Using Risk Mitigation Approaches to Define the Requirements for Software Escrow

Submitted in partial fulfilment of the

requirements for the degree of

MASTERS IN SCIENCE

of

RHODES UNIVERSITY

Karel Rode

June 2014

Abstract

Two or more parties entering into a contract for service or goods may make use of an escrow of the funds for payment to enable trust in the contract. In such an event the documents or financial instruments, the object(s) in escrow, are held in trust by a trusted third party (escrow provider) until the specified conditions are fulfilled. In the scenario of software escrow, the object of escrow is typically the source code, and the specified release conditions usually address potential scenarios wherein the software provider becomes unable to continue providing services (such as due to bankruptcy or a change in services provided, etc.).

The subject of software escrow is not well documented in the academic body of work, with the largest information sources, active commentary and supporting papers provided by commercial software escrow providers, both in South Africa and abroad.

This work maps the software escrow topic onto the King III compliance framework in South Africa. This is of value since any users of bespoke developed applications may require extended professional assistance to align with the King III guidelines. The supporting risk assessment model developed in this work will serve as a tool to evaluate and motivate for software escrow agreements. It will also provide an overview of the various escrow agreement types and will transfer the focus to the value proposition that they each hold.

Initial research has indicated that current awareness of software escrow in industry is still very low. This was evidenced by the significant number of approached specialists that declined to participate in the survey due to their own admitted inexperience in applying the discipline of software escrow within their companies. Moreover, the participants that contributed to the research indicated that they only required software escrow for medium to highly critical applications.

This proved the value of assessing the various risk factors that bespoke software development introduces, as well as the risk mitigation options available, through tools such as escrow, to reduce the actual and residual risk to a manageable level.

Acknowledgements

My wife, Beatrix has, to the end, been very tolerant and patient and I want to thank her for her constant motivation and concern over my progress with assignments as well as this thesis. Her support and unreserved commitment to my success in writing this thesis has not gone unnoticed and I wish to acknowledge the value she adds to both my personal and professional life.

To the individuals that have assisted me in research - end users as well as the three escrow providers that have willingly contributed in time, material and words of encouragement - I say thank you., You have assisted in the making of something that at one time looked very difficult. Thank you for helping me realise my goal.

To my supervisors, Dr Alapan Arnab and Prof Barry Irwin who have provided constant guidance, comments and review, thank you for assisting me throughout this endeavour. Further thanks to Professor Barry Irwin and the team at Rhodes University for accepting me onto this Masters program.

Finally, thanks to my friends Rob Watson and Dr Werner Swart, who performed the initial reviews of this work.

Table of Contents

List of Figures

List of Tables

List of acronyms and abbreviations

BCP	Business Continuity Planning
CDMA	Code division multiple access
CD	Compact Disk
CISSP	Certified Information Systems Security Professional
DR	Disaster Recovery
DVD	Digital Versatile Disk
EU	European Union
GAK	Government Access to Keys
IP	Intellectual Property
IT	Information Technology
JD	Job Description
LEAF	Law Enforcement Access Field
MM	Maturity Model
MD5	Message-Digest Algorithm 5
NSA	National Security Agency
SHA1	Secure Hash Algorithm
SaaS	Software as a Service
SDLC	System Development Life Cycle

CHAPTER 1 – Introduction

The central focus of this research is reducing the risk that a user of bespoke software may experience in case of an unforeseen disruption of the support and service that the software developer provides. In this section we will discuss the history of escrow in various forms as well as the common risks that end users should attempt to address through the use of a software escrow agreement.

According to the Merriam-Webster Dictionary, escrow is[1] "*a deed, a bond, money, or a piece of property held in trust by a third party to be turned over to the grantee only upon fulfilment of a condition"*. Black's Law Dictionary describes escrow, from a legal point of view, as a deed delivered by the grantor into the hands of a third person, to be held by the latter until the occurrence of a contingency or performance of a condition, and then by him (the third party) delivered to the grantee (Black's Law Dictionary Free 2nd Ed. and The Law Dictionary, 2013).

The best example following this description is often seen in real estate dealings. Escrow is commonly used in the field of property sales, where the estate agents will be dependent on the use of a funds escrow managed by the transferring attorneys (Jennings, 2013). These attorneys will act as the curator of the funds received from the purchaser and place it in a trust account. Once the deeds registration and transfer conditions are completed the curator will pay over said funds to the seller and estate agent (Miller, 1982).

Escrow is not only used in property transactions, and there are various forms of escrow that can be used in legal transactions. As per the lyrics of "Sue Me, Sue You Blues", wherein musician George Harrison transcribed the process of escrow to song:

Hold the block on money flow
Move it into joint escrow
Court receiver, laughs, and thrills
But in the end we just pay those
Lawyers their bills[2]

It was whilst doing reading for the ISC2 Certified Information Systems Security Professional (CISSP)[3] certification that the topic of escrow - "*The safe keeping of an entity which two or more parties have an interest in."* (Ince, 2009) - became a topic of interest. Many areas of the IT software development lifecycle, as well as, most centrally key escrow for encryption, came up for review. With over 12 years of direct insurance experience the researcher tended to see the purpose and value of risk assurance in many aspects of IT governance and risk management. Moreover, in the sphere of software code development, with various parties involved from supplier and customer to consumer, each will have individual agendas to ensure levels of continuity, access and succession as well as protection of intellectual property.

[1] http://www.merriam-webster.com/dictionary/escrow
[2] http://www.azlyrics.com/lyrics/georgeharrison/suemesueyoublues.html
[3] http://www.isc2.org

For the purpose of this research, the focus will be to discuss software escrow only, which can be defined as "*placing original software programs and design documentation into the trust of a third party"* (Cannon, 2011).

During a CompTIA webinar in May 2012, Craig Motta from NCC Group shared a number of statistics with the information provided referenced as "independent research". The following is an extract from that data (Motta, 2012):

1. 97% of organizations were reliant on critical software applications, yet 82% were unprotected should the suppliers of these applications fail
2. 63% of organizations had a software risk assessment policy and process in place yet only half of them always carried out a risk assessment on new software applications
3. 87% of organizations claimed to have a business continuity/disaster recovery plan in place yet only 12% of them have escrow protection specified within it
4. 97% of organizations had business-critical software applications yet only 18% of organizations had protected all of these with escrow agreements with a further 26% protecting some
5. The IT Director/Manager was responsible for implementing escrow protection in 58% of organizations - 13% of organizations didn't know who was responsible

At the time that this research commenced, it became clear that the last point noted above also held true for the respondents that were approached. Each of the points above will be elaborated on in this thesis as they are each individually valid. Together they paint a gloomy picture of the underutilised potential of software escrow within most commercial organisations.

It can be proposed that software escrow is a type of insurance where the resulting insurance 'pay-out' acts as a compensating control that providing cover after an uncontrollable eventuality. Within the agreement the contracting parties agree to specific eventualities such as an act of God or business failure. As these eventualities are not, by design, within the control of the organisation, they serve as the trigger mechanisms for the desired outcomes such as the release of source code. A review of what risks can be mitigated - through a software escrow agreement for example - is conducted, as well as of the residual risk to the business after such an agreement is concluded.

In ISACA (2009) it is stated that "Management of business risk is an essential component of the responsible administration of any enterprise. Almost every business decision requires the executive or manager to balance risk and reward". As the developer and maintainer of major IT management standards such as COBIT, Information Systems Audit and Control Association (ISACA) provides a good guideline on the alignment of IT risk management and business objectives; these guidelines are incorporated by organisations that choose to follow the IT frameworks promoted by ISACA. Similar concepts are expressed in the King III report (Engelbrecht, 2009) where it is clear that the directors cannot discharge their accountabilities to manage risk as per "*Principle 4.5: The board should ensure that risk assessments are performed on a continual basis*".

When a company relies on a bespoke developed application they should consider the risk of business disruption that can be introduced. When software fails at an inopportune time, software that is key to the successful operation of the business, one might argue that from a business continuity perspective, the software purchaser should have the right to acquire the source code. In reality, the business only gains access through the license to use the executable code within the agreed licensing framework in addition to ongoing support, maintenance and development enhancements, as was agreed to in the software purchase agreement. Significant failure, therefore, in the software or support of the software would call for remedial action; it is the board's duty to address such risks, as will be highlighted later with examples from the King III corporate governance code in Section 2.7.

1.1 Research Questions

While there is a significant amount of commercial and marketing material available on software escrow, critical academic investigations and external influence into software escrow are limited. Major software escrow providers' web sites provide some case studies and white papers as well as marketing material that expands and positions the various products and solutions that they all represent.

1.1.1 Primary Research Question

The primary research question when conducting this research is: can software escrow be seen as a suitable Risk Mitigation Tool within the South African business environment?

The first objective will be to gain a better understanding of what software escrow is and what risks in general it aims to address or alleviate. This will be covered in the literature review. We will question what risks contribute to and influence sound decision making in moving forward with a software escrow agreement. This section will thus provide a guideline of those items to consider, inclusive of a request for proposal or tender. Moreover, these questions are critical to ensuring that we address all criteria a business might consider when viewing different escrow options, as well as some items that pertain specifically to the handling and storage of escrowed content.

Legislatively, South African contract law does not differ much from the UK or Australia (Du Plessis, Hutchinson, & Pretorius, 2011), although South African common law originates from Roman-Dutch law and its contact law is based on common law supplemented by specific legislation and case law. Thus, although the research is focussed on the South African business environment, the findings could extend to countries with similar legal and commercial frameworks.

A second objective of this research is to map the need for software escrow against South Africa's King III compliance framework. The King III compliance framework provides guidance for adhering to corporate governance (a requirement for all companies listed on the Johannesburg Stock Exchange) and is backed by the South African Companies Act (*Companies Act 71 of 2008*). Similarly, in the United States, the Sarbanes-

Oxley Act (2002) was implemented for corporate governance. However, the American Act follows a model of 'comply or else' and financial penalties may be levied if the guidance is not properly followed. In South Africa the prior versions of the King reports used suggestions such as 'should', but never 'must', and this is now reinforced in the King III framework where the "comply or explain" principle is followed. That is, any company in South Africa, that implements a framework for good governance practices other than King III, must provide additional explanatory evidence in the annual financial report to explain such deviations.

1.1.2 Secondary questions

Two secondary questions are also considered, as they are key to long term planning for business continuity and the structure and contents of the agreement between the participating parties:

- Will the business continuity planning and disaster recovery benefit if critical business applications were covered with source code escrow agreements?
- What makes up an acceptable source code escrow agreement for use in South Africa as there are variations in the levels of inspection and the handling of deposit content?

These two points relate directly to the third objective: namely, the need for software escrow agreements. It will be investigated further as to how these agreements are structured and how they are called on when the execution of the stated terms are commenced.

1.2 Research goals

Discussions of software escrow in South Africa, whether in a professional or academic context, have unfortunately been very limited. Providing education on the topic of software escrow, particularly to IT and Risk managers, directors and compliance officers is therefore a main objective of this thesis. This thesis will indicate that it is desirable to map the software escrow topic onto the King III compliance framework for users of bespoke developed applications. Our risk assessment model will serve as a useful tool to evaluate and justify the need for software escrow agreements and our review of the various escrow agreement types will shift the focus to the value proposition that they each hold.

1.3 Research method

The view that software escrow is actually a risk mitigation tool was formulated by the researcher from an initial understanding of the subject based on the value proposition that it represents, however not much has been published on this subject other than items related to software development, licensing and legal matters pertaining to software. Moreover, the attempt to reach out to local software escrow providers to gain cooperation for the initial research interviews presented an understandable challenge: almost all such contact

initiated by them would be in violation of the confidentiality agreements with their respective clients. At this phase service provider assistance was then limited to a review of the proposed model only.

The next step was to engage with various licensees within a number of large financial sector companies, mostly in the South African banking sector. This phase of research highlighted that the actual use of software escrow is limited, and that the persons working within these organisations (those that actually deal with Software Escrow) are limited to a few key individuals that do not make it publically known that this is an area of key expertise or focus.

A choice of proceeding with a very limited number of personal interviews was compared with proceeding with a series of case study reviews and defining a risk model that could then be presented to a select end user base for commenting on. The latter approach was selected as suggested by Creswell (2003), as represented in Figure 1.

