2015 Edition

IT
Due Diligence
Guide

Jim Hoffman

TABLE OF CONTENTS

INTRODUCTION TO THE 2015 EDITION

In the context of M&A, the need to include an IT review is greater than ever. Identifying IT shortcomings that can put a company's (and a deal's) future at risk is critical. Other issues that are uncovered that may not rise to the level of cancelling a transaction can still carry a high price tag to address, and it's important to find these problems prior to the deal's closing.

According to the Boston Consulting Group[1], 63% of acquisitions are completed by companies that purchase no more than one company per year. It's not reasonable to expect that such companies have resources dedicated to M&A, let alone an IT due diligence expert. This book can help to fill that gap.

2014 saw a large number of data breaches and IT-related public relations disasters in the news. Examples included UPS, eBay, Community Health Systems, Home Depot and Sony. Smaller companies are by no means immune to the same types of attacks. The year also saw the rise of the concept of "espionage as a service" - financially-motivated hackers literally taking orders for sensitive data to be stolen from vulnerable companies.

For 2015, the *IT Due Diligence Guide* has been updated to address some of these emerging areas of concern. New questions and more detailed explanations have been added, and there is now a dedicated chapter on cybersecurity issues. In addition, the IT due diligence report template has been expanded and there are new data collection spreadsheets.

If after reading the book you feel you'd prefer to hire an IT expert to assist you, there is a new section that discusses how to identify an IT due diligence consultant.

---

[1] Jens Kengelbach, Dominic C. Klemmer, Bernhard Schwetzler, Marco O. Sperling and Alexander Roos, "Does Practice Make Perfect? How the Top Serial Acquirers Create Value," *Boston Consulting Group and HHL-Leipzig Graduate School of Management,* April, 2011. Accessed on February 23, 2015. http://www.bcg.com/documents/file74843.pdf

AUTHOR'S NOTE

I've been lucky to work in a number of companies that have been very active in M&A.

I've participated in due diligence projects on both the buy and sell side. In some cases I've lead the due diligence team, in some I've worked on only the technology aspects, and sometimes I've been the only person responsible for the entire due diligence effort. The values of the transactions have been as low as a $50,000 investment and as high as a $375 million company sale.

I've been involved in the acquisition and evaluation of technology companies for over twenty years. The due diligence process explained in this book includes the practices that I've found to uncover the most useful information during due diligence.

The book is certainly not an academic exercise. It focuses on real-world examples and experiences. My goals were to provide information that I was only able to gain through trial and error and to give the reader a head start on the IT due diligence process through the advice in the book and the additional material such as the report template and data collection spreadsheets.

This book is intended to be a valuable tool in a broader due diligence effort that should involve legal, financial and accounting experts, at a minimum. The information in this book can provide an excellent base for further discussions involving technology subject matter experts, and should not be the only resource related to technology due diligence.

I'm located in the United States, and my due diligence experience is based on US transactions. Although I believe that the majority of the information in the book is applicable in most places in the world, there may be some local differences in the due diligence process and customs.

Finally, let me state that this book is not a substitute for obtaining expert professional advice. Acquisitions and mergers are inherently complex. Both sides in a transaction should retain expert legal, financial, accounting and subject matter experts to represent their interests.

Thank you for purchasing the book.

Jim Hoffman
March, 2015

DOCUMENT CONVENTIONS

Throughout the book, I'll refer to the acquiring or investing company as *AcquiringCo* and the target company as *TargetCo.*

The majority of the book is structured as a series of topics with questions from the IT due diligence checklist that comes with the book. Each question is followed by a description of why the question is being asked and what the responses might tell *AcquiringCo* about *TargetCo,* and any appropriate next steps and follow-up questions.

Each due diligence request is marked with one or more of the following icons:

	Informational request. Many of these due diligence request responses will serve as the basis for further discussion. It would not be surprising if *TargetCo* can't supply all of the information requested, or if large company best practices are not being followed. These issues can usually be remedied after the transaction without significant expense or risk. These requests can also be used to begin the integration planning process.
	Important request. Depending upon the answer received, you may have concerns about *TargetCo's* qualifications or professionalism. Issues identified can be addressed, but there may be significant expense involved.
	Critical request. Unsatisfactory responses to these requests may be significant enough to consider abandoning the transaction or considering an allowance or adjustment in the transaction pricing or other terms.
	A question that should only be asked when you're face to face with the *TargetCo* staff. Don't include this question in your written due diligence request list.
	A data collection spreadsheet for this due diligence request is included with the book materials.

