentage of total manufactured exports in 1999	Provides the level of IT sophistication in a country's labor pool
Infrastructure			
	Number of Internet users	Measured in 1999	Indicates the level of connectivity
	Units of Secure servers using encryption technology in Internet transactions	Measured in 2000	Indicates the level of online business transactions taking place within and between various countries
	Information and communications technology (ICT) expenditures	As a percent of the country's GDP in 1999	Level of IT infrastructure
Capital			
	Gross capital formation	Measured as the annual average growth rate between 1990 and 1999	Countries that invest more in local industries are likely to have a more advanced IT sector
	Foreign direct investment	Measured as a percentage of gross capital formation in 1999	Measure of a countries openness to conduct business with foreign enterprises; it also measure the social, political, and economic risk
	Composite International Country Risk Guide (ICRG) rating	Country risk rating that collects data on 22 components of risk, rating between 0 and 100 (December 2000)	The higher the risk, the less foreign investment thre is flowing into a particular country

Source: Scholl et al. 2001. Sources for measured data are: the Immigration and Naturalization Service (INS, 2002), World Bank, International Finance Corporation, Organisation for Economic Co-operation and Development, International Monetary Fund, International Telecommunications Union, and the PRS Group.

B.2 Compound Annual Growth Rate

The following equation shows how to compute the CAGR between two years. CAGR is defined according to Babaie (2002) as the annualized rate of unit shipment growth between two given years, assuming growth takes place at an exponentially compounded rate.

$$CAGR_{\text{from year n to year m}} = \left(\frac{Value_{\text{year m}}}{Value_{\text{year n}}} \right)^{\frac{1}{m-n}} - 1$$

Appendix C Four-Year College Students from Computer Science Programs

This appendix presents data on the number of students graduated from Computer Science (CS) programs. The first section presents data for 1990-2000, according to ANUIES (2001); and the second sections presents estimations on the number of graduated students with degrees from CS programs for 2000-2012.

C.1 Four-Year College Graduates from CS Programs, 1990-2000

Table 21 presents data on the number of four-year college students from CS programs from 1990 to 2000.

Table 21: Four-year college students from CS programs, 1990-2000.

Year	First-Year Students	Students other than First-Year	Graduated Students	Graduated Students with Degrees
1990	14742	49232	5031	1981
1991	15793	53062	6222	2198
1992	15006	54187	8026	2840
1993	17938	56977	8501	3763
1994	19925	62958	265	N/A
1995	584	90	11185	5185
1996	24628	75847	11925	6146
1997	29778	102595	12867	6065
1998	33898	89923	13298	6543
1999	38282	100916	15577	7732
2000	42415	108532	N/A	N/A

Source: ANUIES (several years 1991-2001). N/A: Not Available.

C.2 Estimated Number of Graduated Students with Degrees from Four-Year College CS Programs, 2001-2015

The following equation presents the CAGR for graduated college students with degrees from CS programs from 1995 to 2000.

$$CAGR_{\text{from 1995 to 2000}} = \left(\frac{7,732}{5,185}\right)^{\frac{1}{5}} - 1$$
$$= 10.51\%$$

Table 22 shows the number of graduated students with degrees from CS programs growth at an annual rate of 10.51 percent.

Table 22: **Estimated number of students with degrees from CS programs, 2000-2012**

2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
8,545	9,443	10,435	11,532	12,744	14,083	15,563	17,199	19,007	21,004	23,212	25,651	28,347

Appendix D Nearshore Development Workforce Estimating Model

This appendix presents the variables, constraints, equations, and estimated results from the nearshore development workforce estimating (NDWE) model presented in Chapter 4.

D.1 Model Variables

This section presents the variables used by the NDWE model. The different variables are role categories, indices, transition probabilities, leaving probabilities, college graduates, workforce employees, "added" workforce employees, "aggregate" workforce employees, workforce distribution, and nearshore development revenues.

Role categories:

$$x = \{t:\text{trainees}, p:\text{programmers}, c:\text{conceptualizers}, s:\text{project supervisors}, m:\text{managers}\}$$
$$y = \{t:\text{trainees}, p:\text{programmers}, c:\text{conceptualizers}, s:\text{project supervisors}, m:\text{managers}, d:\text{directors}\}$$

Indices:[12]

$$i = [0:12; 2002:2012]$$
$$j = ['Incoming', '2002']$$

Transition probabilities:

xy : probability that an employee from category x transitions to category y

xl : probability that an employee from category x leaves the nearshore development workforce

Leaving probabilities:

$$\text{Average Leaving Probability for Full - Time Employees} = \frac{pl + cl + sl + ml}{4}$$

[12] Note that in the model, index 0 corresponds to the year 2002.