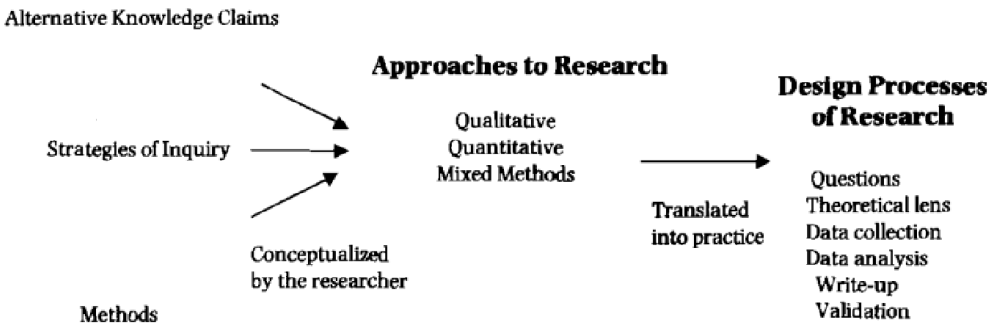


Figure 1 - Knowledge Claims, Strategies of Inquiry, and Methods Leading to Approaches and the Design Process (Creswell, 2003)

Creswell's model is applied to existing knowledge and experience, augmented with research and user inputs from the research groups that participated in the survey. The format of the survey was built out of various data sets as well as through personal interaction with one escrow service provider.

1.4 Key terms defined

Throughout this document terminology specific to the field of software escrow will be used. Our aim, with this list, is to provide the reader with an overview of these key terms for clarity.

Escrow - As defined in the *Oxford Dictionary* (2013), the verb "escrow" means "*to place in custody or trust until a specified condition has been fulfilled*."

Deposit media/objects - Any computer readable media used to store digitally created content. This may include CD or DVD and in some cases will include the electronic file transfer of data between the providing party and the agent.

Escrow deposit list - A comprehensive list of items detailing what is included in the actual deposit. Additional data should include the software provider name and address, software name, version, minor and major versions, date of deposit, software fingerprint, encryption keys or contact details for the custodian of the encryption keys, passwords and other relevant data.

Escrow agent - This is the contracted party that will be the intermediary between the software provider and the software purchaser. It is also the party that will review the escrow agreement terms and act on the stipulated release conditions, if and when required.

Escrow system - The system that is used to manage the contracts, relations between the contracting parties, as well as billing and access to the data that is contained in the vault, inclusive of customer identifiers and lookup systems.

Intellectual property (IP) owner - The creator of the intellectual property (IP) that makes up the bespoke application or software system.

Licensee - The end user or organisation using the software that is licensed to use the intellectual property as was created by the software provider.

Licensor - The party that creates and commercially provides the software to the licensee and the owner of the intellectual property that is the end result of the features and functions of the software. The licensor is not necessarily the IP owner and this may be defined in the software licensing agreement.

Object code - The executable code that a computer generates from source code after it was processed by a compiler. It is generally not humanly readable code and consists of a number of commands and instructions that can be interpreted by the computer operating system or applicable object code interpreter.

Package - A comprehensive set of items that will be passed on to the escrow agent for safe keeping. In the context of this writing it will be the media, instructions and supporting documents as per the deposit list above.

Software Provider - The party that produces the desirable software.

Software Purchaser - Also see Licensee. The person or organisation that buys or licenses the software to use.

Source code - The human readable code, prior to being processed by a complier, which made up of the commands and instructions that execute the functions and features of the software.

Storage facility - A safe and secure area where acceptable access control is enforced with a fire suppression system and proper heating, ventilation and cooling so that any items stored will not degrade, become damaged or misplaced.

Verification - The act of inspecting and comparing the media and associated content to accurately represent what was stated on the escrow deposit list.

1.5 Document structure

The remainder of this document is structured as follows.

Chapter 2 provides a literature review, introduces the fundamental components of escrow, and highlights significant concepts within escrow such as the deposit workflow, inspection of the escrow contents, release conditions and procedures, escrow agreement types and fees as well as other types of escrow commonly in use. Finally, this chapter references a number of case studies that highlight some key elements of software escrow.

Chapter 3 is concerned with a number of case studies that highlight how software escrow has performed in the past as well as what to expect when there is no agreement in place.

Chapter 4 is focused on the research approach, research items formulated for the research as well as findings and comments received from the various participants.

Chapter 5 provides commentary on the responses received from the participants when working with the conceptual model to determine the perceived risk that could be faced when engaging an independent software provider.

Chapter 6 presents the conclusion of this research and evaluates areas of potential future research.

Addendum A deals with supporting information as well as the questions that could be used to qualify the software escrow agent in terms of service offerings and capabilities.

Chapter 2 – Literature Review

The purpose of this chapter is to provide sufficient background on the uses of escrow, various escrow agreement types and components for the deposit, inspection and release of software held in escrow. From this point of familiarity, it will then be possible to expand the discussion to include other uses of escrow, its potential benefits for cloud based services, and also the darker use of escrow services in the fraud underground.

This chapter will first provide a background on South African governance in this area. It will then address the fundamentals of escrow, the participating parties involved and provide a review of key concepts, such as inspection of deposit content and the supporting workflow from deposit through to release conditions. Investigation of the types of agreements that exist today are now possible, be it the standard tree party agreement or the type that applies to multiple licensees.

As these licenses differ and each license will be tied in with the level of inspection that the participating parties selected, it is expected that different fee structures will apply and it will pertinent to determine what else could impact the fee structures.

The prominence of Software as a Service (SaaS), application stores for mobile computing devices and the darker side of software escrow also needs to be considered to complete the picture.

2.1 Background

Whilst doing the initial search for applicable research papers it soon became apparent that the researcher would be limited in the number of sources that could be referenced, as this is not a well-researched topic in South Africa. The goal of this review and the related research was to determine how pervasive the current use of software escrow is, how it is implemented by the various parties involved and how software escrow functions should the terms of the agreement be called to action.

The lack of local papers and opinions within the South African context on the subject of software escrow, , led to the use of both ‘traditional’ research procedures and also contact with existing escrow service providers, both locally and internationally. In the process it was possible to identify some local resources that could be included in the research questionnaire, in order to create a local view of software escrow performance in South Africa. Local academic content in this domain cannot compare to the available marketing material and local media content, forcing an expanded research into the legal domain, IT audit standards as well as international research.

2.2 The fundamentals of Software Escrow.

The phrase 'fundamentals of an escrow agreement' refers to a formal and binding agreement between three different parties. The owner/creator of the software will be referred to as the depositor or 'the licensor'. The party for whom the depositor is creating the software, normally delivered as compiled code or freestanding software will be called the beneficiary, or sometimes the licensee, and will be the party that will benefit from the terms in the agreement. Lastly the agent is a mutually acceptable third party who will act on the conditions of the agreement and ensure that the items of the escrow agreement are available. The agent may also provide additional services such a code verification and access control (this subject is addressed in more detail in Chapter 2).

Other common forms of escrow appear in the legal industry where trust accounts are used during financial transaction escrow (Everett-Nollkamper, 2003). Escrow is also used commonly in shipping; a firm may demand a deposit to move, say live poultry from a farmer (the producer) to the market (Botterill, 2012). The market, however, may only want to pay the producer for the goods once they arrive in good order. In addition the farmer, having an expectation of receiving payment for live poultry delivered alive and well to the shipping agent may also have an expectation that the live cargo is shipped and produced to the market agent in a similar condition as when handed over. In this case a slightly more complicated escrow agreement is formulated with release conditions for the funds from the purchaser (market) to the farmer who is dependent on the performance of the shipping and handling agent. The farmer will proportionally penalise the shipping company for losses greater than a fair trade acceptable level, influenced by travel distances and external forces such as the weather(a heat wave, for example, may impact sales, and could even provide a performance bonus for better than expected results). Ultimately it will be the market's goods receiving department that will indicate to the escrow agent how the shipping agent performed, triggering the release of the payments to the involved parties.

In the digital world public key infrastructures can have key escrow working for our benefit (Ince, 2009). This technique is generally considered a good compromise between the needs of law enforcement agencies to access messages sent over a network and the requirements of personal privacy. A key escrow scheme involves the user of an encryption algorithm depositing the key used in the algorithm with a trusted third party, such as a bank. If a law enforcement agency requires the key it is then made available to them by the third party only after a warrant has been issued and inspected for its appropriateness to the key release. There is also another practical rationale for such a service: if the holder of a key loses it or it is destroyed then all the data associated with the key can be lost. Having the key held by a third party guards against this and provides for a recovery option of the data. The third party in these cases are often known as a trusted third party.

In 1993 the U.S. National Security Agency (NSA) proposed the introduction of the Clipper chip, a device that was to be used in phones that supported encrypted telephony (McMurdo, 1996). One half of the binary keys for each chip would be placed in escrow with two separate government agencies and through a court

order they (the NSA) could gain access to the full key pair to eavesdrop on conversations. This was also termed Government Access to Keys (GAK) (Schneier, 2000). Three years later, in 1996, the system was no longer in use as it was exposed that the improper use of the Law Enforcement Access Field (LEAF) would make the system easy to compromise by a user (Blaze, 1994).

In general, an encryption key escrow system is established where external parties may log encryption keys such that on specified conditions a number of these participants will have to convene to access the keys so that a recovery key can be formed that will be used as a master key (Caelli, 1996), (Blaze, 2011). Such a key will be required to decrypt any data for which the individual keys were lost or withheld, causing law enforcement or management to provide for an agreed alternative workaround.

In contrast, a software escrow agreement is often initiated by the beneficiary, also called the licensee or user company or software purchaser, and will have specific conditions that could trigger the release of the source code to the beneficiary. These well-defined release conditions will have to be breached by the depositor or software provider, as they created and own the licensed intellectual property (IP). Finally, the escrow agent will act in the interests of the two parties, ensuring that the stipulated release conditions of the escrow agreement are met before releasing the code to the beneficiary. A common condition for the release of source code may be bankruptcy.

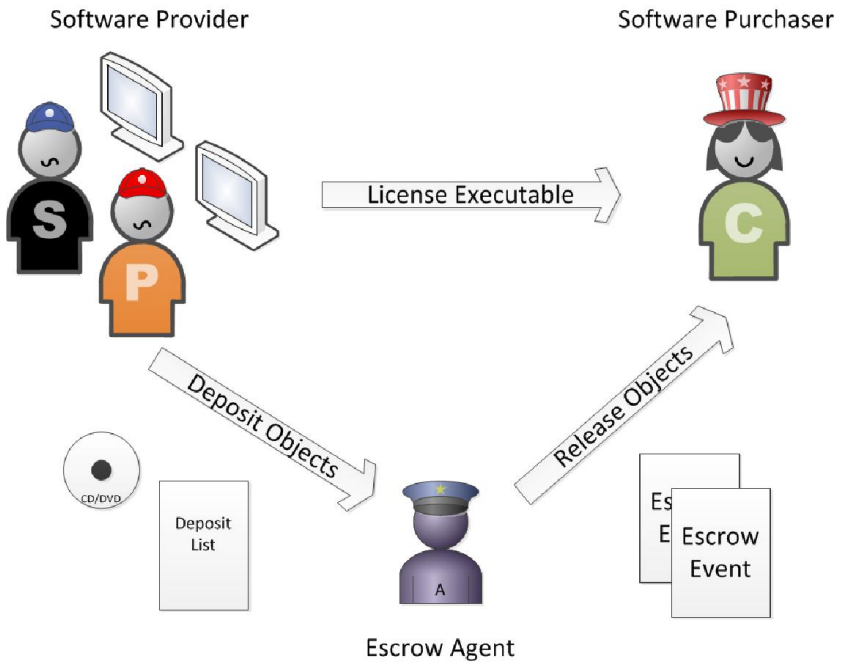


Figure 2 - Relationship between escrow participants (Draws, et al, 2011)

In Draws et al. (2011) a useful graphic depicts the relationship between the three parties in an escrow agreement, as well as three supporting events between the different parties. These events, as can be seen in Figure 2, include the deposit by the software provider (depositor) or the placement of the source code with the agent (deposit objects), the release of the licensed software (license executable) to the beneficiary (purchaser) as well as defining the act(s) that will result in the agent releasing the source code to the beneficiary.

The NCC Group[4] (Steuerbrerater & Wirtschaftsprufer, 1998) defines a licensee as "*An individual or organisation who is licensed to use, or resell, the software developed or supplied by the software owner and who is, or will be, party to a source code agreement*". Moreover, Blatt (1998) refers to the end user as a licensee and to the provider as the licensor. These references underscore the more legal view taken by such experts and serves as a guide to understanding the legal language that forms part of the escrow agreement. Readers need to remain vigilant in understanding the terms and legal reference framework that is used throughout the various documents.

