Protecting Sensitive Data While Outsourcing Software Development Projects

by

Kevin N. Haw

(http://www.KevinHaw.com)

A CAPSTONE PROJECT SUBMITTED

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE

Master of Science In Software Engineering

FULLERTON, CALIFORNIA

MAY, 2006

Abstract

As the outsourcing of software development becomes more widespread, organizations must take efforts to protect their sensitive data. While an outsourcing partner handling portions of a project can yield great efficiencies, how can the contracting organization be assured that their trade secrets will be protected by workers that they have never vetted? How can government contractors take advantage of outsourcing efficiencies but still keep the trust of their government clients? Finally, how can an organization share data with offshore partners without violating export restrictions and compromising national security?

This paper provides an overview of how outsourcing is currently performed in the software industry, presents case studies of outsourcing where data sensitivity issues have arisen, examines the categories of risks to proprietary data, and finally presents strategies for dealing with these problems.

Acknowledgements

I would like to extend my gratitude to Dr. Dorota Huizinga, my project advisor at California State University, Fullerton, for her guidance. Additional thanks go to Dr. Bin Cong for serving as a reviewer.

There are also other individuals that assisted with specific parts of my research. Mister William Kirsanoff of the Boeing Company helped with vital direction for the discussion of Export Restrictions in section 4.1.5. David Morris, the County Attorney of Livingston County, New York was kind enough to clarify the status of the data compromise incident discussed in section 3.2. Lorrie Smith, Legislative Director for Assemblymember James Brennan of the New York State Assembly and Jennifer L. Reinke of the Municipal Law Resource Center at Pace University also assisted with my research on the Livingston County incident. Their help improved this paper greatly.

I am also grateful to the entire faculty and staff of the Software Engineering department for developing this innovative and challenging course of study. I am indebted to Dr. David Falconer for informing me of the program while it was still in development.

Finally, I would like to thank my wife, JoAnn, for her support, patience, and love in the face of all those lost hours.

Keyword List

COTS	Acronym for "commercial, off the shelf." Refers to existing, commercially available products that may be an alternative to commissioning a new project in house or via an outsourcing provider.
Data Accountability Process	Any process of tracking custody of proprietary data to specific individuals. See section 5.2.2.
Data Contamination	Occurs when data that should be partitioned is introduced into a system or area where it should not be kept. See section 5.2.2 for strategies to address this issue.
Data Partitioning	The practice of isolating data by use and sensitivity and then restricting access to it. See section 5.2.1.
GSDOP	Generic Software Development Outsource Process. See section 2.
Marking Plan	Guidelines for including markings on documents that allow their contents to be identified as proprietary.
Offshore	Activity taking place beyond the national boundaries of an outsource client. Usually used in the context of an outsource provider employing workers in foreign countries for economic or other business reasons (e.g. lower wages or leveraging time zone differences to provide around the clock support).
Outsourcing	The provision of goods or services by third party specialists in direct or indirect exchange for money[1]
Outsource Client	An organization employing an outsource vendor to perform service on its behalf.
Outsource Vendor	A firm providing specialized services for organizations that do not wish to perform those with their own employees for reasons of cost or efficiency. Also called "Outsource Provider."
Project Plan	The combined schedule, budget, and statement of work of a project that defines how work on a project is to proceed.
Proprietary Data	Data developed or gathered by an organization that confers a competitive advantage over other organizations. A specific type of sensitive data.
Requirement Sterilization	Structuring requirements to an outsource vendor such that sensitive data parameters are masked. See section 5.2.3.
RFI	Request for Information
RFP	Request for Proposal
Risk	The probability that a threat occurs. As such, it is possible to manage risk by taking preventative measures even though it is not possible to manage threats.
Rule of Separation	A security principle in which one separates sensitive data from the threats around it.
SOW	Statement of Work

[1] C. Warren Axelrod, *Outsourcing Information Security*. (Boston: Artec House, 2004), 1.

Sensitive Data	Data that, if exposed, can cause damage to an organization.
Service Provider	See “Outsource Provider”
Third Party Data	Sensitive data provided by a party other than the outsource client, usually either customer data or proprietary data provided under a nondisclosure agreement from another company. See section 4.1.4.
Threat	An adverse event that occurs during the course of a project. In the context of this paper, this is primarily used to refer to the compromise of proprietary data.
TPS Strategy	Acronym for “Track, Partition, Sanitize.” Describes three interrelated strategies for the management of proprietary data.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Mercenaries... are useless and dangerous; and if one holds his state based on these arms, he will stand neither firm nor safe; for they are disunited, ambitious, and without discipline, unfaithful, valiant before friends, cowardly before enemies; they have neither the fear of God nor fidelity to men... they have no other attraction or reason for keeping the field than a trifle of stipend, which is not sufficient to make them willing to die for you.

-- Niccolò Machiavelli, *The Prince*[2]

As the practice of outsourcing software development efforts becomes more and more prevalent, a continuing problem has emerged: How can an organization protect its interests while making use of the skills and knowledge of employees that are, by definition, beyond their control?

This is hardly a unique dilemma. In 1513, when Machiavelli issued his dire warning, Italy was a disunited sea of warring city-states. While rulers preferred to use armies made up of their citizens, who had a vested interest in the survival of their kingdom, often they were forced to employ mercenaries to supplement these loyal troops. As often as not the reason (or rationalization) for this practice was that it was needed to counter similar actions by aggressive neighbors.

This argument is repeated today as many companies justify outsourcing work that would traditionally have been done in house: It is a matter of survival, not preference. And, while comparing the millions of dedicated professionals that work on outsourced projects to mercenaries may not be particularly fair, Machiavelli's concerns about loyalty are as pertinent to the Information Age as they were to the Renaissance. The fact is that outsourced work is performed by individuals that are not directly hired by the client, managed by the client, or held directly accountable to the client. Regardless of personal standards of honesty and integrity, these workers do not owe any direct allegiance to the organization commissioning their labor.

Any leader worthy of the name will find themselves raising disturbing questions when the decision to outsource is raised. Do these people understand our business well enough to be of value to us? Will their work be of the highest quality? Will our customers be satisfied with this decision or will they take their business elsewhere?

Of all the nightmare scenarios that can erupt from this situation, though, perhaps none is so disturbing as the compromise or loss of that most treasured commodity of the Information Age: Proprietary Data.

Whether it is a customer list, the source code to a flagship software product, or a strategic business plan, data is the lifeblood of the modern company. With the advent of the just in time, globalized economy, dominance of one firm over another, if not an organization's very survival may hinge on the thin advantage created by the value of this commodity.

Sir Francis Bacon said, "Knowledge is power[3]." Although he had a greatly different view of the world than that of Machiavelli, it would appear that his wisdom also rings as true today as in the 16th century.

It is easy to see that if sensitive data is compromised, if an organization looses the competitive edge Bacon described, the results might be catastrophic. Should this occur during an outsourcing effort, all the advantages that outsourcing promises can be erased in an instant. The goal of this

[2] Niccolò Machiavelli, *Il Principe [The Prince]*. 1513.

[3] Sir Francis Bacon, *Meditationes Sacræ. De Hæresibus* [*Religious Meditations, Of Heresies*]. 1597.

paper is to examine ways in which we might pursue an outsourcing strategy while protecting our precious data.

To meet this goal, this paper will begin by providing an overview of how outsourcing is currently performed in the software industry. It will then present case studies of outsourcing where data sensitivity issues have arisen. Then it shall examine the categories of risks to sensitive data and finally present strategies for dealing with these problems.

2 Overview of a Generic Software Development Outsource Process (GSDOP)

To better understand the risks posed to sensitive data, it helps to understand the process by which one might go about outsourcing a software development effort. However, since every project is different and every practitioner has his or her opinions on the best approaches, it is an impossibility to define a definitive process in this small space. Instead, a generic framework will be presented to allow us to continue our discussion of risks and strategies. Portions of the process where the risks to proprietary data are pronounced are emphasized while other areas are not. While this somewhat skews the view of the process, it ideally suits our purposes.

Structurally, we shall use Marks' model[4] to establish the various process phases for outsourcing: Planning and Analysis, Design, Implementation and Operation, and Termination. In addition, we shall lean heavily on Axelrod's discussion[5] of the topic at numerous points. Finally, we shall draw from both sources to create the initial phase of our process, the Decision to Outsource.

An overview of these phases can be seen graphically in Figure 1 as the Generic Software Development Process (GSDOP).

[4] Gene Marks, *The Complete Idiot's Guide to Successful Outsourcing* (New York: Penguin, 2005), 185-189.

[5] Axelrod, *Outsourcing Information Security*, 163-170.