College graduates:

cg_i : number of college graduates at year i
ct_i : number of college graduates that become trainees at year i

cp : year growth rate of college graduates
$cp = 0.1051$

Workforce employees:

x_i : number of employees that transition to category x that stay at year i
x_i^s : number of employees from category x that stay within the same category at year i
x_i^t : number of employees from category x that transition to the next category at year i

"Aggregate" workforce employees:

$x_i^{g\text{-}j}$: number of aggregate *'j employees'* from category x at year i

"Added" workforce employees:

x_i^a : number of added employees from category x at year i
$Total\ Workforce_i^a$: number of added employees at year i

Workforce distribution:

x_i^d : distribution of added employees from category x at year i
$Total\ Workforce_i^d$: distribution of added employees at year i

Nearshore development revenues:

$Labor\ Revenues_i$: nearshore development labor revenues at year i
$Total\ Revenues_i$: nearshore development total revenues at year i

D.2 Model Constraints

This section presents a set of constraints imposed in the NDWE model. Note, however, that there are other imposed constraints by the nature of the model.

$$tt + tp + tl = 1$$

$$pp + pc + pl = 1$$

$$cc + cs + cl = 1$$

$$ss + sm + sl = 1$$

$$mm + ml = 1$$

D.3 Workforce Equations and Estimates

This section presents a set of equations describing how employees within a role category evolve over the years according to the transition times and probabilities described in Chapter 3.

$$cg_{i+1} = cg_i * (1 + cp)$$

$$t_{i+1} = cg_i * ct_{i+1}$$

$$p_{i+1} = t_i * pt$$

$$p^s_{i+1} = (p_i + p^s_i) * pp$$

$$p^t_{i+1} = (p_i + p^t_i) * pc$$

$$c_{i+2} = p^t_i$$

$$c^s_{i+1} = (c_i + c^s_i) * cc$$

$$c^t_{i+1} = (c \ + c^t_i) * cs$$

$$s_{i+1} = c^t_i$$

$$s^s_{i+1} = (s_i + s^s_i) * ss$$

$$s^t_{i+1} = (s_i + s^t_i) * sm$$

$$m_{i+2} = s^t_i$$

$$m^s_{i+1} = (m_i + m^s_i) * mm$$

$$m^t_{i+1} = (m_i + m^t_i) * md$$

Table 23 shows the evolvement of the "incoming workforces" for the next ten years. The data from this table was calculated according to the equations above using data from Appendix C and D.

Table 23: **Evolvement of the "incoming workforces" in NDWE model.**

	0	1	2	3	4	5	6	7	8	9	10
4-Year CS College Graduates	10,435	11,532	12,744	14,083	15,563	17,199	17,199	19,006	19,006	21,004	21,004
College Graduate to Trainee Percentage	0	4.5%	5.0%	6.5%	6.5%	7.0%	7.0%	8.0%	8.5%	10.0%	10.0%
Trainees	0	470	577	828	915	1,089	1,204	1,376	1,616	1,901	2,100
Programmers	0	0	399	490	704	778	926	1,023	1,170	1,373	1,616
stay	0	0	0	160	260	386	465	557	632	721	838
transition	0	0	0	200	325	482	582	696	790	901	1,047
Conceptualizers	0	0	0	0	0	200	325	482	582	696	790
stay	0	0	0	0	0	0	124	278	471	653	836
transition	0	0	0	0	0	0	60	135	228	316	405
Project Supervisors	0	0	0	0	0	0	0	60	135	228	316
stay	0	0	0	0	0	0	0	0	43	128	256
transition	0	0	0	0	0	0	0	0	12	36	71
Managers	0	0	0	0	0	0	0	0	0	0	12
stay	0	0	0	0	0	0	0	0	0	0	0
transition	0	0	0	0	0	0	0	0	0	0	0

Table 24 shows the evolvement of the "2002 workforce" for the next ten years. The data from this table was calculated according to the equations above, using data from Appendix D, from the estimated 2002 workforce at Azteca Software, and from the estimated distribution of the 2002 workforce at Azteca Software (see Chapter 4).