In his book, titled *Software Escrow For Dummies* Richard Kane (2009) describes the following items as the benefits of software escrow:

1. "Serving as a safety net for investment in software, technology, services, and other intellectual property."
2. "Providing controlled access to a vendor's proprietary source code under specific release conditions with limited use rights."
3. "Helping to build trust with your software vendors."

The areas of validation, integrity and applicability of the software escrow are also open to interpretation. From reading on the topic it is clear that some (master) agreements support the notion of code validation. These agreements are reviewed later on in this work and pertain to the legal agreement between the contracting parties. Gartner notes that "*If you don't plan to perform regular audits or verify that the version of the software you are using is in escrow, the agreement may be worthless. If the vendor falls behind on source code deposits and has incomplete or unusable deposits, the escrow agreement will be useless*" (Stekhoven, 2010).[5]

When considering that a depositor may release several major versions of the software over a given time period, as well as updates throughout the years of use, it may be possible to see many variations of the code placed into escrow. This will impact the end user company software version and release management as well as end user support. Care should be taken about the amount of effort and complexity, as well as various technologies, that may be required to do such source code validation. This research will cover this area to determine if there are practical approaches to this problem given that the risk of not doing proper code verification is very real. It may also be determined that the beneficiaries remain comfortable knowing that some code was deposited with the escrow agent.

It has been noted (Out-Law.com, 2010) that the UK based NCC Group had a significant increase in the number of releases that was made in that year. NCC Group showed a 150% increase over the previous year

[4] NCC Group is a global information assurance specialist providing organisations worldwide with expert escrow and verification, security testing, website performance and software testing services.
[5] Gartner, Inc. stated in a Research Note published in February 2005 – Alternative source of the quote as Gartner content is subscription based (http://www.ironmountain.com/Knowledge-Center/Reference-Library/View-by-Document-Type/White-Papers-Briefs/Sponsored/Trust-But-Verify.aspx)

and the author points to the economic downturn as the reason for initiating release conditions. Of interest are the four release conditions cited within this article. These are listed as:

1. Going out of business
2. Willingly ceasing to trade or through bankruptcy
3. Failing to meet the maintenance agreements
4. Assigning the intellectual property rights to another party (company).

In the latter case the new owners need to honour existing escrow agreements, as failure to do so within the assigned time frames could trigger the typical release conditions as mentioned above with a subsequent impact on their customer base through negative publicity.

The above implies that a number of legal issues may need to be considered within the software escrow landscape and that legal guidance needs to be aligned with the constituencies where the participating parties operate. With proper guidance a licensee can gain protection by enforcing software escrow and ensuring that the agreement allows for the recovery of the escrowed materials during a bankruptcy (Info-Tech Research Group, 2009) or any other defined release condition. A licensor might also protect itself against a licensee rejecting the agreement by including special early termination fees (Riskin, 2012). Such provisions may help increase the licensor's unsecured claim and assist in the recovery of the software in escrow, in the event of a bankruptcy case for example.

Aglin (2000) highlights a number of issues associated with the drafting of the software licensing agreement and terms as well as the impact of the US Bankruptcy Act (Bankruptcy Reform Act, 1999)[6] on such legal agreements. It is the author's opinion that careful consideration needs to be given to such matters within the South African legal context as the ownership and succession planning and management of source code, where a deemed value may be attributed to such goods within an estate, could have undesired consequences if not properly understood. The specific legal conditions to trigger escrow have not been investigated further as part of this work.

It is also clear from Prozesky-Kuschke (2006) that South African case law and litigation is not available to formulate additional views as expressed by Jeremy Speres (2012) where he states:

> "*If the liquidator decides to terminate the contract, the other party to the contract is merely left with a concurrent claim for damages*". Speres proceeds to stare that "*In the South African case of Video Parktown North (Pty) Ltd v Paramount Pictures Corporation; Shelbourne Associates & Others; Century Associates & Others, the court classified a licence agreement as a pactum de non petendo, an agreement not to sue. Therefore, it is likely that a licence agreement will be considered executory and liable to rejection by the liquidator, leaving the licensee with an unsecured claim for damages against*

[6] Bankruptcy Reform Act of 1999, as amended, which was codified in Title 11 of the United States Code and commonly referred to as the "Bankruptcy Code"

the developer's insolvent estate and without any right to use the source code, despite the fact that the development arrangement was subject to an escrow agreement."

This could probably attributable to the very low rates of withdrawal that have been indicated. Interestingly, only one report can be found highlighting the growth of withdrawals at 150% over the prior year, as reported in Out-Law.com (2010) for 2009, as a result of the economic recession. This equates to a low number of 150 releases of the over 8000 deposits at NCC in the UK only, versus the only 50 the prior year. The fact remains that actual withdrawal rates remain low compared to the levels of deposits and this was anecdotally confirmed in discussions with local escrow agents. This low withdrawal rates needs to be remembered when we get to the research items, as the very low rate of withdrawal needs to be shown in the options available.

A further question needs to be answered; namely, 'who needs escrow?' Bruno (2003) has asked both risk-related questions pertaining to the size and stability of the software provider and also those relate to the business impact and criticality of the developed software. In stating that the software is only used by a small subset of users, with no significant revenue impact on the company, it is safe to assume that the need for software escrow is negligible. However, any degree of user base reliance that could directly impact the bottom line, putting the company in a position where the software is mission critical, would constitute a risk profile where it would be appropriate to seriously consider the use of escrow.

Christiansen (2004) suggests that the need would be reduced if confidence in the software house is high with regards to their financial stability; he argues that in this case they will not breach the maintenance clauses of the licensing agreement. He goes on to state that such stability is not the only thing to rely on and that an escrow agreement should be implemented on all mission critical applications.

A valid question at this point would be, 'what would one do if one suddenly finds that the software is out of maintenance and failing spectacularly when it is least convenient?' What would be needed to accelerate the recovery process in such a case? If it would be a major inconvenience, then a source code escrow agreement could benefit a smooth recovery as would an alternative software solution, with its associated costs for deployment, training of staff and services disruptions as the transition is initiated.

Bruno (2003) further states that the escrow agreement needs to be treated with the same diligence that is afforded the software licensing terms and agreement. I will expand on this later in the specific sections pertaining to release conditions and release procedures. Considering the above views it was determined that a risk-based approach will be taken to measure the need, impact and cost implications of various bespoke developed software solutions.

2.3 Clarification of Significant Software Escrow Concepts

A number of key concepts within the software escrow domain need detailed expansion as comprehension of each of these items will benefit the reader, both in determining the interrelationship between various working

parts and also in noting the importance of some of the components and higher levels of protection that some have over others.

2.4 What should form part of the Escrow Deposit?

Two papers expand on what to place with the escrow agent. In 2010 it was determined by Iron Mountain (2012a) that over 70% of deposits reviewed were found to be lacking. A comprehensive Escrow Deposit List was produced by Iron Mountain (2012b), and it is included below for reference and further discussion:

1. All source code modules that comprise a single application
2. Both compilation and setup/installation documentation
3. Documentation that describes the integration requirements with other applications
4. Dynamically linked libraries
5. Description of external data sources and/or data feeds
6. Database setup and configuration instructions
7. Database schema and data modelling documentation
8. Hosting configuration instructions
9. Maintenance tools
10. Proprietary or third-party system utilities for build and installation
11. Instructions on "where to get third-party utilities" and "how to deploy"
12. Descriptions of the system/program generation
13. Names and home addresses of key technical employees of the developer
14. A list of any encryption keys or passwords used in the creation or backup of the escrow deposit
15. A copy of the executable with data in SaaS/licensing arrangements – the data is as important as the code

A review of Monnet (2011) provides an additional list of items that should be considered for inclusion into the escrow agreement. Items such as the *functional specification, architectural and design specifications, user guide and reference guide* are all documentary evidence that can also sustain claims against the software licensee should they have the need to defend software ownership in a court of law. A well-documented case within the context of the South African law discusses the ownership of the bespoke developed code (de Kock, 2010) and the implications of owning either components or the whole source code. It is therefore in the interest of the licensee that design and source documents are also placed in escrow. Moreover, this can serve as further evidence of ownership for the service provider should such disputes occur.

Logitas (Monnet, 2011) goes further, and defines an Honour Declaration:

> "*the development environment description is an increasingly important piece of evidence of ownership. The environment specifics are not chosen at random. It is a fact that only the legitimate*

owner of the software is able to correctly reinstall the development platform and explain the choices made consistently with the deposited source code."

One needs to consider that the ownership of the software could be a contentious issue. An important reference point in this area is the South African case of Haupt t/a Soft Copy vs Brewer Marketing Intelligence (Pty) Ltd and Others 2006 (4) SA 458 (SCA) (de Kock, 2010), also referred to as "the Haupt case". Emmie de Kock, in her overview of the case (de Kock, 2010), states "*the Copyright Act provides that 'the person who exercises control over the making of the computer program' is the 'author' of the computer program*". She concludes that the expense of the litigation could have been negated had the two parties reached a proper agreement.

On the subject of completeness of the deposits, as per the deposit list, it was found that (Bruno, 2003) up to one quarter of the verifications of software in escrow fail to compile as they are missing components key to the success of any agreement. Bruno further urges parties to validate subsequent deposits with the same vigilance.

It is clear that the fifteen items listed above by Iron Mountain are key elements to include in the Escrow Deposit List and should not only be part of the initial deposit but actively maintained throughout the software escrow agreement timespan, ensuring that all further deposits are done with similar diligence.

2.5 Deposit Workflow

When considering the above items that need to be placed with the escrow agent it is also important to review the way that an escrow is handled. Some basic escrow providers will only accept material on compact digital media (CD and/or DVD) and this will be archived with specified additional administrative documentation. These will then be placed within a customer specific container within a secure vault or other suitably secured area.

Although this could be sufficient as a standard or entry level of escrow, the reader needs to be mindful that inspection of the deposit material (as detailed below) will not be possible or would be more taxing on the participating parties if an unstructured way of depositing the material is followed. One local escrow agent overcomes some of these challenges by physically overseeing the initial deposit activities, ensuring that the deposit list is properly drafted and complete. Moreover, he also personally validates the readability and integrity of all digital media whilst at the software provider's site.

A good example and guidance on how to properly deposit the code is provided by Iron Mountain. Figure 3 was sourced from this document.

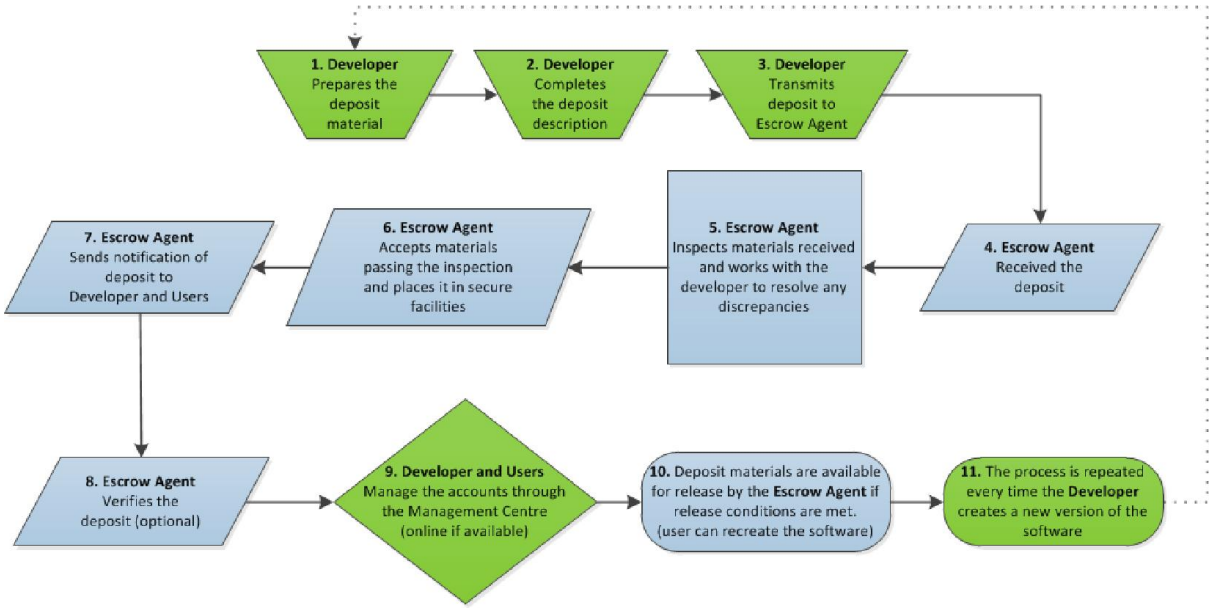


Figure 3 – Iron Mountain Software Escrow Workflow (Iron Mountain, 2012a)

The first three items relate to the activities of the developer (software provider) in preparing for the deposit and the shipment of media or file transfer. Although these steps are very simply stated the importance of the supporting artefacts such as the escrow deposit list, inclusive of the 15 items as described in Section 2.4 (What should form part of the deposit) must be emphasised.