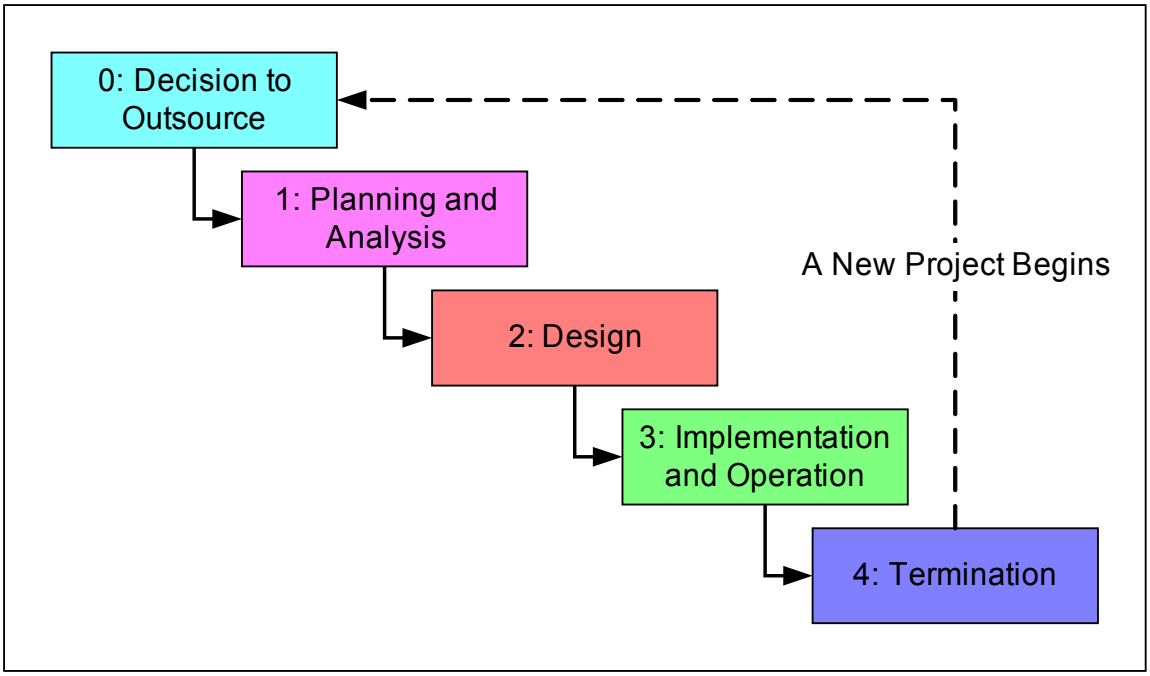

Figure 1 – Phases in a Generic Software Development Outsource Process

Before detailing the phases of our GSDOP in detail, it is interesting to take a moment and note the similarities between it and Royce's classic Waterfall Process Model of software development[6], shown in Figure 2. Where the end product of the Waterfall is a piece of software, though, the GSDOP produces something more intangible. Specifically, it creates an ongoing relationship between the client and an outsource vendor. The software itself is produced by the partnership between the two parties, making it a second-degree artifact of the GSDOP. See section 2.2 for a mapping of steps in the GSDOP to steps where software is produced.

[6] W.W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques", *Proceedings, WESCON*, August 1970: 328-338.

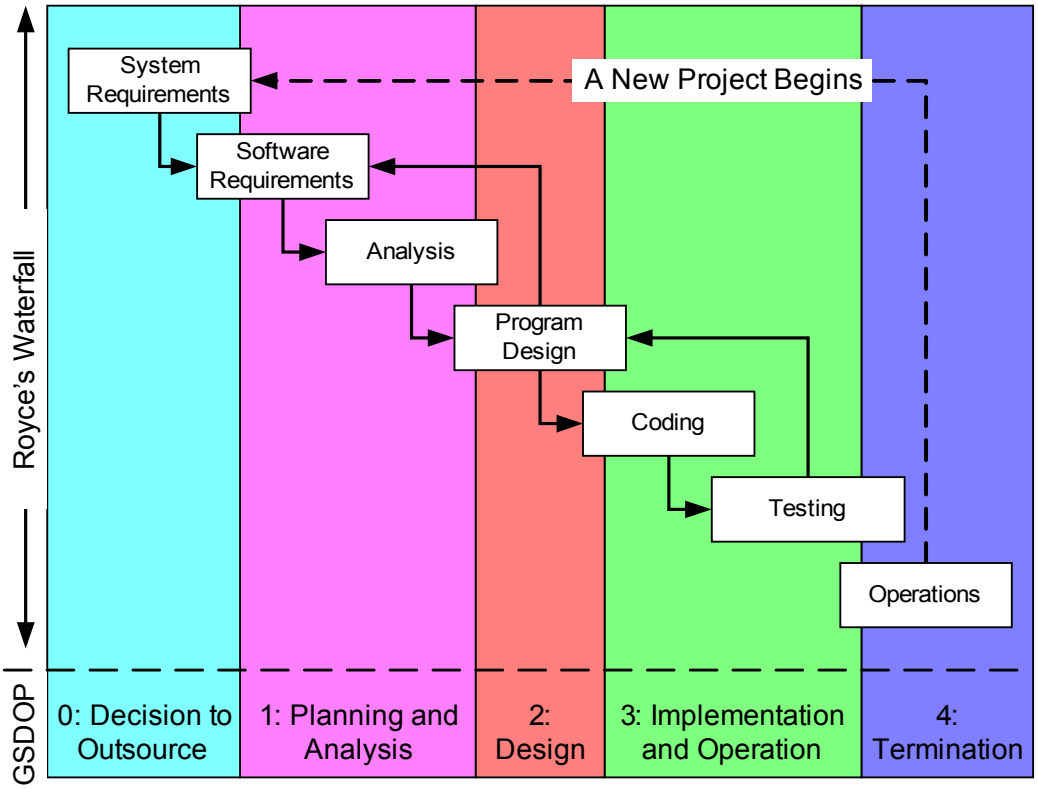

Figure 2 – GSDOP vs. the Waterfall Process Model

In sections 2.1.1 to 2.1.4, we shall revisit this analogy, using it to better explain how outsourcing is performed.

2.1 Phases of the GSDOP

As seen in Figure 1, there are five phases to the GSDOP:

- Phase 0: Decision to Outsource
- Phase 1: Planning and Analysis
- Phase 2: Design
- Phase 3: Implementation and Operation
- Phase 4: Termination

We shall now discuss each of these phases in detail.

2.1.1 Phase 0: Decision to Outsource

Despite its many benefits, not all tasks are suitable for outsourcing. Therefore, the first phase of the GSDOP is the determination of whether or not outsourcing is a viable alternative to performing the task in house (making the term "Phase 0" appropriate from an outsourcing standpoint). This is by no means a trivial matter, as it is essentially a formal bid and purchasing

process, the full scope of which has been more fully addressed elsewhere. For our purposes, we shall present several simplified steps:

1. *Define Project Scope* – Here, a case is presented to justify the need for the project, defining the end software product in the broadest terms possible. While primary the goal is simply to create a working document to support later steps, this step also acts to capture high level system requirements of the actual software product (see Table 1 for a mapping of phases of software development the phases of the GSDOP).
2. *Identify Service Providers and COTS Alternatives* – In this step, an analysis is performed to see if there exist outsource providers that can feasibly perform the project. In addition, existing Commercial, off the Shelf (COTS) or Open Source products are analyzed to see if they might be adapted by in house staff in lieu of commissioning a new product. See section 5.3 for a discussion of these alternatives.
3. *Determine if Outsourcing is an Option* – While this paper discusses ways to mange the risks that outsourcing presents, there may well be some functions that an organization should never outsource. While the decision of where to "draw the line" is obviously a subjective one that needs to be determined on a case by case basis, an organization needs to decide what functions define their core competencies and resist sending them out of house. Even if all safeguards are in place to ensure that sensitive data is not compromised, customers might still wonder at a decision to outsource too much. After all, how can a company be considered the leader in their field, can be trusted to give value to their customers, if all they do is package up incoming work and send it off to the lowest bidder?
4. *Generate and Distribute RFIs/RFPs* – Formal Requests for Proposals (RFPs) defining the project in more detail are developed and distributed to the outsource providers to invite them to bid. If the project is of a very large scope, an earlier cycle of Requests for Information (RFIs) may be performed to allow outsource providers to provide some input in defining the later, more formal RFPs. In such cases, intermediate contracts with multiple bidders may be supported to slowly reduce down the field over time rather than a single selection as shown in step 2.1.1, below.
5. *Select Best Solution* – Once all responses to RFPs are received, an analysis is performed to compare the responses against each other, the COTS alternatives defined above, and keeping the work in house. Only when this analysis is completed can the best course of action be followed.
6. *Award Contract* – Assuming step 2.1.1 resulted in a decision to outsource the software development effort, the remaining details of the Statement of Work (SOW) are resolved with the winning bidder and the contract is finalized.

This phase of the GSDOP can be considered analogous to the System Requirements phase in the Waterfall model, with the broadest definitions being established. For the GSDOP, these definitions are the process parameters specified during RFI/RFP development and form the foundation of the later steps of the process, just as System Requirements are used to define software in Waterfall.

While the decisions made in this phase are vitally important for the protection of sensitive data later in the project, there is actually very little risk of compromise at this point. The sharing of data is kept limited to small teams directly involved in the bidding process. There is ample precedent for the protection of this data via nondisclosure agreements from traditional contract award processes. As the GSDOP progresses to further phases, however, the risks gradually increase.

2.1.2 Phase 1: Planning and Analysis

If the previous phase is analogous to System Requirements in the Waterfall model, then the Planning and Analysis phase is the equivalent of Software Requirements, since the specific requirements of the "product" (here, the outsource relationship) are being developed. Once the outsourcing vendor is selected and the initial contracts are in place, both parties start developing a framework by which the rest of the process can proceed.

In this phase, the following steps are performed:

1. *Assign a Dedicated Project Manager* - A dedicated project manager from the client organization is assigned to oversee the rest of the process in detail. This individual is responsible for seeing that resources are allocated correctly, that the progress of the outsource vendor is monitored, and that two-way communication is maintained. This individual must be held accountable for the success or failure of the entire effort.
2. *Create Initial Project Plan* – If the RFP and SOW developed in Phase 0 described the desired end state of the process, the Project Plan describes how to get there. This is a step by step plan of action, with responsibility for each step allocated between the two parties. The software development methodology[7] is decided upon, interdependencies between the steps are documented, and lists of deliverable artifacts for each step are developed here. A rough schedule may also be developed at this stage.
3. *Identify Stakeholders* – Although some stakeholders have been identified during Phase 0, the client organization must establish a comprehensive list of internal stakeholders and external stakeholders (e.g. their customers, vendors, or even other outsource partners on allied product lines). It is important to identify these stakeholders by their *role*, so that if a specific individual leaves the company or otherwise moves on it will be easier to find a replacement.
4. *Identify Risks and Define Contingency Plans* – Here, the risks of failure of the outsource effort, such as schedule or cost overruns, need to be defined. Contingency plans for each of these risks are developed as well[8]. In our analogy between the GSDOP and the Waterfall, this step can be seen as being more akin to Royce's Analysis phase than Software Requirements. Nevertheless, it is placed here for clarity's sake.