Table 24: **Evolvement of the "2002 workforce" in NDWE model.**

	0	1	2	3	4	5	6	7	8	9	10
Trainees	460	0	0	0	0	0	0	0	0	0	0
Programmers	1,381	391	0	0	0	0	0	0	0	0	0
stay	0	552	377	151	60	24	10	4	2	1	0
transition	0	690	472	189	75	30	12	5	2	1	0
Conceptualizers	767	0	0	690	472	189	75	30	12	5	2
stay	0	476	295	183	541	628	506	361	242	158	101
transition	0	230	143	88	262	304	245	175	117	76	49
Project Supervisors	399	0	230	143	88	262	304	245	175	117	76
stay	0	287	207	315	329	301	405	510	544	517	457
transition	0	80	57	87	91	84	113	142	151	144	127
Managers	61	0	0	80	57	87	91	84	113	142	151
stay	0	49	39	31	89	117	164	204	230	274	333
transition	0	7	6	5	13	18	25	31	35	41	50

D.4 Aggregate Workforce Equations and Estimates

This section presents a set of equations describing the aggregate number of employees within a role category at a given year. The aggregate number of employees refers to a specific workforce (incoming or 2002) within a role category at a given year.

$$t_{i+1}^{g-j} = t_{i+1}$$

$$p_{i+1}^{g-j} = p_i^t + p_{i+1} + p_{i+1}^s + p_{i+1}^t$$

$$c_{i+1}^{g-j} = c_{i+1} + c_{i+1}^s + c_{i+1}^t$$

$$s_{i+1}^{g-j} = s_i^t + s_{i+1} + s_{i+1}^s + s_{i+1}^t$$

$$m_{i+5}^{g-j} = m_{i+5} + m_{i+5}^s + m_{i+5}^t + \sum_{j=0}^{4} m_{i+j}^t$$

$$Total_i^{g-j} = t_i^{g-j} + p_i^{g-j} + c_i^{g-j} + s_i^{g-j} + m_i^{g-j}$$

Table 25 shows the evolvement of the aggregate "incoming workforces" for the next ten years. The data from this table was calculated according to the equations above using data Table 23.

Table 25: Aggregate "incoming workforces" in NDWE model.

	0	1	2	3	4	5	6	7	8	9	10
Trainees	0	470	577	828	915	1,089	1,204	1,376	1,616	1,901	2,100
Programmers	0	0	399	849	1,488	1,971	2,455	2,858	3,287	3,785	4,401
Conceptualizers	0	0	0	0	0	200	508	895	1,281	1,665	2,031
Project Supervisors	0	0	0	0	0	0	0	60	190	403	679
Managers	0	0	0	0	0	0	0	0	0	0	12
Total Workforce	0	470	976	1,678	2,404	3,260	4,168	5,188	6,374	7,753	9,223

Table 26 shows the evolvement of the aggregate "2002 workforce" for the next ten years. The data from this table was calculated according to the equations above using data from Table 24.

Table 26: Aggregate "2002 workforce" in NDWE model.

	0	1	2	3	4	5	6	7	8	9	10
Trainees	460	0	0	0	0	0	0	0	0	0	0
Programmers	1,381	1,634	1,539	811	325	130	52	21	8	3	1
Conceptualizers	767	706	437	962	1,275	1,121	827	566	372	239	152
Project Supervisors	399	367	574	602	596	738	905	1,010	1,011	929	804
Managers	61	56	53	129	178	253	328	384	468	578	682
Total Workforce	3,068	2,763	2,603	2,504	2,374	2,241	2,112	1,980	1,859	1,749	1,639

D.5 Added Workforce Equations and Estimates

This section presents a set of equations describing the added number of employees within a role category at a given year. The added number of employees refers to the total workforce (incoming plus 2002) within a role category at a given year.

$$t_i^a = t_i^{g-incoming} + t_i^{g-2002}$$
$$p_i^a = p_i^{g-incoming} + p_i^{g-2002}$$
$$s_i^a = s_i^{g-incoming} + s_i^{g-2002}$$
$$c_i^a = c_i^{g-incoming} + c_i^{g-2002}$$
$$m_i^a = m_i^{g-incoming} + m_i^{g-2002}$$
$$\text{Total Workforce}_i^a = t_i^a + p_i^a + c_i^a + s_i^a + m_i^a$$

Table 27 shows the added workforce for the next ten years. The data from this table was calculated according to the equations above using data from Table 25 and Table 26.

Table 27: Added workforce in NDWE model.