On receipt of the package the escrow agent needs to proceed with the inspection as described below and needs to inform the participating parties of the deposit, inspection outcome as well as any findings or deviations from what is indicated by the deposit list. If deposit verification services are part of the escrow agreement they should also be performed at this time. A more technologically advanced escrow service provider should have a web-based portal where both the software provider as well as software purchaser can view the state of progress of the initial deposit, inspection and the inspection results. Moreover, this portal will provide on-going visibility to the software purchaser of any future deposits as they are made. Lastly, if the release conditions are met the software will be made available to the software purchaser, either through the web portal or via a courier drop of physical materials that will be prepared by the escrow agent.

2.6 Standard Inspection

Software escrow failures can often be attributed to the lack of inspection of the components that are placed within the deposit as per the deposit list. A matter can sometimes be attributed as the licensee's fault, as they may not want to be burdened with the incremental costs of higher levels of inspection as described below. A significant number of the vendors that provide commercial software escrow services offer inspection with varying levels of validation (Andresen & Salyers, 2008) or review of the deposit materials, with associated

incremental costs. A few examples of these are available and vary from a standard deposit through to highly technical reviews.

Readers that are familiar with disaster recovery management will recognise the five phases of testing, as defined in (Korelc & Tittel, 2008). This includes checklist testing, a structured walk-through testing, simulation testing, parallel testing, and full interruption testing. Likewise, a five level verification model is proposed for software escrow by EscrowTech.com (2011), with the following five levels of verification:

Level 1 - File Listing Verification Report
Level 2 - Technical Verification - Deposit Analysis
Level 3 - Technical Verification - Build and Compile
Level 4 - Technical Verification - Binary Comparison
Level 5 - Technical Verification - Test Plan

At the initial level (Level 1) a visual inspection is done of the media contents and compared with the escrow deposit list as provided by the depositor. Further review may include an anti-virus scan as well as a cryptographical comparison of the deposit materials through the use of a cryptographically secure hashing algorithm such as SHA1 or SHA256, with the hash sets provided in the deposit inventory.[7] A visual review will also be done for the presence of the build instructions. This remains the basic and standard software escrow type, and the one that fails frequently, as very little is done to ensure the overall integrity of the deposit content.

At Level 2, a more thorough review is done of the build instructions as well as the deposited material. An attempt will be made to "identify the hardware, operating system, programming languages, third party software, and library dependencies" (EscrowTech, 2013). The incremental cost per year is nearly double that of Level 1, and this can be attributed to the additional effort that will be made with the individual deposits. Moreover, the assurance level provided to the licensee is much higher, as confidence levels in a successful recovery from the deposit material can be higher.

With Level 3, and at a three times incremental cost of Level 2, the provider will, with the input from the licensor, compile the source code for review. The items that the licensor must provide includes the programming documentation, build and compile instructions, and any other resources or tools needed to enable or facilitate software compilation for the technical verification. The licensee needs to provide comprehensive instructions on how to facilitate the complete process as well as any platform and externally licensed and sourced components to complete the successful compile of the source code.

On Level 4 the final outputs of a build process will be compared to what is in the possession of the client, with Level 5 offering to deploy and test the software onto a similar system to that used in a customer production deployment. Similar levels of verification are offered by local software escrow provider Escrow Europe

[7] These hash sets or checksums are commonly used to check data integrity.

(Stekhoven, 2013) as well as the EU based NCC, both operating in South Africa. The smaller local escrow providers as well as legal firms may not be able to provide these levels of inspection as the incremental infrastructure and technical demands may be beyond their capabilities.

The value proposition that is provided by software escrow services vary, but it is clear from statements in various portals and blogs (Smith, 2012), (National Software Escrow Inc., 2013) as well as the on-going marketing efforts from these groups that they are gravitating to higher value service offerings to ensure that the risks associated with incomplete or corrupted deposits are limited.

2.6.1 Release Conditions

When considering the options for releasing any software from escrow it will be in the interest of the licensor to have these limited to only the very essential triggers. On negotiating, the licensee will ensure at a minimum (Andresen & Salyers, 2008), (Iron Mountain, 2012a), (Motta, 2012) that the following conditions are contractually included:

- *a decision by the licensor or a purchaser of the licensor to discontinue support of a version of the software that the licensee is using*
- *a material failure of the licensor to meet its support obligations*
- *if the licensor is the debtor in a bankruptcy, is insolvent, or makes an assignment for the benefit of creditors.*

These release triggers was previously mentioned in Section 2.6.1. These conditions provide for release when, for example, a licensor is acquired and the new owner does not want to maintain the software (Type A), when the licensor fails to provide on-going support as was defined in the licensing agreement (Type B), or lastly when financial distress causes the licensor to default on on-going commitments (Type C).

According to Stekhoven (Dorrington, 2007), Types B and C are no longer as important as the chance of a developer being acquired. When this happens, the new owner organisation may only focus on some of the software that they acquired. This represents a much more realistic threat against which to protect. Moreover, Monnet (2011) suggests that multiple release conditions should be the standard, as for example a wealthy licensor (depositor) may choose to stop the support of an application, as the cost of maintaining software through the employment of expensive staff may make it less profitable. Moreover, just listing bankruptcy as a release condition will not suffice in terms of reducing risk for the consumer.

In 2010 Boruvka and Covello presented the following statistics related to releases experienced by Iron Mountain (Covello & Boruvka, 2010). An addition of “Type as per classification” to the original data published has been made, as shown in Table 1.

Table 1 - Table of Release Conditions (Covello & Boruvka, 2010)

Reason for release	Percentage of total	Type as per classification
Loss of support	30%	A
Cease Business Operations	22%	A
Insolvency / Bankruptcy	20%	C
Depositor's Request	9%	n/a
Demand Release	6%	B
Payment	2%	C
Court Order	1%	B
Breach Obligation/ Merger/ Transfer or Assets	<1%	A

The above items can be mapped back into the three categories defined above leaving us with a very limited platform for consideration as to what can and will be classed as release conditions. Typically the request for release by the depositor is a willing act and would in the majority of cases be preceded by one of the three conditions mentioned.

Each release, as per the conditions of the agreement, needs to be duly considered by the escrow agent in review with the licensee, as any release request made prior to asserting that the actual conditions have been met could be legally contested. Moreover, the use or right to use the software after a release needs to be defined in the licensing terms.

2.6.2 Release Procedure

Daniels has also noted (2007) that the same three release conditions as stated above are sometimes incorrectly interpreted within agreements that seek to describe the escrow conditions. Daniels also asks a pertinent question: "What is the ultimate objective of the parties?"

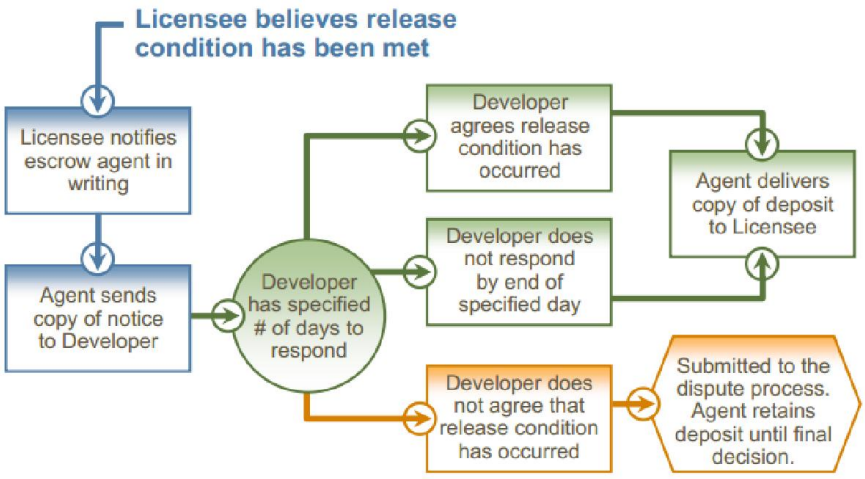


Figure 4 - Release Process Workflow (Covello & Boruvka, 2010)

The specific measurables and defined actions that could result in a release condition, as well as the interpretation guiding the process needs to be clearly defined; it is not enough to rely on the interpretation of a third party who may be called on to adjudicate such a matter.

The workflow in figure 4 is a typical depiction of how the decision making process can be followed should a release condition be triggered. However, it does not make provision for the actual review of the supporting licensing, maintenance and escrow agreements. It is within these supporting artefacts that additional information can be gleaned as to the rights of the licensee, specifically if the licensee is no longer paying for maintenance of the software or is using an unsupported version.

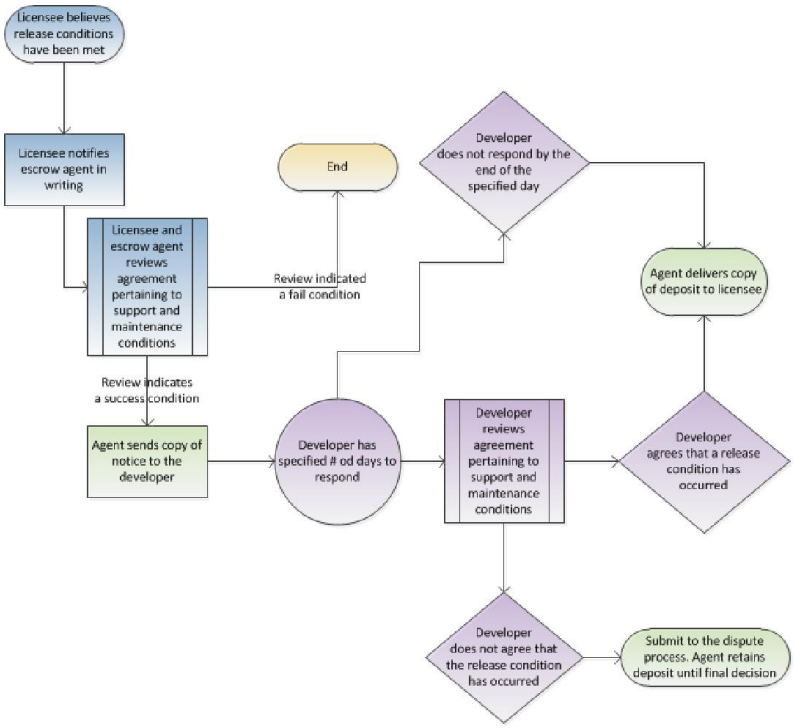


Figure 5 - Revised Release Process Workflow, based on the workflow by (Covello & Boruvka, 2010)

The revision of the release condition as defined by (Covello & Boruvka, 2010) is depicted in Figure 5 and introduces the review of licensing and maintenance agreements. This review should include the validity of the contractual agreements as well as the payment of related fees, if any, to provide the stipulated software maintenance and support.

It is therefore necessary to start considering what risks can be introduced, by the supporting functions such as finance and procurement, for example, should they neglect to pay for specific renewals or software maintenance because they were not informed by the contracting parties about such specific conditions within the software escrow terms. Moreover, the impact of other parties on the effectiveness is not measurable in the real world as it is not within the ultimate control of the licensee, where external influences such as available

budgets, vendor reviews and accounts departments behaving autonomously may have an uncontrollable impact on the continuity of such agreements.

A significant point of contention arises as per the case of Lubrizol Enterprise Inc. v. Richmond Metal Finishers Inc. (Morris, 1988) where in the United States of America the Fourth Circuit committed changes to the United States Bankruptcy Code Section 365 (Raymond, 2007). This act is also referred to in the USA as the Intellectual Property Licenses in Bankruptcy Act of 1988. Moreover, Section 365 (Raymond, 2007) provides for three additional conditions pertaining to the bankruptcy of the licensor in which the licensee could stand to lose access to the use of said software as well as possession thereof.

- Per section 365 (a) retaining the right to the software is defined
- Per section 365 (a) the right to retain the software, where a trustee may require the return of property to monetize it
- Per section 365(e)(1), clauses which use bankruptcy as a trigger and to release the software in an escrow deposit, are per se invalid.