In the context of the risk to sensitive data, the Planning and Analysis phase represents the first real challenge of an outsource relationship. The establishment of an ongoing working relationship between a client and an outsource vendor increases risks substantially. This occurs for two reasons.

First, the volume of data traffic increases as work begins, thus exposing more opportunity for compromise. Second, the staffing levels begin to increase for both parties, thus expanding the pool of individuals that may intentionally or accidentally expose the data. Finally, the character of the data being exchanged, being that it is of a more specific, "nuts and bolts" nature, is more prone to be sensitive.

With all these factors in place, the risks increase dramatically.

---

[7] CMMI Product Team, *Capability Maturity Model® Integration (CMMISM), Version 1.1*, 337.

[8] Note that part of this step is the assessment of proprietary data compromise and the development of contingency plans to deal with them. This is detailed in section 5.2.

2.1.3 Phase 2: Design

In our comparison between the GSDOP and the Waterfall process, the Design Phase is akin to the Waterfall's Design phase. Here, however, instead of designing a software product, we are designing a software process.

In this phase, the following steps are performed:

1. *Specify Plan Details* – A detailed Project Plan is built by clarifying the specifics of the tasks developed in Phase 1. Milestones are set and specific deliverable artifacts are agreed upon. The durations of these tasks are then used to develop detailed schedules. It is vital in this step to eliminate all unspoken assumptions about task allocation, responsibility, and deadlines so as to prevent unpleasant surprises later.
2. *Assemble Internal Team* – With the Project Plan developed, the client now assigns personnel to provide management and technical oversight and coordination for the rest of the project.
3. *Allocate Internal Resources* – Here, internal client resources such as computer or prototype hardware for testing, software licenses, or server space for source code repositories are allocated and placed at either the client or vendor's site. New assets may also be purchased. Throughout the project, it is important to keep a close inventory and use a property tracking procedure so that the ownership of all equipment is clear when the project is terminated.
4. *Finalize Budget and Schedule* – With a detailed Project Plan completed, tasks assigned, and resources allocated, the budget and schedule should be finalized. Both should be compared against earlier estimates to check the accuracy of the early planning steps.
5. *Set Up Reporting Mechanism* – In addition to a mechanism for cost and status reporting, methods of accounting for tracking proprietary data must be established. See section 5.2.2 for details on how this might be done.

The Design Phase deals primarily with the development of the process for building software, as the highest level system requirements were developed in earlier phases and the bulk of the actual software development occurs in later phases. As such, most of the proprietary being shared is of the strategic variety discussed in section 4.1.1 rather than specifics about the product itself as in sections 4.1.2 and 4.1.3 or third party data in 4.1.4. Nonetheless, it is important for both parties to remain diligent and prevent exposure of this data, as its compromise can be even more catastrophic than the loss of other data.

2.1.4 Phase 3: Implementation and Operation

This is the step where the actual work begins and the outsource vendor begins the bulk of the software development work. Requirements, design, code, documentation or other artifacts called out in the Project Plan are delivered (see Table 1 for a mapping of phases of software development to this part of the GSDOP).

In terms of our Waterfall analogy, this phase (with one exception, below) is the equivalent of the Coding phase.

In this phase, the following steps are performed:

1. *Finalize Design* – The final details of the Project Plan are determined and any loose ends uncovered in the Design Phase (e.g. a particular specialist that was expected to be utilized is unavailable, timelines need to be adjusted for changes to the business case, etc.) are resolved. Extending our Waterfall analysis, this step would be a good time to perform a "requirements trace," with the resolution of every issue, artifact, and milestone found in the earlier phases uncovered and documented here.

2. *System Testing* – In this step, one or more dry runs of the procedure are performed to see how things go. In a software development environment, this might take the form of working out a few trouble reports and then performing a trial build and release, monitoring the result for any problems. In the Waterfall analogy, this step would be the equivalent of the testing phase, as we are testing the product of the GSDOP (here, the outsourcing relationship instead of a piece of software). Any issues discovered here should be examined and once resolved should cause a change to the Project Plan, as addressed in the step above.
3. *Execute* – This step is the goal of the entire outsourcing effort: the production of actual software artifacts. The details of this procedure are laid out in the Project Plan.
4. *Evaluate and Measure* – Any mature software development process has a provision for the collection of metrics and evaluation of the results. This step reflects that realization.

This step forms the bulk of the overall project schedule, sees the highest staffing levels, and results in the bulk of the technical interchange for the effort. As a result, it is also the step where the greatest risk to proprietary data is encountered. Efforts to protect these assets as described in section 5 must be in full effect or catastrophe may result.

2.1.5 Phase 4: Termination

In this phase, work on the project is shut down per the terms defined in the contract and Project Plan. Any outstanding issues are resolved and loose ends tied up. In our Waterfall analogy, this phase is the equivalent of the Operations Phase.

The GSDOP Termination Phase consists of the following steps:

1. *Shut Down Operations* – After delivery of all artifacts is performed, a final accounting of all property and proprietary data should be performed. Security badges and keys should be returned to terminate physical access to facilities and any electronic accounts deleted. A wholesale changing of locks or system passwords at this point by the client would also be prudent.
2. *Verify Contract Fulfillment* – Ideally, a series of audits should be performed to verify compliance with all the terms laid out in the contract and allow final financial settlements to be calculated and paid out. In a well-structured Project Plan, this should be a mere formality with earlier acceptance testing and property accounting satisfying terms of the contract to the satisfaction of all parties. A failure to settle these issues, however, can lead to dissatisfaction or even legal action.
3. *Offer Reference* – Assuming the project outcome was positive, the client might consider providing an unsolicited reference for the service provider to use with other clients in the future. Not only will this help goodwill with the vendor, but it allows the client to set the terms of the letter of reference and not have to revisit the issue at some later date.
4. *Move on* – From this point, there are three possible outcomes:
    - If further maintenance for the product is required (i.e. the maintenance period on the original project expired or none was included in the Project Plan), then a new, dedicated maintenance project should be put out to bid and the GSDOP process should be repeated from Phase 0 (section 2.1.1) onwards. The project may wind up awarded to the incumbent service provider, a new provider, or brought back in house as the business case supports.
    - New projects may be developed under the GSDOP for upgrades to the product or for entirely new products. As above, the final location of this work must be supported by the

business case. Section 2.3 discusses models that might be used instead of the GSDOP that take this repeating cycle into account as part of the process.

- Finally, if the project was a failure, either because of the events during development or due to outside factors (i.e. changing market directions), then the product is abandoned. In catastrophic cases, the entire line of business may also be abandoned. Whether this occurs or not, care must still be taken to shut down operations and provide a full accounting of all equipment as described above.

While the Termination Phase sees activity on the project "winding down," there is still substantial risk to proprietary data. Source code, design documents, and user manuals are assembled for easy access, presenting a tempting cache of sensitive data. Combined with the possibility of complacency, this presents many opportunities for data compromise.

2.2 Mapping GSDOP Phases to Software Development

As noted in section 2 (and detailed throughout sections 2.1.1 to 2.1.5), it is possible to map the various phases of the GSDOP to the Waterfall model. However, the mapping of what actual software development occurs at each step of the GSDOP was not explicitly summarized. Table 1 is presented in order to eliminate any confusion on that matter.

Table 1—Software Development Activities During the GSDOP.

GSDOP Phase	*Waterfall Equivalent*	*Software Development Phase*
0: Decision to Outsource	System Requirements	None
1: Planning and Analysis	Software Requirements, Analysis	Basic System Requirements (as needed to develop Project Plan)
2: Design	Program Design	None
3: Implementation and Operation	Coding, Testing	Requirements, Design, Code, Test, Maintenance
4: Termination	Operations	Some Maintenance Operations (see below)

As can be seen, the bulk of software development is actually performed in Phase 3, Implementation and Operation. However, some initial work for the System Requirements is performed early on in the Planning and Analysis phase, as discussed in section 2.1.2. If specified as part of the contract, then Maintenance is also performed as part that phase. However, if a separate maintenance project is put out to bid following the initial development project, then some Maintenance activities will need to be carried out to support the contract during the transition. In those cases, it pays to structure the initial contract terms to allow continuation of service while the changeover is occurring.

2.3 Other Model Paradigms

Throughout section 2.1, we drew analogies between the GSDOP and the Waterfall. In addition to the similarities in the specific steps, the two models are more or less identical in structure and

share a number of weaknesses. Both processes are highly linear and rigid, assuming a minimum of interaction between the various phases. They also assume that the scope of the work is known early on and will not change drastically during the process. In the arena of software development, this has led to concerns about the suitability of the Waterfall model and the development of new paradigms such as Spiral Development[9] or Extreme Programming.

It is possible to adapt the GSDOP to these other paradigms and gain some of their advantages. For example, Figure 3 represents how one might "wrap" the phases of the GSDOP over a spiral model, repeating the individual steps of the GSDOP with each iteration of functionality. While the approach would require adaptation (e.g. it would be inefficient to end a contractual relationship or perform the other steps in the Termination phase of the GSDOP every time a new level of incremental functionality is released), it still poses some intriguing possibilities.