	0	1	2	3	4	5	6	7	8	9	10
Trainees	460	470	577	828	915	1,089	1,204	1,376	1,616	1,901	2,100
Programmers	1,381	1,634	1,939	1,661	1,813	2,100	2,507	2,878	3,296	3,788	4,402
Conceptualizers	767	706	437	962	1,275	1,320	1,335	1,460	1,653	1,904	2,182
Project Supervisors	399	367	574	602	596	738	905	1,070	1,201	1,333	1,483
Managers	61	56	53	129	178	253	328	384	468	578	694
Total Workforce	3,068	3,232	3,579	4,182	4,777	5,501	6,280	7,168	8,233	9,502	10,862

D.6 Workforce Distribution Equations and Estimates

This section presents a set of equations describing the workforce distribution within a role category at a given year. The workforce distribution refers to percentage of a role category over the total workforce at a given year.

$$t_i^d = \frac{t_i^a}{\text{Total Workforce}_i}$$

$$p_i^d = \frac{p_i^a}{\text{Total Workforce}_i}$$

$$c_i^d = \frac{c_i^a}{\text{Total Workforce}_i}$$

$$s_i^d = \frac{s_i^a}{\text{Total Workforce}_i}$$

$$m_i^d = \frac{m_i^a}{\text{Total Workforce}_i}$$

$$\text{Total}_i^d = t_i^d + p_i^d + c_i^d + s_i^d + m_i^d$$

Table 28 shows the workforce distribution for the next ten years. The data from this table was calculated according to the equations above using data from Table 27.

Table 28: Workforce distribution in NDWE model.

	0	1	2	3	4	5	6	7	8	9	10
Trainees	15.0%	14.5%	16.1%	19.8%	19.2%	19.8%	19.2%	19.2%	19.6%	20.0%	19.3%
Programmers	45.0%	50.5%	54.2%	39.7%	37.9%	38.2%	39.9%	40.2%	40.0%	39.9%	40.5%
Conceptualizers	25.0%	21.8%	12.2%	23.0%	26.7%	24.0%	21.3%	20.4%	20.1%	20.0%	20.1%
Project Supervisors	13.0%	11.4%	16.0%	14.4%	12.5%	13.4%	14.4%	14.9%	14.6%	14.0%	13.7%
Managers	2.0%	1.7%	1.5%	3.1%	3.7%	4.6%	5.2%	5.4%	5.7%	6.1%	6.4%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

D.7 Revenues Equations and Estimates

This section presents a set of equations describing nearshore development revenues at a given year.

$$\text{Labor Revenues}_i = \text{Total Workforce}_i * 52{,}000$$

$$\text{Total Revenues}_i = \frac{\text{Labor Revenues}_i}{3} + \text{Labor Revenues}_i$$

Table 29 shows the nearshore development revenues for the next ten years. Total revenues refer to labor revenues plus extra revenues according to Amoribieta et al.'s (2001) claim that in offshore development, labor costs represent 75 percent of total costs, and extra costs represent 25 percent of total costs.

Table 29: Estimated revenues in NDWE model (US$ million).											
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Labor Costs Revenues	$160	$168	$186	$217	$248	$286	$327	$373	$428	$494	$565
Total Revenues	$213	$224	$248	$290	$331	$381	$435	$497	$571	$659	$753

References

Agile Manifesto. (2003). Manifesto for Agile Software Development [WWW Dcoument]. URL http://www.agilemanifesto.org (visited 2003, April 9).

Amoribieta, I., Bhaumik, K., Kanakamedala, K., and Parkhe, A. (2001). Programmers Abroad: A Primer on Offshore Development. The McKinsey Quarterly, 2001 (2), pp. 129-139.

Arora, A., Gambardella, A., and Torrisi, S. (2001). In the Footsteps of Silicon Valley? India and Irish Software in the International Division of Labour (Discussion Paper No. 00-41). Stanford, CA: Stanford University, Stanford Institute for Economic Policy Research.

Asociación Mexicana de la Industria de Tecnologías de Información (AMITI) and Bancomext (2001, April). *Industria del Software: Análisis y Recomendaciones.* [WWW Document]. URL http://www.amiti.org.mx/biblioteca.asp (visited 2003, April 10).

Asociación Nacional de Universidades e Instituciones de Educación Superior (ANUIES). (2001). *Anuarios Estadísticos.* Mexico City, Mexico: Asociación Nacional de Universidades e Instituciones de Educación Superior.

Babaie, E. (2002). IT Services Market Definitions Guide (Technical Report ITSV-WW-GU-0004). Stamford, CT: Gartner Dataquest.

Brooks, F. (1975). The Mythical Man-Month: Essays on Software Engineering – 20th Anniversary Edition. Addison-Wesley.