This piece of legislation has many contentious issues, and it seems that even the well know case reflected on above was potentially overturned nearly three decades later as mentioned below, where they open the article by stating:

> *Chief Judge Frank Easterbrook of the Seventh Circuit recently created a split of authority regarding the rejection intellectual property licenses in bankruptcy by upholding a decision protecting a trademark licensee's ability to use a debtor licensor's trademark after the licensing agreement had been rejected*. (Riskin, 2012)

It is therefore advisable to ensure that any well-crafted software escrow agreement makes provision for the country of operation's legislative mandates so that such issues as pertaining to the US Bankruptcy act as well as changing views by appeal court systems, as highlighted above, do not creep in. Lastly, confidentiality (Banisar, 1996) as to the reasons that triggered the release needs to be maintained by all parties involved. It was attention to confidentiality agreements that limited information flow from all of the escrow providers that were contacted for this research.

With the source code, supporting documentation and processes now in the hands of the licensor, a usability test of the release procedure can be performed. Although testing may have been part of the source code deposit phase, any additions and updates could have introduced new variables.

A compiling or usability failure will call for a scrapping of the system or extensive development on the source code to get it to acceptable functionality. These are both costly endeavours and something the new active escrow offering attempts to overcome through measures such as extensive inspection and extended validation options (as described in 2.6).

2.7 King III report

The King III Report (Engelbrecht, 2009) is focused on corporate governance and is a non-legislative approach that follows a number of principles and requires adherence to these principles via a 'comply or explain' approach. King III applies to all entities from public to private and non-profit. This is considered as non-legislative approach based on South African Company Law, but was not validated by any legal expert.

When considering risk management within the South African context the guidance from the King Report on Governance is invaluable, because it is the foundation of reporting guidelines for public companies in SA as per the Johannesburg Stock Exchange rules. This makes it an ideal set of guidelines for risk management and governance (Engelbrecht, 2009). Chapter 4 of the King III Report is dedicated to the management of risk. The following extracts from Chapter 4 imply that additional caution and guidance could be associated with software escrow:

1. Principle 4.2: The board should determine the levels of risk tolerance
2. Principle 4.6: The board should ensure that frameworks and methodologies are implemented to increase the probability of anticipating unpredictable risks

Moreover, a standards based method should be used to determine risk, potentially comparing it to a baseline, as seen in Principles 4.7 and 4.9:

3. Principle 4.7: The board should ensure that management considers and implements appropriate risk responses
4. Principle 4.9: The board should receive assurance regarding the effectiveness of the risk management process

Should a risk be discovered, the plans and methods to control such risks - be it through risk acceptance, avoidance, mitigation or transfer - should be formally documented and reported on, inclusive of on-going reviews to ensure that proactive steps are in place to review and reassess such risks for on-going monitoring. The following guidance can be found in Chapter 5 of King III:

5. Principle 5.2: IT should be aligned with the performance and sustainability objectives of the company

The implication of the statement in principle 5.2 is that any actual risks exposed within IT systems and processes should be treated seriously, and raised with the board of directors as the introduction of any additional risks can impact the operational objectives of the company. An example of this scenario would be one where the active support for a key software application fails as on-going maintenance was withdrawn. In this case alternative plans need to be implemented to mitigate the risk.

6. Principle 5.4: The board should monitor and evaluate significant IT investments and expenditure

Whatever controls are presented to mitigate these risks, they should always be well-justified expenses that can effectively reduce risk. Insurance is a typical example, where for a low contribution a significant value can be derived, in cases of demonstrable loss. Likewise, the cost of software escrow is reflected in the risk that it attempts to mitigate, and the pricing model typically has a significant value-to-risk proposition.

Other legislative and compliance mandates also make provision for such controls, and the following items related to Basel and Turnbull are noted, as the control frameworks and systems have some similarities.

2.8 Basel and Turnbull

Historically, banks extended loans based on investments, deposits and capital at hand. Through revised risk models and more enlightened approaches these capital reserves may become overextended, making the bank technically insolvent. Basel II, an international standard for banking regulators was created to provide guidelines on how much capital reserves banks need to protect the bank against financial and operational risks.

Basel II (Bank for International Settlements, 2004) is a set of banking regulations that regulate finance and banking internationally. Basel II states that "Losses arising from disruption of business or system failures" pertaining to Hardware, Software, Telecommunications and utility outage or disruption needs to be provided for in the Detailed Loss Event Type Classification. It is then assumed that Basel II desires an Application Configuration Management Database, where all actively used applications are listed with maintenance cycles defined and support staff and other contacts documented and indexed for future lookups and use.

In the UK based Revised Guidance for Directors on the Combined Code, also known as the Turnbull Report (The Financial Reporting Council, 2005) , it is stated in Principle C.2 of the Code that;

> "*The board should maintain a sound system of internal control to safeguard shareholders' investment and the company's assets". It subsequently states that "the review should cover all material controls, including financial, operational and compliance controls and risk management systems*" (The Financial Reporting Council, 2005).

Like Basel II, Turnbull is also concerned with what controls are in place to manage various business risks.

Other acts such as the USA-focused Sarbanes-Oxley Act, as well as the Health Insurance Portability and Accountability Act, could also be consulted but have been left out of this research as their relevance on South African based companies will be limited. International readers should take note of this omission.

It is then concluded that from a corporate governance perspective the directors owns the responsibility to manage risk and to ensure continuity in case of a disaster or other type of failure. They are accountable for the

risks that may be introduced by licensing software that does not transfer full ownership and control to the company.

2.9 The Software Escrow Agreements

A number of different types of escrow agreements can be negotiated. These differences will be addressed through the levels of inspection that were defined in Section 2.6. Based on this model, types of escrow agreements can be divided into three sections: basic, standard and secure escrow, as depicted below.

Moreover, the traditional type of escrow, now referred to as passive escrow (Hanse Escrow Management GmbH, 2013a) did not provide for any inspection or verification of what was provided in the package. The introduction of active escrow (Hanse Escrow Management GmbH, 2013a), with various levels of inspection and verification, has introduced a number of value added features to the software escrow realm, allowing licensees more scope for risk reduction within the specific licensing agreement, as per Table 2.

Table 2 - Types of Escrow Agreements (Assembled from Hanse Escrow Management GmbH (2013a)**)**

Types of Escrow Agreements	**Basic Escrow**	**Standard Escrow**	**Secure Escrow**
Source code and documentation verified for completeness against manifest	X	X	X
Documents and Media tested for working order and readability	X	X	X
Current versions and updates requested from vendor at stipulated intervals	X	X	X
The source code is compiled (translated into the executable program) using the manufacturer's systems, and tested for correct operation		X	X
The manufacturer's development environment is documented in detail		X	X
Source code and documentation stored at two independent locations	X	X	X
The software manufacturer's development processes, programmer guidelines and system environment are audited and documented in detail			X
Optional - Selected code excerpts are evaluated for methodology, programming clarity and reusability by a neutral expert			X

Table 2 provides an overview of the typical contractual agreements provided by one escrow agent. We can note that there are some variations on this theme between the major providers. Two sample software escrow agreements have been included in Appendix B and C as obtained from a local escrow agent Escrow Europe (Pty) Ltd.

2.9.1 Agreement types and fees

With two types of agreements available the licensee needs to ensure that the correct type of agreement is selected for the applicable use case. If a software provider is providing bespoke developed software to only one end user company, for example, or if the licensee is the only party insisting on a software escrow agreement, the Three Party Agreement can be utilised. In some cases, however, the software provider is developing the same software for multiple end user companies and has taken the view that by initiating a software escrow agreement he/she will provide a competitive advantage by indicating that his/her licensed software will also be underpinned with software escrow.

2.9.2 Three Party Agreement

The Three Party Agreement (see appendix B) is the standard agreement between the software provider and the software purchaser and includes, at the request of the purchaser, the software escrow agent of the purchaser's choice. It is therefore referred to as a three party agreement. No consideration is made for any additional licensees of the stipulated software and the consensus is that this type of agreement will apply where bespoke development is undertaken for one client and that the software will not be licensed to additional parties. The contractual obligations (Karlyn & Overly, 2012), terms of reference and release conditions are negotiated between the software provider and the software purchaser, and only once all parties are in agreement will the final contract be signed by all three parties. Payment for services in these one-to-one types of agreements are highly flexible and can be a financial burden to the licensee if the contract initiation was by the licensee, who may in such a case also have the option to select the escrow agent. Should the software licensing terms stipulate the initiation of a software escrow agreement, the choice of escrow agent and annual fee are most commonly the responsibility of the software provider. Greater care needs to be exercised by the licensee in this case as fee default by the software provider could place them at risk, as fee default with the escrow agreement can result in a termination of service by the escrow agent.

2.9.3 Multiple Licensees

The Two Party Frame, or Multi Beneficiary agreement (see appendix C), is negotiated between the software provider and the software escrow agent, initiated by the software provider (possibly to provide a form of competitive advantage or assurance to current and future licensees).

Each licensee will be acknowledged into an addendum which they are not party to. The licensee may not influence the existing agreement terms or release conditions. As the agreement was initiated by the software provider, they remains accountable for the contract initiation fees, annual contract fees and additional deposit fees, should the latter be included in the agreement.

2.9.4 Other Fees

A basic software escrow agreement will have a contract initiation fee (EscrowTech, 2013) paid to the escrow agent, used to create the initial agreements between the parties and to establish the deposit processes. An annual fee as well as renewal fees provide for the actual terms of inspection and will be based on the basic, standard or advanced type of agreement that will be settled on.

These agreement types will also allow for at least one full deposit per year and potentially one additional deposit, with defined fees payable for each of the further deposits made throughout the contract year. These may be defined as scheduled or unscheduled deposits and could attract different fee structures for each type. Readers should remain vigilant when considering too few deposits throughout the contracting term, but must remember that detailed inspection and verification services will be priced as premium services and will attract incremental fees.

Finally, the source code release fees will at all times be payable by the individual licensees and the escrow agent will demand these fees prior to initiation of his investigation to determine if the agreed release condition(s) have been met. This will also be stated in the agreement signed by both parties and will be declared to the licensee on request in the case of a multi beneficiary agreement.

2.10 Other Uses of Software Escrow

Conventional software escrow is not just limited to bespoke developed software where a relationship between the developer and licensee may be one to one or one to many. Other areas for review include items in the popular app stores of Apple and Google, as well as cloud based applications.

2.10.1 App Stores

Many thousands of useful applications and utilities are now available for popular cell phone and tablet computer-based platforms as can be seen on the Apple Store or Google Play. Some of these even spills over onto set-top boxes such as Apple TV and Android-based mini PC's, smart TV boxes and multimedia players.

Consumers have little concern with the on-going development and availability of these small bespoke applications. Key to this is the many to one relationship with the developers – a relationship that, in practical terms, is non-existent. What is more significant, however, is the actual cost of these applications, which varies from free to US$999.00 for specialist applications in the Apple Store (Curtis, 2013). Seen as commodity items, they are frequently replaced for other applications that offer better user experiences or additional features and functions. Average users are not therefore generally interested in either long-term access or the long-term availability of these applications.

Corporate demands for the use of personal computing devices have spawned a number of vendors that provide Bring Your Own Device (BYOD) and mobile device management features and functions, where private application stores are also offered so that users can access and update company-provided productivity and sales enablement applications specific to their industry verticals. It is within this context that business needs to also provide for standard software licensing and software escrow agreement terms. A failure to consider these smaller applications and productivity tools in a similar context as larger enterprise developments may influence productivity and profitability through replacement lifecycle and end user computing cost, such as training for new user applications.

2.10.2 Cloud Services

Cloud services will usually strive to be a low cost commodity (Jones, 2012) and aim to provide for standard options that will be offered to multiple end user companies, with some or potentially very little client specific customisation. Access will be through a remote client, probably a browser, implying that the software, supporting systems as well as end user specific data, are all within the domain and control of the hosting provider, not the software licensor.

Asking such a licensor to provide a software escrow agreement to one licensee would probably affect the licensing costs structures, although it would be a better practice to offer such a service as part of the overall licensing agreement and the initiation of a Multi Beneficiary agreement. To date no such instances were found. The initiation of such an escrow agreement will introduce a level of confidence to all the licensees and could become a significant marketing differentiator.

As a licensee, you can ask specific questions pertaining to the hosting agreements that are in place. These should include;

- Who is the hosting service provider?
- Are potentially multiple service providers involved in the hosting agreement, so as to ensure better availability of the application to geographically dispersed users?
- If not, what alternate arrangements are in place to fail over to another hosting provider?
- If the software licensor should fall on hard times and default on hosting fees or experience connectivity issues to the off-site applications, what plans are in place to limit such service disruptions?