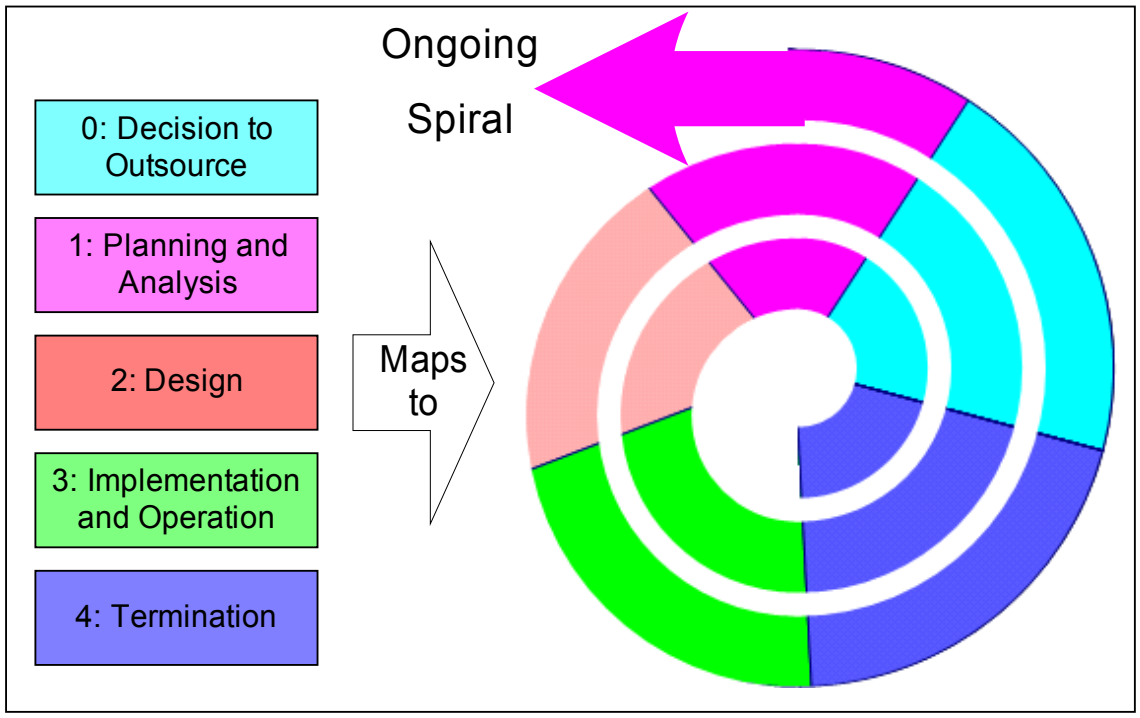

Figure 3 – The GSDOP adapted to Spiral Development Paradigm

It is interesting to speculate what the future holds for successors of the GSDOP. Much as the Waterfall represents the state of the art of Computer Science's infancy, the GSDOP represents the state of the art of an infant software outsourcing movement. One can only guess what will be seen as the field matures.

3 Case Studies

To understand the potential costs of the compromise of sensitive information, we shall examine three high profile incidents and their ramifications.

[9] Barry Boehm, "A Spiral Model of Software Development and Enhancement." *ACM SIGSOFT Software Engineering Notes*, vol 11, no 4 (1986): 22-42.

3.1 Case Study 1: Chinese Ballistic Missile Accuracy

From an engineering standpoint, the technology required to launch communications satellites into orbit and to deliver nuclear weapons with an ICBM (Intercontinental Ballistic Missile) are very similar. Both technologies take a payload and deliver them above the earth's atmosphere in a controlled manner. In the case of a satellite, the goal is to place the object in a long term orbit above the earth. For a ballistic missile, the goal is to land the payload at a specific target on the earth's surface where a nuclear weapon will detonate. Both endeavors require a high degree of precision and mastery over any number of arcane aspects of rocketry in order to develop a reliable, effective system.

In February 1996, a Chinese rocket carrying a satellite of U.S. origin exploded during launch. In the aftermath, Loral Space and Communications Ltd. and Hughes Electronics Corp. performed an analysis of the accident. This analysis was shared with China Great Wall Industry Corporation, which was responsible for the launches. As a result of this and other assistance, the People's Republic of China went from three highly publicized explosions of rockets between 1992 to 1996 and an inability to place satellites in the correct orbit to a string of 20 consecutive, successful satellite launches starting in 1996[10]. Afterwards, at least three classified studies reportedly found that U.S. national security was harmed as a result[11]. In 2002, Loral agreed to pay $20 million in fines to settle a federal investigation in the incident[12]. In 2003, Hughes and Boeing Satellite Systems agreed to pay a record $32 million in penalties[13].

In this case, a commercial partnership with a foreign company to launch civilian satellites had a direct military impact on the security of the United States. In addition, it resulted in record fines for some of America's largest aerospace companies.

3.2 Case Study 2: Sensitive Data on Children Posted on Public Website

In 2004, a subcontractor working on behalf of the New York State Office of Children and Family Services posted the names, birthdays and daily whereabouts of hundreds of children to the Internet, where the information remained publicly available for weeks until reporters investigating the incident notified authorities. In addition to the personal information already mentioned, a memo field in the database used by the Livingston County day care center chronicled each child's daily routine (i.e. "M,Tue. & Fri when mother attends treatment program and therapy,(approx. 20 hrs per wk)")[14].

The breach was the result of a chain of outsourcing for work on the database. The programming work was originally been outsourced to Genesee Community College. From there it was assigned to a third party consultant, who posted the entire database on the rentcoder.com website, seeking someone to help with a difficult formatting problem in the database. While the work was eventually subcontracted to a New Jersey-based programmer (the fourth link in the chain), the damage had already been done.

---

[10] Shirley A. Kan, "China: Possible Missile Technology Transfers from U.S. Satellite Export Policy – Actions and Chronology. Updated September 5, 2001," 4. http://www.fas.org/spp/starwars/crs/98-485.pdf.

[11] Ibid, p. 2.

[12] Christopher Marquis. 2002. Satellite Maker Fined $20 Million in China Trade Secrets Case. *New York Times*, January 10, sec A, p. 6.

[13] Jeff Gerth. 2003. 2 Companies Pay Penalties For Improving China Rockets. *New York Times*, March 6, sec A, p. 9.

[14] Bob Sullivan, "Government agency exposes day-care data: Daily whereabouts of hundreds of children posted on public Web site." MSNBC, Feb. 8, 2004, http://www.msnbc.msn.com/id/4186130.

Thankfully, the fallout from this particular incident was limited. Livingston County officials quickly informed the children's parents of the incident, but none reported back any untoward consequences[15]. The contractor, however, no longer performs work for the county[16] even though he remains active on rentacoder.com[17]. There were no civil actions or criminal penalties for any of the parties involved. This incident and others, though, have helped spawn legislation in 26 states requiring affected persons to be notified in the event of a security breach which results in, or reasonably may result in, the unauthorized acquisition of unencrypted personal information[18].

3.3 Case Study 3: Data Compromise at Bagram Air Base

In 2005, an investigation by the Los Angeles Times revealed that workers at the Bagram air base in Afghanistan had been selling USB "Thumb Drives" stolen from Coalition personnel at local flea markets. While the hiring of local workers at a foreign military base always entails the risk of theft of easily concealable valuables such as personal jewelry or currency, the ability to store large amounts of data in such a small package adds a whole new dimension to the issue. Data recovered on drives obtained by the Times included the names, addresses and photographs of Afghan spies providing information to U.S. Special Forces[19], documents that named militants targeted for attack and identified Afghan officials suspected of corruption[20], and the Social Security numbers of hundreds of soldiers, including four generals[21]. Many of the documents were marked "Secret," the second highest classification in the United States government[22].

In this instance, we have seen how easy it is for sensitive data to be lost when trusted personnel are not trained in the proper use of new technologies. While some files had been deleted, they were easily recovered using software freely available on the Internet. Had troops at Bagram been aware of this, perhaps they would have kept better physical control of these devices or used encryption to keep them secure.

4 Categorization of Risks

As we can see from our case studies, the compromise of sensitive data can take many different forms. To better understand the nature of the risks incurred during an outsourced software development effort, we need to classify them into broad categories. However, there is no obvious way to define these categories. One might choose classify risks as being entirely internal to an

---

[15] Phone interview with David Morris, Livingston County Attorney. May 2, 2006.

[16] Ibid.

[17] On May 7, 2006, http://www.rentacoder.com/RentACoder/SoftwareBuyers/showBuyerInfo.asp?lngAuthorId=74675 showed the individual's last login as being May 4th. In addition, it shows that he is highly regarded, earning a 9.98/10.0 rating by coders suing the site. Even Mr. Morris referred to the individual as "He probably does good work, but just wasn't on the ball in this case."

[18] Jennifer L. Reinke, "New York's Information Security Breach and Notification Act," *NYSBA/MLRC Municipal Lawyer*, vol. 20, no. 1 (2006): 19.

[19] Paul Watson. 2006. Leaks of Military Files Resume. *Los Angeles Times*, April 25, sec A, p. 1.

[20] Los Angeles Times Editorial Board. 2006. Flea-market secrets. *Los Angeles Times*, April 14, sec B, p. 14.

[21] U.S. Computer Files Remain on Market in Afghanistan. 2006. *Los Angeles Times*, April 15, sec. A, p. 14.

[22] Wikipedia contributors, "Security clearance," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Security_clearance&oldid=50053968 (accessed April 29, 2006).

organization, such as security breaches as discussed in section 4.1.7, and those that impact third parties as discussed in section 4.1.4. Conversely, one might address issues directly relating to the product like the operational details discussed sections 4.1.2 or 4.1.3 and those impacting a corporation's overall strategic position (section 4.1.1). It is even possible to classify risks by the cost of failure, from a company's marketing position being damaged all the way to the civil and criminal penalties resulting from violating export restrictions (section 4.1.5).

The fact is, there is no simple way to classify the myriad of risks that present themselves. For our purposes, though, we shall present seven categories: four based on the nature of the information placed at risk and three reflecting other aspects of risk. These categories are:

- Risk categories based on information type:
    1. Strategic Corporate Information
    2. Internal Product Design
    3. Product Operating Environment
    4. Third Party Data
- Risk categories based on other aspects:
    5. Export Restrictions
    6. Breach of Security
    7. Internal Threats

These categories are graphically represented in Figure 4, with the variations in shading showing the categories based on information type.

Figure 4 – Categories of Risk

Naturally, this classification scheme is far from definitive. It is entirely possible that a specific incident might fall into more than one category or that new categories might emerge as time marches on. Nonetheless, this list provides us with a good framework with which to discuss the topic.

4.1 Risk Categories in Detail

4.1.1 Strategic Corporate Information

Strategic information in the form of marketing or financial plans can be amongst the most sensitive data that a company can generate. While the compromise of other categories of data we shall address might be catastrophic to a particular project or product line, the risk is much higher for strategic information where the entire future of a company might be placed in jeopardy.