CHAPTER TWO

Due Diligence Requests

The following chapters organize the due diligence requests into logical categories matching the sections in the IT due diligence checklist that comes with the book.

It would be very unusual for every one of these questions to apply to a particular due diligence effort. I recommend that you utilize the initial information gathering process to determine the due diligence requests that are most applicable to *TargetCo*. If you overwhelm your contact with a huge list of requests upfront, you'll most likely get pushback from *TargetCo* and you'll also probably soon hear from the person in charge of the overall due diligence effort at *AcquiringCo*.

Keep the initial due diligence request list limited to the items that will help make your onsite visit more productive, and cover the other issues that you think are important in person.

The IT due diligence checklist template included with this book is numbered. It's a good idea to number the requests, as it makes it easier for you to track the responses and for your *TargetCo* contact to indicate the request to which they're responding. There is less confusion for everyone if the files provided by *TargetCo* as they respond to the due diligence requests include the item number in the file name.

Many due diligence projects utilize a virtual "data room." This is a service or website that manages all of the documents and other information gathered during due diligence and securely stores it. The checklist includes a column to indicate the location in the data room if you're using one.

Finally, some questions are sensitive or otherwise best asked in person instead of in writing, and those are noted in the discussion where applicable, along with the appropriate icon.

CHAPTER FIVE

Software Development Process

DUE DILIGENCE REQUEST

A description of the version control process and system(s) utilized.

WHY THIS IS IMPORTANT

This is an example of a sign of a professional software development organization. A version control system allows every change to a project's source code to be tracked and stored. Think of it as the vault for the software, which in many cases can be a company's most valuable asset.

Sometimes a new version of a software product or website can cause unexpected problems in the real world, even when it's been tested prior to being released. In that situation, a version control system allows the company to "roll back" the software to an earlier, stable version very quickly and easily. A version control system also protects against accidental deletion of source code on a developer's computer.

DUE DILIGENCE REQUEST

A description of the process for approving and developing new software products or features.

WHY THIS IS IMPORTANT

You want to know if there is good control over the company's technology direction. There are many potential follow-up questions. Is there plan or just reaction to problems? Does *TargetCo* have the business domain experts to develop its own ideas, or are all product features the result of customer suggestions? Is there a product manager? Are requirements documents developed? A good practice is a functional requirements document, which

explains *what* the new feature or product does, and a technical specifications document, which defines *how* it is to be developed.

Are new market opportunities carefully considered, or are resources committed without adequate research and thought? Does *TargetCo* perform competitive and market analysis before embarking upon new product efforts? You want to get a sense of the care of the decision-making at *TargetCo* before they start spending *AcquiringCo's* money.

Be particularly wary of a company that developed its main product based on the needs of one early or large customer. In this situation, the product that has been built may meet the specific needs of that one customer, but may not be something needed or desired by the market at large. If this client is the source of most new product or feature ideas, consider it a warning sign.

DUE DILIGENCE REQUEST

A description of any outsourced software development arrangements and copies of the related agreements.

WHY THIS IS IMPORTANT

If *TargetCo* outsources or offshores any critical software development or IT functions, there are number of issues to investigate:

- What is *TargetCo's* future ability to support the software if it was not developed in-house?
- How well documented is the software?
- Does *TargetCo* physically possess or own the legal rights to the source code? There are many issues around contract language and proper copyright assignment when it comes to contracting for software development. You can easily end up not owning the rights, even for custom development, without the proper contract language in place. At the same time, in certain states and countries, you can inadvertently create an employment relationship with the contractor if the wrong language is used.
- Where was the software developed? If outside the United States or EU, are the copyright laws in that country enforceable? If critical

software was developed offshore, you'll never really be sure that a copy of it isn't floating around in cyberspace.

Seeing the agreements will allow you to understand the contractual obligations of both sides, including any ongoing commitment to use or pay the outside developer or company.

Here is a good overview of the US law around ownership of rights related to contracted software development:

http://www.acc.com/legalresources/quickcounsel/qc-software-work-for-hire.cfm

DUE DILIGENCE REQUEST

A description of the software development model that is used (examples include "agile" or "waterfall" and may include more specific models such as "Scrum").

WHY THIS IS IMPORTANT

It's beyond the scope of this book to fully explain the models listed above. At a very high level, agile software development is an iterative, fluid and inherently flexible process that is a relatively new concept. Waterfall is a more traditional, structured software development process that has its origins in manufacturing industries. Neither is inherently right or wrong, and the best choice depends on the situation.