In both (Stulman, 2008) and (Jansen & Van De Zande, 2010) the readers are pointed to the additional components within a Software as a Service (SaaS) environment that are not controllable variables as per traditional software licensing. Within the SaaS solution, the hosting providers are also part of the software escrow contractual agreement and it becomes prudent that the software licensor includes software escrow as part of his standard licensing terms. The opportunity to further provide an escrow service for the actual data associated with the Software as a Service application space allows us to focus on data escrow, or the availability and capability to externalise the data that is utilised by the cloud-based application.

Knowing who these hosting service providers are, with Amazon[8] and Rackspace[9] as two well-known providers, opens up the opportunity to engage with then directly as a licensee, should that become necessary (in case of a material failure on the side of the software provider, for example). Clearly, a licensing agreement will have to be structured in such way as to provide for this, but more importantly, items such as super user and other administrative accounts and passwords, encryption keys and access to account unlocking functions will also have to be addressed, should you choose to follow this path.

[8] http://www.amazon.com
[9] http://www.rackspace.com

A case in point is the closure announcement from Nirvanix[10] as was posted late in September 2013, and a follow up stating:

> "*On October 1, 2013, Nirvanix voluntarily sought Chapter 11 bankruptcy protection in order to pursue all alternatives to maximize value for its creditors while continuing its efforts to provide the best possible transition for customers*" (Nirvanix, 2013).

The time window provided to clients of less than one month to extract data was very short, and has been criticized by the media (Cheredar, 2013). Moreover, in his blog John Sloan (Sloan, 2013) suggests the following as a takeaway. "A shift to the cloud (for data) needs a solid investigation and due diligence as well as structured plans to address associated risks." When contracting with any cloud service provider an organisation needs to carefully plan how they will exit the service, especially if reliance on the service provider for connectivity and bandwidth availability comes into play when they need to proceed with bulk data migration.

2.10.3 Other Use of Escrow

The internet economy is not just one of legitimate commerce and trade (Kshetri, 2010). Fraudsters have built up an economy where free and open trade of credit cards, stolen identities and many other commodity items happens on a daily basis. To support such an economy without the trust that exists in the brick and mortar banking systems, a different way to process these trades was required.

Historically, fraudsters would use, for example, Trojans and phishing to gather banking login data and credit card details, inclusive of CVV numbers[11] and expiry dates from their targets. These fraudsters are the primary technical specialists and are often referred to as harvesters, as they utilise technical infrastructure to collect their bounty. Once a harvesting fraudster has a reasonable number of 'cards' (a term used in their communication forums), he/she would seek to monetise these cards (Rivner, 2009), as the harvesting fraudster does not specialise in the operations where the cash-out phase commences.

Cash out fraudsters (VISA, 2013), (Kayle, 2003) are not as technology-focused as the harvesting fraudsters are, and operate human infrastructures and human resource management tools to manage and measure the effectiveness of mules performing specific tasks. They also use task and activity tracking software to allocate various tasks to the mules. These mules then perform cash withdrawals with fake cards, make bulk purchases with stolen credit cards and break up large bundles of high value commodities for onward shipping to receiving parties who will further monetise these goods for the profit of the cash out fraudster.

Moving these stolen cards from the harvesting fraudster (Kayle, 2003) to the cash out parties can be risky, as a non-verified fraudster can offer to pay for the goods with another stolen card that may already be blocked or

[10] http://www.en.wikipedia.org/wiki/Nirvanix - link no longer available for http://www.nirvanix.com

[11] The CVV Number ("Card Verification Value") on your credit card or debit card is a 3 digit number on VISA®, MasterCard® and Discover® branded credit and debit cards. On your American Express® branded credit or debit card it is a 4 digit numeric code (http://www.cvvnumber.com/).

see its transactions reversed. Therefore, a new underground economy has risen that takes care of its own through currencies like Bitcoin[12].

A further step often taken is to only accept transactions negotiated via a trusted third party, who acts as an escrow partner, keeping payment for 'cards' until the recipient has reviewed them and verified the quality, quantities as well as validity of the accounts. Therefore, if a harvesting fraudster sells 100 credit cards that each have an account balance of US$200 or more, the cash out fraudster will need to determine from a sample that these values are accurate. Figure 6 represents an endorsement from Rockcandie who makes use of iSellPizza, a harvesting fraudster.

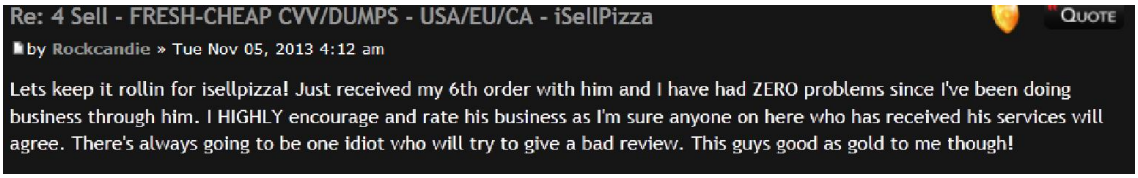


Figure 6 - Fraudster endorsement [13]

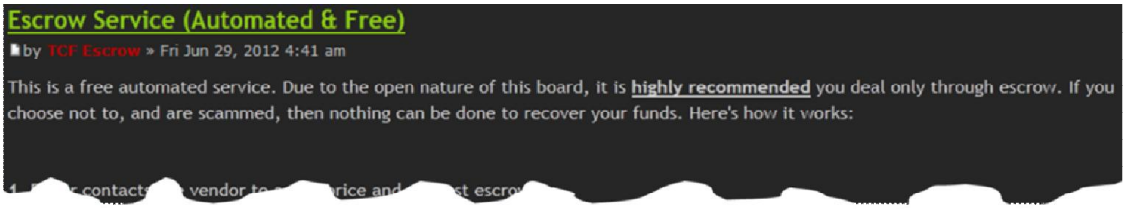


Figure 7 - Verto Escrow Service

Figure 7 shows Verto, a known escrow service provider that advertises his services on the Darknet. His text highlights the fact that a buyer can be scammed and goes on to describe his services as follows:

> *Escrow Service (Automated & Free)*
>
> *Post by TCF Escrow » Fri Jun 29, 2012 4:41 am*
>
> *This is a free automated service. Due to the open nature of this board, it is highly recommended you deal only through escrow. If you choose not to, and are scammed, then nothing can be done to recover your funds. Here's how it works:*
>
> *1. Buyer contacts the vendor to agree price and request escrow.*
>
> *2. Vendor sets up escrow (Create option in UCP). For most digital products, immediate delivery option should be selected. Default maximum inspection time for digital products is 2 days. For physical products, select the expected shipping time. Bitcoin transaction fees (normally only a few cents) can be shared, or covered by buyer/vendor.*

[12] http://www.bitcoin.org

[13] This and the text copied from the post by TFC Escrow as well as additional screenshots all requires access to Tor, also known as The Onion Router. A method to browse the Internet anonymously. No link is provided as access to this portal in the Darknet is restricted and the link is transient.

3. Buyer will then receive a Private Message (PM) from TCF Escrow with instructions. You can check the status of your escrow transaction at any time in your User Control Panel (UCP).

4. Check the agreement carefully and send funds to the bitcoin address provided. If you are not happy with the item description/delivery time/price, you should contact the vendor and ask them to set up a new escrow transaction with the correct details. Funds will appear in escrow after 2 confirmations.

5. Vendor is notified by TCF Escrow to send the product.

6. Finalize or dispute purchase within the inspection time period. If you fail to do so, funds will automatically be released to the vendor after the inspection time has expired. If you require more time, you can file a dispute citing the reason.

7. Funds are released to vendor after finalization, or after inspection time has expired.

In the event of a dispute, funds will be held in escrow until an Administrator has looked into it and a resolution is reached.

Accepted Currency: Bitcoins [BTC]

If you have any questions or problems with this service, you can contact any Administrator.

Disputes do arise and they get the required attention from the active participants as well! Some of these transactions need verification of the goods presented by seller, especially if the seller is new to the market and without an established reputation; not that reputation alone is the key ingredient to the creation of trust in this market place.

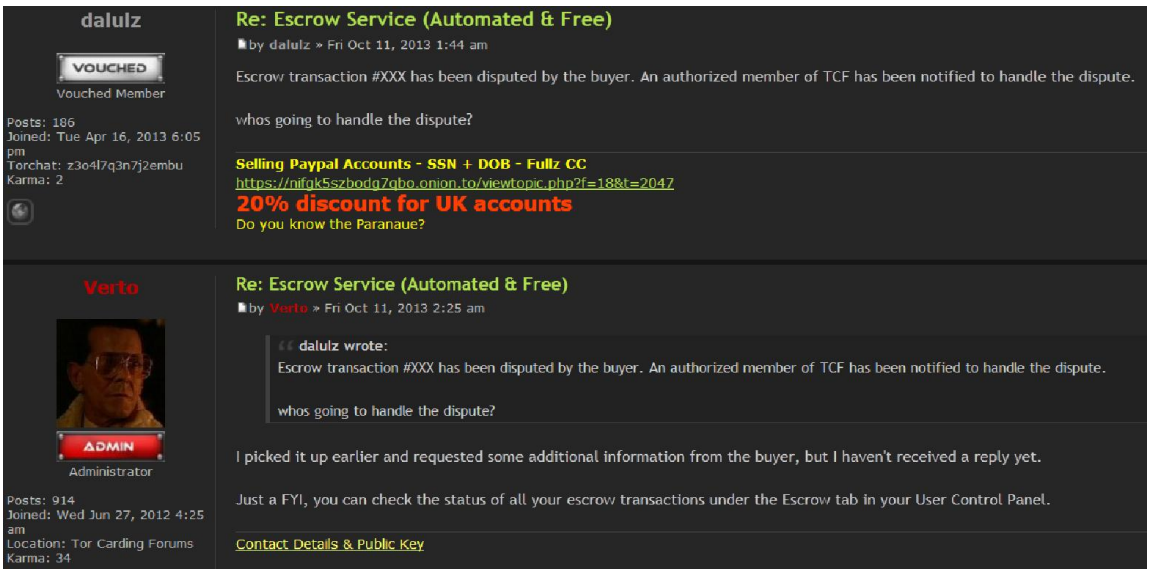


Figure 8 - Escrow Dispute

Indeed, as per Figure 8, some challenges do exist in the underground economy. This is not the type of commerce where all transactions are performed and delivered in a professional way as we are accustomed to in conventional businesses. It is not unusual to see these fraudsters also taking the opportunity to abuse their own potential customers. Therefore, a trusted escrow service as described above, where the buyer can park his

e

payment until after inspection, or with a release condition to release after a number of days have passed, irrespective of inspection, is a safer option for these underground traders. In cases where the agreed escrow time period lapses, funds are automatically released based on the release condition or time. However, if inspection takes place and the goods on offer are found to be inferior or significantly below an agreed baseline for, say gold credit cards with an available balance of over US$1000.00 and platinum cards with available balances exceeding US$5000.00, a dispute can be raised and will be handled by the escrow provider for settlement as he/she deems fit.

2.11 Summary

The purpose of this section is to provide sufficient background on the various area of software escrow, the common processes involved with committing to an escrow agreement as well as how such an agreement will be structured for different client needs (single end user or multiple end users).

From a governance perspective the guidance from the King III report is very clear that South African companies need to comply or explain why they do not comply to reduce the risks associated with bespoke software development.

A final review of other areas where software escrow can and will assist the prudent and risk sensitive compliance officer is also included as the reliance of cloud based services will only increase over the coming years.

This background also serves as good foundation information to make further critical assessments of the cases studies in the following section.

Chapter 3 - Software Escrow Case Studies

The third objective of this research is to make a case for software escrow; Indeed, a prime motivation for initiating a software escrow agreement today is an examination of past successes and failures, and an analysis of where software escrow could or has made a difference.

As mentioned, academic research on this topic is limited. It is therefore valuable to include reference to material outside of the traditional academic scope, such as cases where software escrow was called to action, or those where available escrow would have offered a more elegant way forward. Some of these are success stories, some not; in one the pure stubbornness of the successor of a bankruptcy has prevented the escrow agreement from triggering the release through substantive further legal actions.

3.1 Telecom Limited New Zealand and Aldous Limited

In 2003 Telecom Limited New Zealand (Plaintiff) and Aldous Limited (in receivership) entered into a software licensing agreement (Peters & High Court of New Zealand, 2012), with Aldous modifying the software over the next four years on a regular basis. It was further estimated that Telecom spent over NZ$ 7,000,000.00 on these updates, notwithstanding the initial perchance. This particular software was a key element for the running of the Telecom CDMA mobile telecommunications network.