Traditionally the protection of this information is reasonably straightforward, being restricted to an inner circle of managers and marketing personnel within a company. In outsourcing a software development effort, however, details of the product being commissioned and its place with the product line are inevitably exposed. For this reason, the RFIs or RFPs generated as part of the bid process are discussed in section 2.1.1 must include provisions for protecting such data.

4.1.2 Internal Product Design

Organizations put enormous amounts of time and effort into engineering new products. As a result, the details of these designs can be quite valuable. In a military system, this might include information about operating frequencies, sensitivities of sensors, or the range and accuracy of systems. In a civilian system, such as an online shopping or banking system, this would include details about the specific cryptographic techniques used to protect customer data. Proprietary device interfaces may also fall into this category.

If this information is exposed, a competitor may take any number of actions to the detriment of the data's rightful owner. In the case study presented in section 3.1, the Chinese military used information provided by American companies for civilian endeavors to improve its own ballistic capabilities. While this was in violation of United States export law (see section 4.1.5) it is also a classic case of internal product design compromise.

4.1.3 Product Operating Environment

While it might be comforting to think that a product's secrets are contained within its interior, the fact is that modern systems are often greater than the sum of their parts. The simple fact that a company goes to the trouble to develop a device or a piece of software implies plans for how that product will be used. For example, if an e-commerce site's published XML interface suddenly included a provision to present prices in a new currency, it might signal to competitors that the company is entering new markets. This piece of intelligence can be gleaned despite the fact that the implementation of the feature (multiplying a price by a conversion factor) is hidden and of little importance in and of itself.

In this example, it can be seen that information about a product's operating environment can reveal as much or more than details about its internal design. For military applications, information about operating environments that need to be protected would include an aircraft's operating altitudes, a submarine's depths, or even ocean salinity or mold and spore ruggedness requirements. For civilian systems, specifications about interfaces to specific products (implying business partnerships with the other company), regional operations (e.g. currency units, power

sources, time zones), or other operational characteristics such as portability or ruggedization (implying target markets) should be treated as sensitive.

4.1.4 Third Party Data

In addition to protecting its own data, an organization has a responsibility to protect data entrusted to it by third parties. This data generally takes two forms: customer data and technology data provided for a development under a partnership. An example of compromised customer data was presented in 3.2, with families suddenly finding sensitive information being posted on the Internet.

Technology partnership data can best be demonstrated with an example. Let us imagine that company A signs a nondisclosure agreements with company B in order to use its technology (e.g. parts, interfaces, etc) in its products. When this occurs, company A has an obligation to protect any proprietary data provided by company B. If company A then hires an outsource vendor, it must ensure that that this vendor will also protect any of company B's data with the same level of diligence.

Depending on the structure of the nondisclosure agreement, there may also be contractual obligations to inform company B about such an outsource relationship. Any potential conflict of interest must be identified at the outset. In our example, such a scenario might be if company A's outsource vendor also does work for one of B's competitors.

Liabilities for exposure defined by the nondisclosure agreement need to be examined by an organization's counsel and the business risks need to be folded into any assessment made when the decision to outsource (section 2.1.1) is made.

4.1.5 Export Restrictions

In the interests of security the United States and many other nations impose restrictions on the export of specific technologies developed within their borders. The goal is to present potential enemies from developing capabilities that can threaten the country's interests and also to comply with international treaties. While this includes explicitly military products (i.e. conventional small arms or artillery, munitions, chemical, nuclear, or biological weapons), these restrictions also are in place on a number of civilian, "dual use" technologies, such as encryption[23] or high performance computers[24].

The licensing and review requirements that allow export of products that might touch on these issues can be quite complex, varying not only by the technology involved but also by the country to which one intends to export. In addition, exporters must check against a "Denied Person List"[25] to see that they are not selling items to individuals and companies that are known to violate export laws.

The issues become even muddier with technologies that do not fall clearly within the established boundaries set for under the laws. For example, the United States requires a license to export polygraph equipment "except biomedical recorders designed for use in medical facilities for monitoring biological and neurophysical responses[26]." Let us imagine, however that one were to develop an advanced, portable monitor that could be used by paramedics in the field when they

[23] See http://www.bis.doc.gov/encryption/default.htm.

[24] See http://www.bis.doc.gov/hpcs/default.htm.

[25] Found at http://www.bis.doc.gov/dpl/thedeniallist.asp for the United States.

[26] "Export Administration Regulations, April 24, 2006, Commerce Control List Supplement No. 1 to Part 774 Category 3," p 38, http://www.access.gpo.gov/bis/ear/pdf/ccl3.pdf.

respond to medical emergencies. Such devices, however, could also be used in police interrogations in nations that persecute political dissidents. In such cases, it is prudent to retain experts in the field of export compliance to ensure that no laws are being broken.

The cost of violating such laws can be very costly. In addition to administrative fines or denial of the right to export, here is a partial list of criminal penalties enforced by the United States[27]:

"Willful violations:"

- Corporation - A fine of up to the greater of $1,000,000 or five times the value of the exports for each violation;
- Individual - A fine of up to $250,000 or imprisonment for up to ten years, or both, for each violation.

"Knowing violations:"

- Corporation - A fine of up to the greater of $50,000 or five times the value of the exports for each violation;
- Individual - A fine of up to the greater of $50,000 or five times the value of the exports or imprisonment for up to five years, or both, for each violation.

As can be seen, these are serious matters. In light of this, it is perhaps fortunate that the companies in our section 3.1 case study faced only fines and no one was imprisoned.

4.1.6 Breach of Security

This category describes the threat of sabotage, intrusion, or criminal action as a result of actions by employees of an outsource vendor. As direct examples, we can look to cases of fired employees committing sabotage or offshore call center employees stealing credit card information[28]. In these cases, information or physical security is directly compromised by the actions of contracted employees breaking trust.

Such threats may also be facilitated by the compromise of internal design information as discussed in section 4.1.2. As an example, a disgruntled contractor developing a web portal might exploit the site's weaknesses to orchestrate a denial of service attack.

Beyond the direct threats discussed above, this category also includes indirect threats from third parties exploiting holes in an outsource vendor's security or taking advantage of negligence by vendor employees. An example of the former would be intrusion into a network via a vendor's insecure network or viruses spreading through a shared e-mail server. An example of the later would be the case study discussed in section 3.2, where carelessness by a subcontractor resulted in private information being placed on the Internet.

4.1.7 Internal Threats

As a final category, we turn our eyes back to the outsource client and look at internal threats. An example of this might be a project manager using the same passwords given to an overseas call center in order to steal credit card information. In that case, the outsource relationship is used as "cover" to reduce the chance that the internal employee is caught.

---

[27] See http://www.bis.doc.gov/enforcement/eeprogrm.htm#Penalties.

[28] In 2005, an Indian call centre worker sold the bank account details of 1,000 UK customers to an undercover reporter. See "Indian call centre 'fraud' probe," BBC News, June 23, 2005, http://news.bbc.co.uk/1/hi/uk/4121934.stm.

Another internal threat that should be considered is the possibility of employees angry at an outsourcing decision decide to commit sabotage or theft as a way of "getting even" with their employer. Given the continued controversy and anger over outsourcing as exemplified by commentaries in professional journals and elsewhere[29], this is an area of growing concern.

4.2 Risks Mapped to the GSDOP

If we analyze the risks discussed in section 4.1, it is possible to make some basic assumptions about where in the outsource process they will be most prevalent. While these are merely educated guesses, this information becomes very useful when determining when various strategies for safeguarding proprietary data should be implemented. This is precisely what we shall do in section 5.

[29] As an example, see Lauren Weinstein, "Outsourced and Out of Control," *Inside Risks 164, Communications of the ACM*, vol 47, no 2 (2004).

Likelihood of Occurrence: L – Low M – Moderate H – High ***GSDOP Phases***	***Risk Categories***						
	Strategic Corporate Information	Internal Product Design	Product Operating Environment	Third Party Data	Export Restrictions	Breach of Security	Internal Threats
Phase 0: Decision to Outsource	M	L	L	L	L	L	L
Phase 1: Planning and Analysis	H	L	M	L	L	L	L
Phase 2: Design	M	M	H	M	L	L	M
Phase 3: Implementation and Operation	L	H	M	H	M	M	M
Phase 4: Termination	L	M	L	M	M	H	M

Table 2 – Risks Mapped to the GSDOP

We shall begin this analysis by looking at the risk categories based on the type of proprietary information. In doing so, we shall assume that the risk of compromise of a particular type of information is roughly proportional to its common usage on the project. As such, we can see in Table 2 that Strategic Corporate Information is at highest risk early in the project when the existence and nature of the project is first being shared with outsource vendors, peaking in the Planning and Analysis phase of the GSDOP. As the project advances, however, the sensitive nature of this information diminishes as the client prepares to launch the product and begins to actively advertise it to the target market.

The risk to information about a product's operating environment, on the other hand, peaks slightly later that strategic information. This reflects the fact that such information would be reflected in the product's system requirements and be exposed while those requirements are being developed. Finally, Internal Product Design and Third Party Data are used primarily during the actual implementation of the project, reflected in the table with a peak during the Implementation and Operation phase.

The remaining three risk categories may be mapped into the table by making some basic assumptions about how a project is implemented under the GSDOP.

- Export Restrictions – As discussed in section 4.1.5, Export Restrictions tend to take the form of prohibitions against specific types of technology transfers. Given that, we can assume that the sensitive information covered by these regulations (i.e. how to implement these technologies) does not become necessary until the implementation of a project. Furthermore, we can assume that once a technology is placed in a product, it is "contaminated" and its risk of exposure will remain constant for the rest of the product's life. Given this chain of logic, we can assume a low level of risk at the beginning of the GSDOP that then increases during implementation and remains constant.