Here is an interesting read and a good primer on the pros and cons of agile vs. waterfall software development methodologies:

http://arstechnica.com/information-technology/2014/08/how-microsoft-dragged-its-development-practices-into-the-21st-century/

Software development methodologies are a question of compatibility between *AcquiringCo* and *TargetCo*. Developers are often very attached to the methodology they use for developing their software, and making a change to accommodate the standards of *AcquiringCo* may be very disruptive to *TargetCo's* technology staff. In addition, if there is no specific development methodology, or no one you speak with can describe it clearly, that raises a red

flag and would tend to make you question the professionalism of the software development team at *TargetCo.*

DUE DILIGENCE REQUEST

A description of the process in place for identifying, tracking and correcting product bugs.

WHY THIS IS IMPORTANT

This is a yet another good indicator of the overall organization and professionalism of *TargetCo.* Is there a product roadmap that guides software development, or is the company constantly in reactive mode in which newly discovered bugs derail new functionality?

Ideally there is a separate group or individual developer that works on product bugs and issues while the main development team drives product features and functionality according to the product plans.

DUE DILIGENCE REQUEST

A description of the process for designing product user interfaces.

WHY THIS IS IMPORTANT

Here is another area where you get to judge the professionalism of *TargetCo.* If the user interfaces of the products are designed by the programmers, chances are they're not as good as they can be. It's very difficult to find somebody who's both an excellent programmer and a qualified user interface designer. The more true expertise that has gone into *TargetCo's* products, the better. To be fair, it's not common to find a small to medium sized software company that has this expertise on staff or has made the commitment to obtain it through consultants or contractors.

DUE DILIGENCE REQUEST

A description of the database design process.

WHY THIS IS IMPORTANT

Similar to the previous question about user interface design, it's a rare software developer who is also an excellent database designer. Has a true database expert designed *TargetCo's* databases, providing a good foundation upon which to scale and grow, or did the programmer for the product that uses the database create something that was just "good enough?"

Again, the latter situation is not unusual, but the database should be reviewed during the source code review to determine if a scalable design is present. This very much "behind the scenes" functionality can require an expensive rewrite if it's not up to the task of the planned growth.

DUE DILIGENCE REQUEST

A description as to how product security is considered during product development.

WHY THIS IS IMPORTANT

Does *TargetCo* have a consistent approach to security from product to product, or does every product handle it differently? It can be very difficult and potentially expensive to add good security to a product once it's already developed.

One quick way to determine how carefully security has been considered is to determine how users access the company's products. If there is a login required, does the same login work for every product? A separate login for each product most likely means a separate security mechanism for each product. This is a sure way to frustrate your users if you would expect one user to access multiple *TargetCo* products.

DUE DILIGENCE REQUEST

A description of the software development coding standards used by the company.

WHY THIS IS IMPORTANT

A professional software development organization, no matter the size, will have this. A coding standards document lays out the in-house programming standards, naming conventions and the specific technology choices that will be used in the company's software development efforts.

Without a defined and enforced process, every developer may create software with a different style and technique. This makes is difficult for another employee to take over the work, and for a new employee to come up to speed on the company's source code. A lack of coding standards can make the company very dependent on a specific employee to always be available to support a piece of software.

I've seen independent, individual developers with a coding and naming standards document to keep their own work organized and consistent, so it's not too much to expect that this exists in a company of any size or age.

DUE DILIGENCE REQUEST

Explain the company's mobile development strategy.

WHY THIS IS IMPORTANT

In 2014, various studies of Internet traffic showed that, for the first time, mobile usage of the Internet exceeded that of desktop. This means that if *TargetCo* has a website, you'll want to be sure it works well on mobile devices.

This is especially important if there is any ecommerce component to the site. Companies without a website that is usable on phones and tablets will be at more and more of a disadvantage going forward, and could be losing sales.

In addition, in February 2015, Google announced[3] that it would begin in April 2015 to more heavily weight the mobile-friendliness of websites when

---

[3] "Finding more mobile-friendly search results," *Google Webmaster Central Blog,* February 26, 2015. Accessed on March 8, 2015.
http://googlewebmastercentral.blogspot.com/2015/02/finding-more-mobile-friendly-search.html

calculating its search result rankings. If *TargetCo* counts on search engine traffic from Google, this should be a key consideration.

There are a number of software development approaches that can accomplish the task, including responsive web design, a designated mobile site or native mobile apps.

If no one at *TargetCo* can explain the thought process the company has gone through up to this point regarding mobile strategy, consider that a warning sign.

If *TargetCo's* market situation requires a mobile presence and one does not already exist, you'll want to consider that in the post-integration budget and the overall cost structure of the transaction.