An escrow agreement was signed in 2007 and it was agreed that the source code would be placed in escrow within ten days of each update.

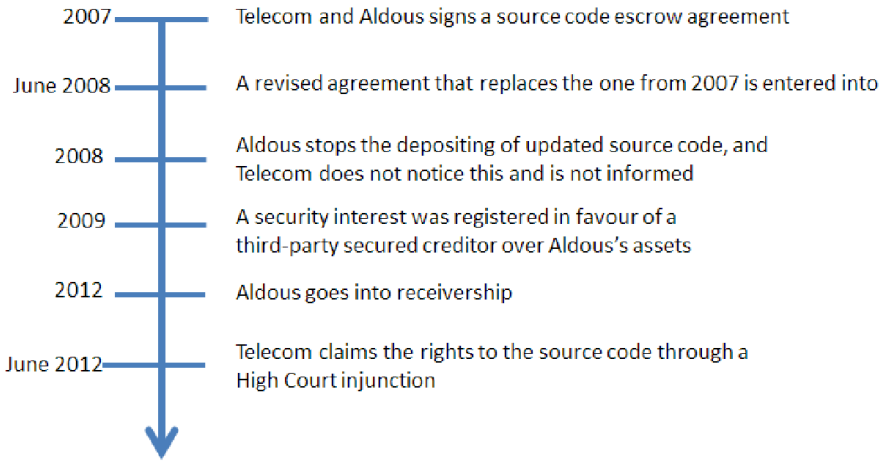


Figure 9 - Telecom and Aldous timeline

Burgess (2012) notes a number of facts that can be depicted into the timeline as shown in Figure 9. He notes the 2009 activity as a complication, but should have stated that the escrow agreement was probably not sufficiently scrutinised by the Telecom legal team. Burgess then proceeds to make a number of points,

including that Telecom should have registered a secuity interest in the intellectual property; he preceeds his claim, however, by stating that "*A receiver is not necessarily bound to observe an escrow agreement or other 'onerous' contracts, and the unsecured customer or licensee may lose out*". This clouds the view that an escrow agreement is legally enforceable should the licensor go into financial distress.

In the judgment as it was delivered by Justice Peters on 28 June 2012 (Peters & High Court of New Zealand, 2012) it is clearly stated that Aldous was in breach of the agreement in that it did not continue to place software, as it was updated, in escrow with the escrow agent, and that if they did deliver on the agreement in a proper way, that Telecom would have had access to the source code.

Other key points to take note of in this case are:

- The software was developed exclusively for Telecom, who had spent over NZ$7 000 000 in the customising of the code
- The operation and maintenance of the Telecom CDMA mobile telecommunications network was completely dependent on the software.

Considering these two points, both in isolation as well as in the context of the eventual outcome, makes clear the fact that any party with a high dependency on bespoke developed software must consider the risks they will face if software deposits are not made regularly; in the case of Telecom, they should have been more vigilant when notice of the on-going deposits ceased after the June 2008 agreement.

3.2 South African fund manager 'Manco' and SoftwareX

A South African fund manager Manco with over R200 billion of assists under its control, sought to implement software called SoftwareX (Stekhoven, 2012) from an intellectual property provider that, at the time, was under financial distress. It was also known at that time that SoftwareX were open for negotiations to be acquired.

The value proposition of the SoftwareX solution was attractive and the solution was successfully implemented. At the same time, the developer of SoftwareX was acquired. Within nine months the new controlling company, a listed entity, lost interest and stopped maintenance and support, whereupon Manco called on the new owners, the listed company, insisting that they were in breach. The request was not agreed to and Manco, who fortunately insisted on an escrow agreement with the initial party with whom they negotiated, called on the software escrow agent. On review of the matter the escrow agent released the source code as well as all supporting documentation to Manco. A graceful transition to new software was achieved, and Manco successfully negotiated its internal compliance requirements, which valued the risk and rewards equation of a software escrow agreement.

3.3 South African Short Term Insurance Lessons

The South African short-term insurer Natsure, established in 1968, contracted with Escrow Europe (Stekhoven, 2008) to protect its investments in key software technologies, with the escrow provider agreeing to release source code on certain conditions. An important moment in the history of short term insurance was the failure between IT outsource provider Dexdata and Prestasi ("Escrow Leads Business Continuity Charge," 2010). Here Prestasi lost access to its entire client database and had to resort to manual processing systems. When a court order was passed to return the tapes with the client database to Prestasi it was found that the tapes did not contain meaningful data!

A valuable point to take forward is therefore that the validation of any escrowed intellectual property, as previously highlighted in Section 2.6 regarding inspection, will be key to the success of such agreements once they are called upon to deliver.

3.4 Not Verified and Tested

A small automotive supplier uses bespoke software for its production facilities. Teams in the purchasing and logistics, as well as the management information system, rely on data from this application to perform specific tasks. When a request for changes to the program was made to the supplier it was found that they had filed for bankruptcy (Hanse Escrow Management GmbH, 2013b). Although the source code was filed with a source code escrow provider, it was found that the cost to make the required changes and recompile the code was much more than anticipated.

In this case, although the escrow for a key application was implemented, it was not part of the standard deposit practice to verify and test the code, nor was the deposit material properly maintained and updated. Therefore the actual cost of making the changes to the now-legacy systems needs to be compared to the entire replacement of the solution with a new one that includes the required escrow and validation of all deposits.

3.5 Trust but verify

An Australian multi-national retailer, who insisted on an escrow agreement with the service provider, needed changes made to its point of sale and stock replenishment system (Harbinger Escrow, 2013). It was decided to request that an independent third party do a review of the escrowed material, inclusive of recovering and recompiling the code.

To achieve this, a special quarantined room was created where two dedicated staff members were allowed to perform their duties of the recovery, but all communications with outside parties was arbitrated, and audited.

In total, eight communication attempts were needed to get to the end goal of compiling the code. The lack of documentation and clear instructions resulted in a revised document that provided a 140 step software recovery process. In the end, these were also tested and it took two dedicated engineers more than four days to come up with a successful build. Therefore, although the code in escrow was valid, the methods associated with the final compile and build was missing. A complete deposit list, inclusive of build instructions, type and versions of the complier needed to be validated and tested.

3.6 No conclusive outcome to date

The Australian case of Vemics, Inc. v. Radvision, Ltd is an instance where the requested release of source code from the escrow has been in litigation for so long that the applicant will have suffered significant losses through the on-going litigation and loss of access to support of the software (Anonymous, 2011), (Jansen & Van De Zande, 2010).

Vemics entered into a licensing agreement with First Virtual Communications (FVC) and agreed to a software escrow with Iron Mountain. In 2005, First Virtual Communications filed for bankruptcy and Radvision acquired its interests. Ten months later, in 2006, Venics requested the release of the source code, citing the bankruptcy as the release event. Radvision refused, and defended its position, stating that the First Virtual Communications bankruptcy was not a valid release trigger, starting a legal dispute that was still unresolved as of April 2011.

To quote ("General form for Registration of Securities of Small Bussiness Issuers," 2007):

> *We are subject to a legal proceeding (*Vemics, Inc. v. Radvision, Ltd.*, 07 cv 00035 (CLB) (LMS), United States District Court for the Southern District of New York) described in "Legal Proceedings" below that involves our intellectual property. Further, we may be subject to claims in the ordinary course of our business, including claims of alleged infringement of the trademarks, copyrights and other intellectual property rights of third parties by us and our licensees.*

That is also the last traceable reference available on the topic by Vemics, who is now known as iMedicor,Inc.(VMCI)[14] Present day publications regarding the conclusion of this case are not available and it is assumed that the case was eventually dropped, as legal fees and support costs for the software would not be advantageous for iMedicor.

3.7 Failure to make bookings

When the Radisson Hotels group hired a small developer to provide maintenance for its reservations system (Cheng & Helms, 2008), they stipulated that this critical system would require the developer to place the source code into escrow to ensure they would then be able to support it internally if the provider failed to

[14] http://www.imedicor.com

maintain it. When the developer stopped trading and the release was triggered, initial requests to approve the release were not met favourably, until a threat of a personal fraud suit was issued (Denson, 1998). On inspection, the code was so inadequate that it could not even perform basic hotel functions, such as booking a guest. Moreover, no documentation was provided with the deposit and it was missing several key components.

A clear lesson here is that ongoing inspection of the deposits will be needed and that deposit lists as well as supporting documents of how the executable code was created needs to be provided for in the agreement terms.

3.8 Review of Case Studies

From the case studies above it can be deduced that the act of implementing a software escrow agreement on its own is not sufficient protection for the licensee. Too many external influences can still impact the outcome of any claims. The maintenance and updating of the items within the escrow is of prime importance, but so too are the supporting documentation, testing of the source code, compilers and any other supporting processes. It is also clear that leaving the actual on-going deposit activities up to the software provider and expecting the escrow agent to update the licensee on a timely basis is not sufficient, as was seen in the Aldous and Telecom case. Moreover, newer escrow services are now fully web enabled and can provide for deposits in full digital format, with alerts and notifications extended to all participants.

Standard best practice as per Section 2.6, therefore, guides us to these five steps:

1. File Listing Verification
2. Deposit Analysis
3. Build and Compile
4. Binary Comparison and
5. A Test Plan.

As highlighted in Section 2.6, standard inspection of the escrowed items is the best practice and remains the only way to ensure that risks associated with the items within the actual software escrow deposits(as per Section 2.4) are a true reflection of what is required to convert the source code to the complied application. Standard inspection reduces these complexities to manageable levels more aligned with the cost value proposition that the software escrow services offers.

In Table 3 we can see how each of these use cases also maps back to the release conditions as discussed in 2.6.1.

Table 3 - Use Cases Mapped to the Release Conditions

Standard Release Conditions	Case Studies that applies
A decision by the licensor or a purchaser of the licensor to discontinue support of a version of the software that the licensee is using.	3.5 Trust but verify – implied only
A material failure of the licensor to meet its support obligations	3.1 Telecom Limited New Zeeland and Aldous Limited – partial
The licensor is the debtor in a bankruptcy or is insolvent.	3.1 Telecom Limited New Zeeland and Aldous Limited 3.4 Not Verified and Tested 3.7 Failure to make bookings
The licensor makes an assignment for the benefit of creditors.	3.2 South African fund manager ('Manco') and SoftwareX 3.6 No conclusive outcome to date

3.9 Summary

The total number of reference items for these case studies was limited: confidentiality agreements as well as the low level of actual releases contributed to this. Moreover, no software provider will willingly participate in such a cases study as it provides only negative publicity, whereas escrow providers can use these examples to motivate and support their business.

The above analysis provided a view into this domain, inclusive of the common software escrow concepts, namely:

1. What should make up the Escrow Deposit
2. How a good deposit workflow would function
3. What inspection of deposit material should be made
4. What makes up the standard release conditions as well as the release procedure?

A review was then given of the types of software escrow agreements - a legal area as these multi party agreements tend to be written to protect the interests of multiple parties and need to limit the liabilities of the escrow provider. It was also shown that different types of agreements also offer different fee structures, with the incremental inspection rigour adding to cumulative costs. It was noted that escrow can be extended into other areas, including those where software as a service will become the norm, although these agreement types are only now gaining some level of market maturity and commercial acceptance. The darker use of escrow was highlighted and it is noteworthy that this area mimics the way that property conveyance leverages escrow through a transferring attorneys trust account.

At this point readers should have a good understanding of software escrow and its use, limitations and practical applications. In the following chapter some case studies will be reviewed where the uses, and the failures, of escrow will be expanded on, with the aim to highlight specific risks and failures to heed certain cautions that can lead to unnecessary business risk.

It is important, before moving forward, to consider the areas of research that have influenced the specific questions that will be asked of our research participants. These relate to the size and stability of the service providers, as well as what can happen if a service provider is acquired by a larger entity with limited interest in maintaining some or all of the acquired software solutions (as was illustrated above with three such examples).

It is now clear that the area of software escrow is not well documented, and from the evidence of interactions with various parties it also can be seen that it is a domain that is not well understood within business. In the next chapter we will explore the survey and some related questions that will lead to the analysis phase of this thesis.

Chapter 4 – Survey Approach and Methodology

The initial approach to gathering data for this research was to reach out to various entities that fell into one of three categories: software providers, licensees and escrow agents. A detailed questionnaire pertaining to the use of software escrow, targeting software providers and licensees, was disposed of early in the research cycle, as it was established that in such cases confidentiality agreements would prevent the respondents from providing real and meaningful responses to the set questions, even if it was undertaken through personal interviews with anonymous responses.