- Breach of Security – For this category, we shall assume that the risk first increases when the most people are assigned to the project (i.e. Phase 3, Implementation). When the Termination phase occurs, though, the risk will actually increase. It becomes vital to close out and deactivate every network account, to account for every piece of media containing proprietary data, to collect every key and security pass issued. If even one of these assets "falls through the cracks," it falls completely outside the control of either the client or the vendor. In such circumstances, handing proprietary data over to unauthorized parties becomes much more attractive since the possibility of being held accountable diminishes to insignificance.
- Internal Threats – This is perhaps the most difficult category of threat to characterize, being a catch all for malevolent intent by internal employees. For this reason, we shall only assume that the risk becomes significant when the largest number of employees become involved in the project (again, during project implementation). We shall assume that the risk remains constant even as staff levels drop, with diminished management oversight of the project encouraging any dishonest employees to act.

As we can see, this analysis is entirely speculative, built upon a series of assumptions and logical chains. Nonetheless, it allows us to understand what sort of information is at risk at what stage.

5 Strategies for Protecting Sensitive Data

With our discussion of risks completed, we can now address the topic of strategies that can be implemented to mitigate those risks. As with the risks, there are many possible ways to categorize the different strategies available. We shall examine three basic categories:

- *Organization Centered Strategies* – Strategies centering on the outsource provider's organization.
- *Project Centered Strategies* – Strategies dealing with the way in which a particular software development project is implemented and managed.
- *COTS Centered Strategies* – The use of commercial off the shelf (COTS) and Open Source (OS) products instead of outsourcing as a way to mitigate risk.

Below, we shall discuss these strategies and which risk categories they are most suited in mitigating.

5.1 Organization Centered Strategies

As discussed in section 1, one of the basic concerns over the security of proprietary data is the lack of institutional allegiance to the outsource client. Simply put, the workers at an outsource vendor have no personal stake in the success or failure of the client or in the security of the client's data. The most direct way to address this concern is to develop strategies centered on the outsourcing vendor's organization.

Table 3 introduces two such strategies: the vetting of workers employed by the outsource vendor and the use of substantial penalties in contracts should data entrusted to the vendor be compromised. Each strategy is evaluated with regard to its ability to mitigate the various risk categories presented in section 3.1.

Ratings: E – Effective L – Less effective N – Not effective I – Inappropriate Strategy R – Required by law or contract ***Strategy***	***Risk Categories***						
	Strategic Corporate Information	Internal Product Design	Product Operating Environment	Third Party Data	Export Restrictions	Breach of Security	Internal Threats
Vetting of Workers	E	L	L	L	R	E	I
Contract Penalties	E	L	L	E	R	L	I

Table 3 – Suitability of Organization Centered Strategies to Risk Categories

5.1.1 Vetting of Workers

In this strategy, an outsource vendor performs background checks or other research on its employees in an attempt to weed out those that might be prone to compromising the security of a client's project. While this might be time consuming and expensive, it can nevertheless be an effective tactic if the screening process is thorough. The strategy can be attractive to vendors as it can serve to reduce dishonesty in general by excluding employees that might participate in timecard fraud, theft, abuse of equipment, etc. Vetted employees can also be used as a selling point when a vendor wishes to solicit business from security conscious clients.

In Table 3, we consider this strategy most effective in countering the risks of security breaches. The reason is that such a threat implies either active malevolent intent on behalf of an employee (for sabotage or theft) or negligence in observing security procedures (resulting in mishandled data, network penetration by a third party, or the spread of computer viruses), both of which can be screened for. Strategic information may also be better protected by vetted employees, as they should be less vulnerable to being compromised by industrial espionage attempts.

For protecting data during project implementation (Internal Product Design, Operating Environment, and Third Party Data), however, this strategy may prove to be slightly less effective. The logic here is expressed in the maxim "He who would defend everything defends nothing[30]." Given the sheer volume of information needed to implement a modern software project, it is foolish to assume that even the most diligent of workers can protect every last scrap of data from accidental disclosure while still being productive. For this reason, schemes to track or isolate proprietary data, such as presented in section 5.2, are necessary to allow workers to protect items worthy of protecting. When this occurs, vetted employees allow the tracking and isolation strategies to be even more successful.

[30] Carl von Clausewitz, *Vom Kriege, [On War]*. 1874.

With these categories evaluated, we are left with two more areas of risk to discuss. By law, technologies subject to Export Restrictions can not be allowed to leave the country of origin. Thus, employees using the technology must by definition meet specific citizenship and residence criteria, a form of defacto vetting. Further, by definition, the vetting of employees at an outsource vendor is ineffective in preventing dishonest actions by a clients employees ("Internal Threats").

5.1.2 Penalties for Violations

In this strategy, a client structures the contract with its vendor during the RFP and RFI stages to impose financial penalties for the compromise of sensitive information. In order to write such a clause, though, one must identify and define what data is and isn't proprietary. This in turn implies a tracking system such as described in section 5.2.1. Other costs might be elevated as well if the penalties discourage some vendors from bidding and thus reducing competition for the bid. In addition, a vendor might choose to purchase insurance against the event of the clause being activated and pass the costs back to client.

In analyzing this strategy, we can see that the intent is to provide an incentive for an outsource vendor to protect the client's proprietary data. While this might encourage the vendor firm's managers to provide more oversight on this issue, the fact is that this strategy revolves around enforcing a sense of honesty that one would already expect from an outsource vendor. In this way, it is very similar to the vetting of employees as presented in section 5.1.1. That strategy attempts to reinforce personal integrity while this one focuses on organizational integrity.

For this reason, Table 3 presents a similar analysis for the two strategies. The exception is in the area of Third Party Data, where we give Contract Penalties a slight edge in effectiveness. This is based on the assumption that the data is provided to with its own contractual clause against disclosure, a clause that should be equally effective for the vendor as the client.

5.2 Project Centered Strategies

The strategies below are prefaced on a basic security principle known as the *Rule of Separation*. Kevin Day presents the concept and the consequences of failing to heed it in this manner:

> The Rule of Separation states that to secure something, it must be separated from the dangers and threats of the world around it... By not practicing the Rule of Separation, an organization multiples its exposures [with each new user] and, at the same time, reduces the overall level of security...[31]

While Day is primarily discussing access to electronic information systems, the same principle stands for access to any form of sensitive data. Thus, we should enforce the Rule for all forms of data access. We must not only take care with system passwords and account privileges, but also with access to hard copy records, backup media, or physical access to project workspaces and laboratories.

Table 4 introduces the three strategies that build upon the Rule of Separation and maps which strategies are most suitable for the seven risk categories presented in section 4.1. We can summarize these three strategies with the acronym TPS: Track, Partition, and Sanitize.

---

[31] Day, *Inside the Security Mind: Making the Tough Decisions*, 58.

Ratings: E – Effective L – Less effective N – Not effective I – Inappropriate Strategy R – Required by law or contract ***Strategy***	***Risk Categories***						
	Strategic Corporate Information	Internal Product Design	Product Operating Environment	Third Party Data	Export Restrictions	Breach of Security	Internal Threats
Partition Types of Data	L	E	L	L	E	I	I
Track and Control Proprietary Data	E	E	E	E	I	I	E
Sanitize Project Requirements	L	L	E	L	I	I	I

Table 4 – Suitability of Project Centered Strategies to Risk Categories

5.2.1 Partition Types of Data

One of the underlying principles of the Rule of Separation is that one can categorize data according to its sensitivity and use. For example, one might keep payroll information on one server and engineering design data on another server. Further, one might mandate that highly sensitive information such as long term strategic marketing plans be kept isolated from less sensitive data like daily status reports. In the first instance, we are partitioning data *by use* while in the second we are partitioning data *by type*.

The use of two-dimensional partitioning can be seen in the security apparatus of various governments. For example, the United Sates classifies the sensitivity of information as Confidential, Secret, or Top Secret[32]. It also imposes additional designations based on the type of information involved such as cryptographic information, satellite intelligence, or nuclear weapons data. When used in conjunction with a security clearance system, access to sensitive information can be controlled.

Figure 5 demonstrates how a private company might impose two dimensional partitioning via its information network. We can see that the company has set up a confidential LAN for use by management and servers for each department. By placing specific types of data on the correct server or LAN, we partition them by type (so that marketing personnel do not have access to engineering documents). We can also control the privileges for accounts to each server based on the sensitivity of the data therein (since a junior manufacturing engineer preparing daily productivity reports does not need access to supply chain details). The use of a separate management LAN adds an extra layer of isolation for the company's most sensitive documents.

[32] Wikipedia contributors, "Security clearance."