Cusumano, M. (1991). Japan's Software Factories: A Challenge to U.S. Management. Oxford University Press.

Dedrick, J., Kraemer, K., and Palacios, J. (2001). Impacts of Liberalization and Economic Integration on Mexico's Computer Sector. Working paper, Center for Research on Information Technology and Organizations – University of California, Irvine.

De Icaza, Miguel. (2003, May). *El Impacto al sistema e-Mexico de la incorporación de software libre* [WWW Document]. URL http://primates.ximian.com/~miguel/emexico.html (visited 2003, May 15).

Eischen, K. (2002, May). Software Development: An Outsider's View. IEEE Computer, 2002, pp. 36-44.

E-Mexico. (2003). *Página Oficial del Sistema Nacional e-México* [WWW Document]. URL http://www.e-mexico.gob.mx (visited 2003, April 9).

Expedia (2003). Expedia Travel [WWW Document]. URL http://www.expedia.com (visited 2003, May 14).

Freeman, P., and Aspray, W. (1999). The Supply of Information Technology Workers in the United States. Washington, DC: Computing Research Association.

Gori, G. (2002, April 24). Fears About Microsoft Return, in Mexico. The New York Times, Section W, p. 1.

Karamouzis, F. (2002). Offshore Service Offerings Spectrum (Research Note M-15-0832). Stamford, CT: Gartner Research.

Maroto, C. and Zavala, J. (2003). *La industria del software en México* [WWW Document]. URL http://www.software.net.mx (visited 2003, April 10).

McBreen, P. (2002). Software Craftsmanship: The New Imperative. Addison-Wesley.

Naur, P. and Randell, B. (1969). Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968. Brussels, Belgium: NATO Scientific Affairs Division.

Paulk, M., Curtis, B., Chrissis, M., and Weber, C. (1993). Capability Maturity Model for Software, Version 1.1. (Technical Report CMU/SEI-93-TR-024). Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute.

Rosenthal, B. (2001, December). How the September 11 Attacks Changed Offshore Outsourcing. Outsourcing Journal.

Sargent, J. and Matthews, L. (2002). Boom or Bust: Is it the End of Mexico's *Maquiladoras*? (Working Paper No. 2002-6). Edinburg, TX: The University of Texas-Pan American, Center of Border Economic Studies.

Scholl, R., Kempf, E. and Sinha, D. (2001). Cross-Border Collaboration: A Service Aggregator Model for Offshore IT Services (Technical Report ITES-WW-FR-0109). Stamford, CT: Gartner Dataquest.

Secretaría de Economía (SE). (2001). *México y la Industria Maquiladora de Exportación* [WWW Document]. URL http://www.economia-snci.gob.mx/PRESENTA/MAQUILA/MAQUILA.PDF (visited 2003, May 18).

Secretaría de Economía (SE). (2002). *Programa para el Desarrollo de la Industria del Software* [WWW Document]. URL http://www.economia.gob.mx/?P=1128 (visited 2003, April 10).

Servicios de Estrategia en Electrónica (Select). (2000). *Aprovechamiento de las Tecnologías de Información y Comunicaciones para el Desarrollo de México.* Mexico City, Mexico: Servicios de Estrategia en Electrónica.

Sharma, S., Vollenweider, M., and Sood, R. (2000). Globalizing Services. McKinsey & Company. The McKinsey Quarterly, 2000 (4), pp. 9-11.

Sinha, D. and Scholl, R. (2002). Demand Analysis of Offshore IT Services: Executive Summary (Technical Report ITSV-WW-EX-0190). Stamford, CT: Gartner Dataquest.

Sinha, D. and Terdiman, R. (2002). Potential Risks in Offshore Sourcing (Technical Report ITSV-WW-DP-0360). Stamford, CT: Gartner Dataquest.

Smith, G. (2002, April 29). The Decline of the *Maquiladora.* BusinessWeek.

Software Engineering Institute (SEI). (2003). Software Engineering Institute (SEI) Home Page [WWW Document]. URL http://www.sei.cmu.edu (visited 2003, April 9).

United States Department of Justice, Immigration and Naturalization Service (INS). (2002, April). Report on Characteristics of Specialty Occupation Workers (H-1B): Fiscal Year 2000. Washington, DC: U.S. Government Printing Office.

Vizcaino, A. (2003, January 25). *Inicia primera fase de proyecto e-México.* Reforma.

World Information Technology and Services Alliance (WITSA). (2002). Digital Planet 2002. Arlington, VA: World Information Technology and Services Alliance.