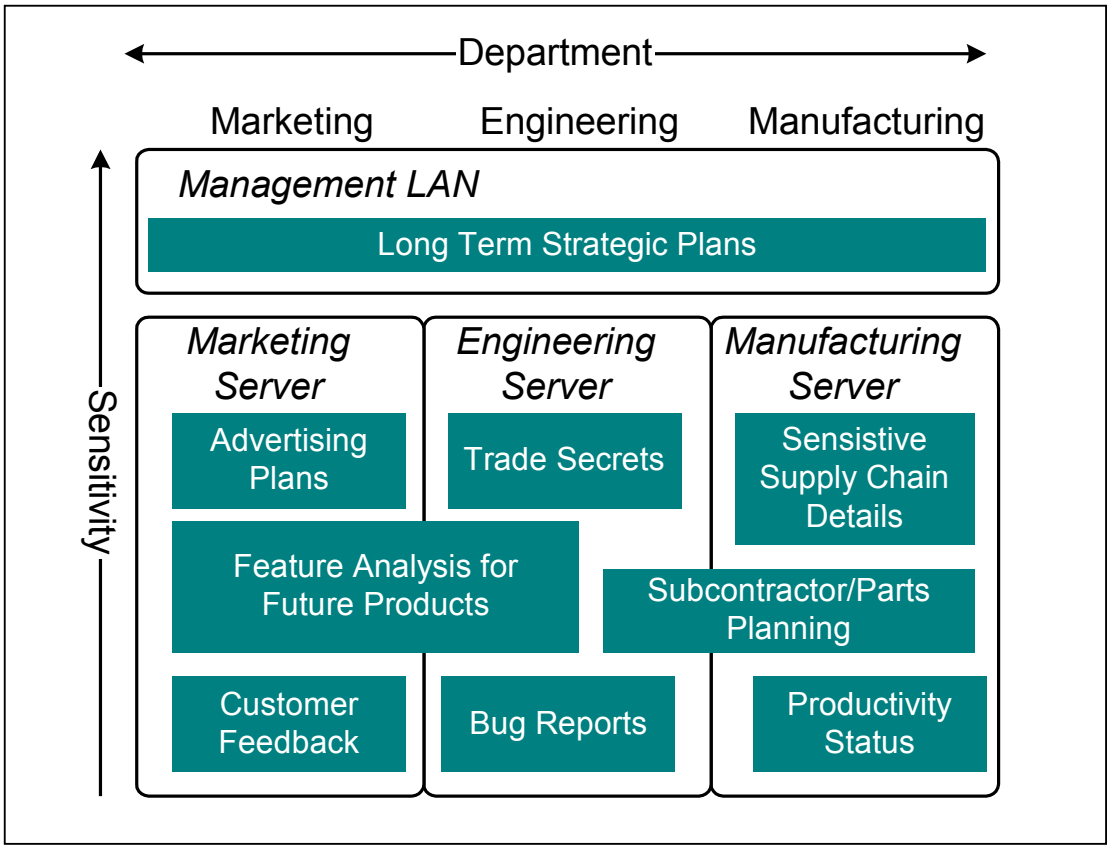

Figure 5 – Example Data Partitioning in Two Dimensions

Moving on to risk categories, we can refer back to Table 4 and see that partitioning data in a software development project is most effective in protecting the internal design of a product and in ensuring that no export violations occur. For the Internal Product Design category, the isolation of work products, information systems, and work areas that employees use daily helps to enhance overall security. This isolation also allows us to ensure that no sensitive data covered by export laws wind up being transferred.

We can also see that this strategy is less effective in protecting data that is shared with stakeholders in multiple companies, reflecting the difficulty of imposing a data partitioning plan over multiple organizations. While offering some benefits for Strategic Corporate Information, Product Operating Environment, and Third Party Data, additional strategies must be used to ensure proper protection.

5.2.2 Track and Control Proprietary Data

Expanding upon the data partitioning strategies discussed in section 5.2.1, we can introduce additional measures to protect proprietary data. We can begin by addressing the problem of *data contamination*.

The strategy of Data Partitioning is prefaced on the assumption that once data is isolated it will remain that way. However, this is not always the case. If a document is misfiled or a computer file is accidentally copied to the wrong server, it has broken the partitioning we have established. If the original user of the information is diligent in regularly examining their work area and user accounts, it is theoretically possible that they can catch these errors and restore the data back to

its correct area. The difficulty, however, lies in other workers that do know the context of the data and may not understand that it is sensitive. For example, it is unreasonable to expect a junior engineer to realize from brief inspection that marketing data regarding competitor A is common knowledge but that data regarding competitor B is the result of a confidential analysis that took weeks to develop. In such an instance, the engineer might propagate this sensitive data in his own documents without taking the proper safeguards to protect it.

To address this, we might choose to begin tracking sensitive data. The most basic technique for this is to establish marking guidelines for documents. In that way, every document in an organization is clearly identified at a glance by the categories established as part of the data partitioning scheme (e.g. "Internal Use Only," "Confidential," etc). In addition to identifying instances of data contamination, markings such as these also allow the implementation of policies for the reproduction and destruction of data (i.e. the use of a shredder instead of the trash, etc).

Once a Marking Plan is in place, extremely sensitive data might be further controlled by imposing a formal Data Accountability Process. Such a policy would require individual copies of a document to be identified by serial number and tracked by individuals. It is also possible to use technological innovations such as embedding hidden serial numbers in electronic documents or images to track data even if visible markings are removed.

The greatest asset of a Data Accountability Process is that it ensures accountability for the sensitive data workers have in their custody. For this reason, we can consider it an effective technique in countering risks based on all types of data (Table 4). Further, this accountability lessens the possibility that Internal Threats might use outsourcing as an excuse to cover their own actions (see section 4.1.7).

5.2.3 Sanitize Project Requirements

Another strategy that builds upon Data Partitioning is the structuring of requirements such that it is impossible to determine specific sensitive data. This process is called *requirement sterilization* and is demonstrated in the example below.

Let us imagine that we are constructing a military aircraft and wish to outsource software development for the fuel management system. However, the maximum altitude, speed, and range of the craft are classified. In addition, the weight and fuel consumption rates of the aircraft are classified because someone might be able to derive information such as weapon load or types of engines from this information. If we provide this data as part RFP, our subcontractors must use only employees with the proper government clearances and obey the rules for the handling and storage of classified data. Not only will this impose substantial overhead costs but it will also restrict the number of available bidders and lessen competition.

Imagine, though, that we structure our requirements such that they encompass a wide range of parameters. We specify that the system must handle inputs for the weight as a double precision floating point number in the range of 0 grams to 1 petragram ($1x10^{12}$ grams). Since it is no secret that the weight of an aircraft will be somewhere in that range[33], we need not consider that interface sensitive. Building the requirements for the other sensitive parameters around similar broad ranges eliminates the need of the vendor to possess any of this sensitive data.

This strategy is best suited for protecting specific ranges of parameters in systems with well-defined interfaces. It serves of little use when a vendor must know internal details of a system or of third party technologies embedded within. This is reflected in the entries in Table 4. It is also

[33] For reference, the battleship New Jersey displaces a little under 250000 Kg, or one quarter of a petragram.

of little help when protecting Strategic Corporate Information and is completely ineffective in countering any of the other threats we have discussed.

5.3 COTS Strategies

The use of Commercial, Off the Shelf (COTS) products as an alternative to outsourcing is an attractive one. Here, instead of commissioning a component of a system from scratch, an existing, marketed product is used in its place. Naturally, the limiting factor is the existence of a suitable product to be used in this way. While there are many options in some fields (operating systems, graphics packages, and others), it is quite likely that there exist no prepackaged products that can fit the bill. Nonetheless, if there exists a COTS alternative that can be supported by sound engineering and cost rationale, an organization should strongly consider it as an option.

Table 5 analyzes of these options, differentiating between COTS and Open Source products.

Ratings:

E – Effective

L – Less effective

N – Not effective

I – Inappropriate Strategy

R – Required by law or contract

Strategy	***Risk Categories***						
	Strategic Corporate Information	Internal Product Design	Product Operating Environment	Third Party Data	Export Restrictions	Breach of Security	Internal Threats
Use COTS Products	E	E	E	I	L	N	E
Use Open Source Products	E	E	E	I	I	N	E

Table 5 – Suitability of COTS Strategies to Risk Categories

5.3.1 Use of COTS Products

As mentioned before, the use of COTS products can be very attractive for purposes of protecting proprietary information. The reason for this is that there is essentially no interaction between the developers of the COTS product and the client implementing it as part of their own system. The component is developed and marketed independently, built for use by any number of clients. As such, there is no risk that the COTS developers take custody of proprietary data from the client and no risk of compromising it. This one way flow of information, while restricting engineering innovation and delaying corrective action on defects in the COTS product, inherently enforces the Rule of Separation presented in section 5.2.

The analysis presented in Table 5 reflects this, enforcing isolation on matters of Strategic Corporate Information, Internal Product Design, and Product Operating Environment. By eliminating any actual outsourcing efforts, it also denies the clouding of responsibility that is leveraged by Internal Threats. This strategy can also be effective at safeguarding technologies protected by Export Restrictions, but only if those technologies are included as features in the COTS product. It is entirely possible, though; that the COTS vendor might simply choose to

leave such features out of their product to reduce its own responsibility under those laws, forcing the client to develop those features themselves.

On the other hand, when one purchases a COTS product one inherits the defects of that product. While this is of crucial interest in making the engineering decision to use or not use a product, it also has impact in our analysis of risk categories. Specifically, if the COTS product used has security weaknesses, these will be propagated into the client's system. Even worse, since the COTS product will likely be in much wider use than the system that the client is building, these weaknesses will probably be well known. For this reason, the analysis in Table 5 gives this strategy low marks in the Security Breach category.

5.3.2 Use of Open Source Products

In the context of this paper, Open Source products are those developed under licensing schemes that do not restrict the distribution or adaptation of the product or require royalties for their use. As such, they can be freely developed and distributed. Currently, the widely known Open Source product is the Linux operating system, although there exists many others that are suitable for inclusion within a larger software system.

The decision to use Open Source components instead of proprietary COTS products is one that requires sound engineering and business judgement. Offerings can vary greatly in quality, feature sets, and the level of support. For our purposes, though, the two choices can be considered to be nearly identical in terms of risk categories. This is because both choices imply the same one way relationship between client and the developer of the COTS or Open Source Component. The only exception between the two shown in Table 5 is on the matter of technologies subject to Export Restrictions, which Open Source products (being freely distributed) do not support.

6 Evaluation of Strategies

With each individual strategy introduced and analyzed, we can begin to examine the costs and effectiveness of each strategy. We can also look at how these strategies can be combined in order to ensure broad protection for proprietary data when we decide to outsource a software development project.

6.1 Effectiveness of Strategies

In examining the analysis of strategies presented in Table 3, Table 4, and Table 5, we can see that some strategies are more effective in protecting against specific risk categories than others. In Table 6, we revisit these results and note the effectiveness of the various strategies, highlighting some patterns of interest.

Ratings:

E/L/N – Most to least effective

I – Inappropriate Strategy

R – Required by law or contract

Color Key:

General purpose strategies

Engineering case specific

Less effective strategies

Strategy	***Risk Categories***						
	Strategic Corporate Information	Internal Product Design	Product Operating Environment	Third Party Data	Export Restrictions	Breach of Security	Internal Threats
Vetting of Workers	E	L	L	L	R	E	I
Contract Penalties	E	L	L	E	R	L	I
Partition Types of Data	L	E	L	L	E	I	I
Track and Control Proprietary Data	E	E	E	E	I	I	E
Sanitize Project Requirements	L	L	E	L	I	I	I
Use COTS Products	E	E	E	I	L	N	E
Use Open Source Products	E	E	E	I	I	N	E

Table 6 – Annotated Risks and Strategies

By marking the "Effective" entries, we can first see that the tracking and control of proprietary data as presented in section 5.2.2 is the most effective, earning high marks in five of the seven risk categories. Further, because it is built upon the strategy of Data Partitioning (section 5.2.1), we see that when the two are implemented together we get additional coverage of an additional risk category (Export Restrictions) and supplemental coverage of Internal Product Design. This leaves only the Breach of Security category open.

If we choose to add the strategy of worker vetting (section 5.1.1) we can fill this gap and incidentally make the earlier two strategies that much more effective, since honest workers of high integrity are most likely to follow the procedures requested of them. In this way we have discovered a suite of three strategies (Vetting of Workers, Partition Types of Data, and Track and Control Proprietary Data) that result in complete coverage of all the risk categories presented.

In examining the remaining strategies, we see that three (Sanitize Project Requirements, Use COTS Products, and Use Open Source Products) offer coverage in various categories. However, from our discussion of the topics we can recall that these three are of use in only limited circumstances since they are dependent on engineering case specific conditions. Specifically, requirements can't be sanitized if a high degree of coupling is required between vendor and client portions of a design and the COTS and Open Source strategies are dependent on the availability of suitable components. We can thus consider these strategies, regardless of their specific merits, of use only in limited circumstances.

Finally, we can examine the last remaining strategy: Contract Penalties. As can be seen from the analysis, this technique results in limited benefits. Adding to this is the discussion in section 5.1.2 of how it increases costs and limits competition. For these reasons, we can see that this is the least effective approach to take amongst all the strategies presented.

6.2 Infrastructure & Cost

We shall now examine the required infrastructure and associated costs to implement each strategy. We shall assess each in the following terms:

- Imposition on Outsource Vendor – How much the strategy imposes on the vendor's normal operations (assuming the vendor does not already use this strategy internally already)?
- Initial Cost – The cost to initiate the strategy before the Implementation Phase of the GSDOP (i.e. in Phases 0-2).
- Ongoing Cost – The cost of the strategy while the project is being implemented (i.e. in GSDOP phases 3 & 4).

Table 7 shows estimated ratings for each of these categories. While conjectural, they are nevertheless informative (see section 8 for discussions about further research on this and other categories).

Ratings: H – High M – Moderate L – Low N – Negligible ***Strategy***	Imposition on Vendors	Initial Cost	Ongoing Costs	Comments
Vetting of Workers	H	M	M	Assumes expenses of attracting and retaining vetted workers
Contract Penalties	L	H	N	Assumes minimal costs once contract is in place
Partition Types of Data	H	L	L	Assumes up front cost of building isolated networks, but that maintenance of same is low
Track and Control Proprietary Data	H	L	M	Assumes ongoing effort to train workers and track data
Sanitize Project Requirements	N	M	N	Assumes up front cost of "sanitizing" requirements
Use COTS Products	N	M	M	Ongoing cost assumes royalties to COTS vendor
Use Open Source Products	N	M	M	Ongoing cost assumes dedicated personnel to maintain open source components

Table 7 – Costs of Strategies

As we look at the table, we can see some patterns. First, the most effective strategies discussed in Table 6 (Vetting of Workers, Partition Types of Data, and Track and Control Proprietary Data) are the ones that impose upon an outsource vendor the most. This indicates that one must have a high degree of interaction between client and vendor if one wishes to protect proprietary data. Without the client dictating the terms of safeguards for his or her sensitive data from the outset, the risks become unmanageable.

Second, we can see that nearly all the strategies (with the exceptions of Contract Penalties and Sanitize Project Requirements) have an ongoing cost rating that is identical to or greater than the initial cost rating. This indicates that safeguarding sensitive data is an ongoing investment and can not be forgotten once a project begins implementation. Fortunately, only one strategy (Track and Control Proprietary Data) shows an increase in costs. This indicates that these costs should be controllable with good management oversight.

7 Conclusions

So, what lessons have we learned over the course of this paper? While there are many questions left unanswered, we can make the following generalizations.

First, we have seen that software outsourcing follows a process just as software development does. The dominant model for this process, which we have named the GSDOP, has strong similarities to Royce's Waterfall model for software development. We have also speculated that as the practice of outsourcing software development projects matures other alternative models may emerge.

Second, we have seen from our case studies and discussions of the natures of threats that it is possible to categorize the risks to sensitive data. Most of these categories are based upon the type of data involved, but others address more traditional threats of behavior from employees of both vendor and client.

Third, we have learned that there are numerous strategies for addressing these threats. While some are centered on the vendor organization, others must be implemented on a project by project basis, and others look to COTS and open source solutions as alternatives to outsourcing entirely. In examining these strategies in detail, we have learned that some are more effective than others at handling certain threats.

In our final analysis, we determined that some otherwise attractive strategies are limited in that they can only be applied when the engineering case justifies their use. Taking this into account, we designated a core set of strategies (Vetting of Workers, Partition Types of Data, and Track and Control Proprietary Data) as providing the most effective coverage for all the risk categories. We also determined that these strategies tended to be the most intrusive on the vendor and required ongoing investment and attention by the client.

Protecting sensitive data is an expensive, time consuming proposition, requiring planning and strategy well before the decision to outsource is even made. An organization's leadership must balance the risks and benefits in order to determine what course to take. Even then, they must remain diligent and attentive to new threats as the project progresses through its implementation and eventually ends its life cycle. While working as a team with their vendors, they must never loose sight of the inherent risks that outsourcing presents. Like Machiavelli with his mercenaries, they must always remain aware of the price of failure.

8 Directions for Further Research

As the practice of outsourcing software development increases, the study of the risks this trend presents to proprietary data must grow with it. While the intent of this paper is to serve as an introduction to the issue, much more work needs to be done. Below are areas that deserve further scholarly scrutiny.

- A case study of an implementation of a software development effort that utilizes the GSDOP introduced in section 2.
- An analysis of alternate outsourcing models expanding upon or serving as alternatives to the GSDOP, as briefly touched upon in section 2.3.
- A study of risks actually encountered during an outsource effort as they fall into the categories discussed in section 3.1. Such a study should also address the estimated cost of failure for each risk and rank which risks might be the most expensive to an organization.
- An analysis of actual proprietary data from a project to see what categories of information are present in the various stages of the GSDOP to prove or disprove the assumptions made in section 4.2.
- The gathering of empirical data to prove or disprove the costs of implementing the strategies shown in 6.2.

9 Bibliography

Axelrod, C. Warren. *Outsourcing Information Security*. Boston: Artec House, 2004.

Bacon, Francis. *Religious Meditations, Of Heresies*. Trans of *Meditationes Sacræ. De Hæresibus*. 1597

Boehm, Barry. "A Spiral Model of Software Development and Enhancement." *ACM SIGSOFT Software Engineering Notes*, vol 11, no 4 (1986): 22-42.

CMMI Product Team, *Capability Maturity Model® Integration (CMMISM), Version 1.1*. Pittsburgh: Carnegie Mellon Software Institute, 2002.

Day, Kevin. *Inside the Security Mind: Making the Tough Decisions*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, 2003.

Gerth, Jeff. "2 Companies Pay Penalties For Improving China Rockets." *New York Times*, March 6, 2003.

Kan, Shirley A. "China: Possible Missile Technology Transfers from U.S. Satellite Export Policy – Actions and Chronology. Updated September 5, 2001." http://www.fas.org/spp/starwars/crs/98-485.pdf

Los Angeles Times Editorial Board. "Flea-market secrets." *Los Angeles Times*, April 14.

Machiavelli, Niccolò. *The Prince*. Trans. Marriott, W. K. (William K.). Project Gutenberg, February 11, 2006, http://www.gutenberg.org. Trans. of *Il Principe*. 1513

Marks, Gene. *The Complete Idiot's Guide to Successful Outsourcing*. New York: Penguin, 2005.

Marquis, Christopher. "Satellite Maker Fined $20 Million in China Trade Secrets Case." *New York Times*, January 10, 2002.

Reinke, Jennifer L. "New York's Information Security Breach and Notification Act." *NYSBA/MLRC Municipal Lawyer*, vol. 20, no. 1 (2006): 19.

Royce, W.W. "Managing the Development of Large Software Systems: Concepts and Techniques." *Proceedings, WESCON*, August 1970: 328-338.

Sullivan, Bob. "Government agency exposes day-care data: Daily whereabouts of hundreds of children posted on public Web site." MSNBC, Feb. 8, 2004. http://www.msnbc.msn.com/id/4186130/.

United States Department of Commerce. "Bureau of Industry and Security Website." http://www.bis.doc.gov/encryption/default.htm

Clausewitz, Carl von. *On War*. Trans. Graham, J. J., Colonel. Project Gutenberg, February 26, 2006, http://www.gutenberg.org. Trans. of *Vom Kriege*. 1874.

Watson, Paul. "Leaks of Military Files Resume." *Los Angeles Times*, April 25, 2006.

Weinstein, Lauren. "Outsourced and Out of Control." *Inside Risks 164, Communications of the ACM*, vol 47, no 2 (2004).

Wikipedia contributors, "Security clearance," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Security_clearance&oldid=50053968 (accessed April 29, 2006).

"Indian call centre 'fraud' probe," BBC News, June 23, 2005, http://news.bbc.co.uk/1/hi/uk/4121934.stm.

"U.S. Computer Files Remain on Market in Afghanistan." *Los Angeles Times*, April 15.

"Export Administration Regulations, April 24, 2006, Commerce Control List Supplement No. 1 to Part 774 Category 3." http://www.access.gpo.gov/bis/ear/pdf/ccl3.pdf.