A different approach was therefore developed, one that would provide the respondents with a set number of selectable responses to research items that would then return a risk-based result, as well as a baseline for measurement and comparison. Moreover, it was decided not to attempt to solicit any input from the known escrow agents, as their client confidentiality agreements would severely limit how they participate in such research. This question set was forwarded with an explanatory email to 65 recipients, of whom 13 completed responses which were then utilised in this thesis. A low 20% return is still deemed acceptable, as those that took the time to participate showed sufficient depth of knowledge and insight into the subject domain to consider their views as valuable. This impact of this is discussed in the following chapter in Section 5.2.2.

This chapter deals with elements of the research survey, the requirements for software escrow and the research items. It also outlines the research items as they were presented to the research group. Finally, this chapter deals with how the research items were scored, with the results thereof provided along with a summary of the survey.

4.1 Elements of the Research

A number of research items were defined and they were divided into the following six categories (with supporting text to guide the respondents should they require additional clarity when responding to the survey):

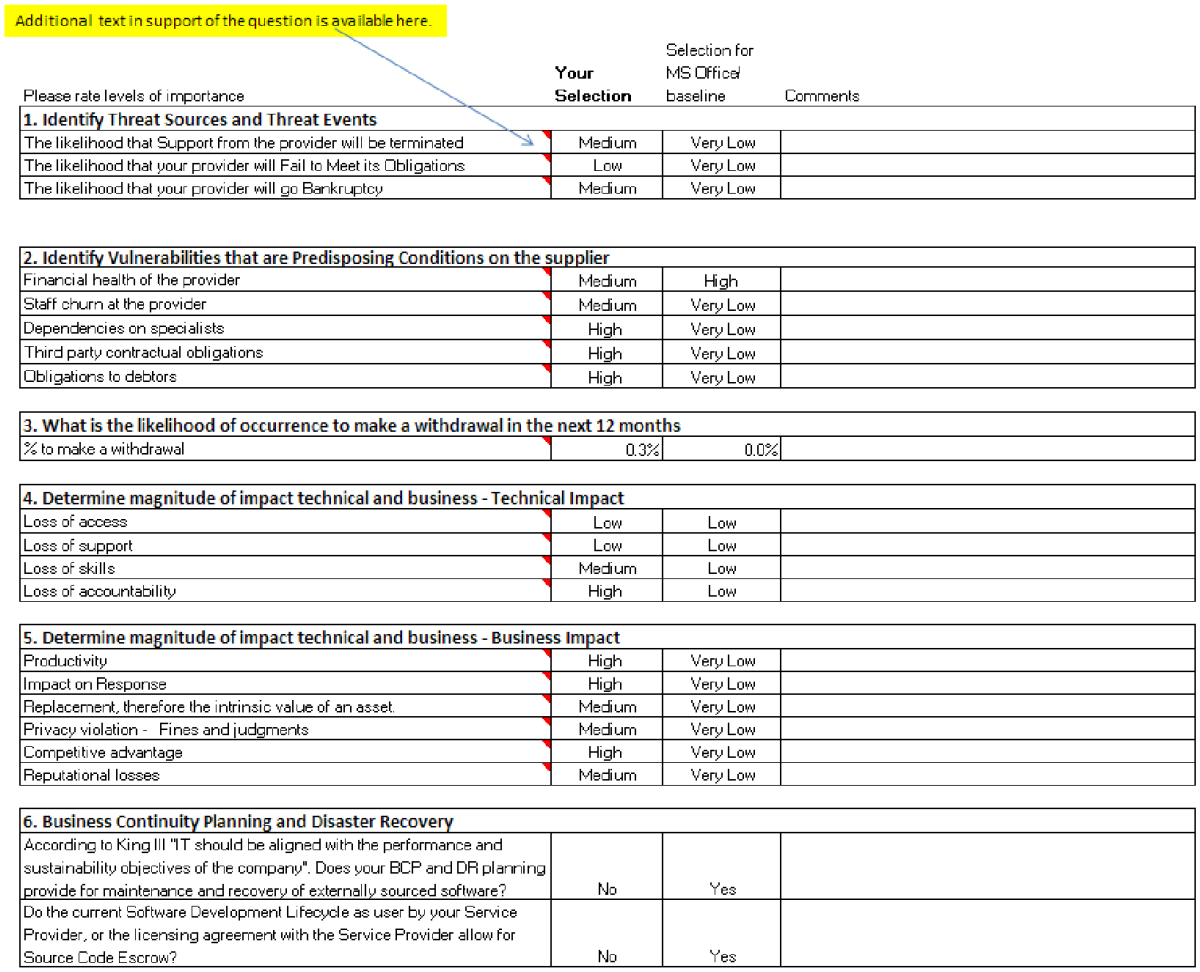
1. Identify Threat Sources and Threat Events
2. Identify Vulnerabilities that are Predisposing Conditions on the Supplier
3. What is the Likelihood of Occurrence to make a Withdrawal in the next 12 Months
4. Determine Magnitude of Technical Impact
5. Determine Magnitude of Business Impact
6. Business Continuity Planning and Disaster Recovery

On completion of the research items and a supporting baseline, a detailed matrix was presented to the participants, providing context and a guiding conclusion. This artefact served as supporting evidence for our third objective: to provide motivation for the use of software escrow agreements.

For comparison purposes, the Microsoft Office Suite was selected as a baseline application. As a common application within enterprises and a very well-known organisation, the respondents would be familiar with Microsoft as an organisation as well as be in possession of a moderate working knowledge of the Office Suite of products. It was assumed, therefore that they would be able to provide qualitative assessment of the Microsoft offering, as it compared to the in-house product under consideration, without confidentiality issues and related bias.

The six main areas of the survey were based on knowledge and insights gained from prior research, with the first two areas influenced by an examination of the actual disruptive events that can result in software escrow code releases. Finally, the third area is based on the low frequency of actual releases and the last three areas on the impact that the lack of software escrow can have on a business as a disruptive force.

Figure 10 is a representation of the spread sheet with the research items as presented to the research participants.



Additional text in support of the question is available here.

Please rate levels of importance	Your Selection	Selection for MS Office baseline	Comments
1. Identify Threat Sources and Threat Events			
The likelihood that Support from the provider will be terminated	Medium	Very Low	
The likelihood that your provider will Fail to Meet its Obligations	Low	Very Low	
The likelihood that your provider will go Bankruptcy	Medium	Very Low	
2. Identify Vulnerabilities that are Predisposing Conditions on the supplier			
Financial health of the provider	Medium	High	
Staff churn at the provider	Medium	Very Low	
Dependencies on specialists	High	Very Low	
Third party contractual obligations	High	Very Low	
Obligations to debtors	High	Very Low	
3. What is the likelihood of occurrence to make a withdrawal in the next 12 months			
% to make a withdrawal	0.3%	0.0%	
4. Determine magnitude of impact technical and business - Technical Impact			
Loss of access	Low	Low	
Loss of support	Low	Low	
Loss of skills	Medium	Low	
Loss of accountability	High	Low	
5. Determine magnitude of impact technical and business - Business Impact			
Productivity	High	Very Low	
Impact on Response	High	Very Low	
Replacement, therefore the intrinsic value of an asset.	Medium	Very Low	
Privacy violation - Fines and judgments	Medium	Very Low	
Competitive advantage	High	Very Low	
Reputational losses	Medium	Very Low	
6. Business Continuity Planning and Disaster Recovery			
According to King III "IT should be aligned with the performance and sustainability objectives of the company". Does your BCP and DR planning provide for maintenance and recovery of externally sourced software?	No	Yes	
Do the current Software Development Lifecycle as user by your Service Provider, or the licensing agreement with the Service Provider allow for Source Code Escrow?	No	Yes	

Figure 10 - Research Items and Response Sheet

As mentioned, the set of research items as per each heading were provided to a sample group with fixed responses available for the selected software as well as a baseline for comparison. Once the completed surveys were received a scoring was attached and an output was provided by return email.

4.2 Requirements for Software Escrow

Considerations for determining the risk are derived from the six key question areas; the business itself should determine whether use of a given software application is so pervasive, or whether the software will have such a significant impact on its financial success that they cannot do without it (for example, a key application with a large number of users, or software that contributes significantly to the bottom line).

By highlighting the various successes and failures in the case studies above, it can be shown that should conditions allow, a software escrow agreement with the licensor is the prudent way to proceed.

4.3 Research Items and Results

The organisation's IT Audit team, together with the internal audit (ISACA, 2013), legal department and procurement departments will normally be the operationally-focused teams involved in determining what risks might be introduced by the sourcing and procurement of bespoke software solutions. The aim of this research is to build a model that can provide significant insight into the process of selecting a new software vendor and a means to evaluate the new software vendor against a baseline that will indicate the extent of the risk undertaken with that vendor. Moreover, with a detailed overview of the various components, terms and workings of software escrow agreements, inspection and release procedures described, it is possible to provide a consolidated overview for any person intent in gaining significant insight into the operation of software escrow. This process also aligns with our third objective: to motivate for the signing of a software escrow agreement.

The following tables provide an overview of the questions, supporting information and scoring.

4.3.1 Identify Threat Sources and Threat Events

The purpose of the three research items in Table 4 was to determine how comfortable the participants were in the good faith or formalised licensing and support agreements provided by the software provider.

These three areas directly relate to the key release conditions discussed in Chapter 2.

Once it has been established that the licensee have good or poor report with the service provider, it is possible to consider secondary items, as was seen in the next section.

Table 4 - Identify Threat Sources and Threat Events

Questions	Additional Context	Scoring
The likelihood that Support from the provider will be terminated	To determine the likelihood that Support from the provider will be terminated and that no further development will take place or no updates and requested timely patching provided.	All scored as High, Medium, Low or Very Low
Identify Threat Sources and Threat Events	The likelihood that Support from the provider will be terminated	
Identify Threat Sources and Threat Events	The likelihood that Support from the provider will be terminated	

4.3.2 Identify Vulnerabilities that are Predisposing Conditions on the supplier

Often a software escrow release is triggered as a result of financial distress, the loss of significant staff or when the service provider is acquired by another company.

The overall health, financially as well as in terms of staffing, staff transitions and external dependencies or demands on the software provider's staff needs to be assessed, as shown in Table 5. The value in the two groupings in Tables 4 and 5 (Identify Threat Sources and Event and Identify Vulnerabilities that are Predisposing Conditions on the Supplier) are multiplied in the scoring system to provide the "actual exposure to this threat" in the summary that will be expanded upon later. Moreover, the actual result of this calculation is also turned into a statement terminator for the following line in the feedback component: "Our findings based on the above data to implement software escrow for the evaluated application is that it is …". This is based on a complex IF statement that returns one of the following; Highly Recommended, Recommended, Advised or To Be Considered.

Knowledge of past release incidents, either through bitter experience or good marketing, frequently drive customers to seeking out a software escrow agent who can provide such a service. Knowledge of the low withdrawal rates will impact the price sensitivity of software escrow agreements. Although industry averages of withdrawals from escrow are remarkably low it was possible to determine if the respondents had had prior withdrawals, or if their confidence in their software provider was high. With these low percentages, as noted in Table 6, it is possible to also see that software escrow operates as a kind of risk transferral, or insurance, which the licensee utilises for risk mitigation

Table 5 - Identify Vulnerabilities that are Predisposing Conditions on the Supplier

Questions	Additional Context	Scoring
Financial health of the provider (High, Medium, Low or Not Known)	Your view on the provider's financial health: High - very profitable Medium - just profitable Low - marginal at best Not Known	All scored as High, Medium, Low or Very Low, unless stated otherwise
Staff churn at the provider	The rate of staff churn. High - I speak with different staff each time we engage Medium - some level of staff consistency Low - Mostly the same people still dealing with us Very low - one man shop or very stable	
Dependencies on specialists	At times the development team needs extra help or specialists to come in to assist	
Third party contractual obligations	Do the providers have long-term or prioritized commitments that get more urgent attention than what you get?	
Obligations to debtors	Does the provider have financial commitment that puts them under strain? Do they often or sometimes ask for urgent payment?	

4.3.3 What is the likelihood of occurrence to make a withdrawal in the next 12 months

Knowledge of past release incidents, either through bitter experience or good marketing, frequently drive customers to seeking out a software escrow agent who can provide such a service. Knowledge of the low withdrawal rates will impact the price sensitivity of software escrow agreements. Although industry averages of withdrawals from escrow are remarkably low it was possible to determine if the respondents had had prior withdrawals, or if their confidence in their software provider was high. With these low percentages, as noted in Table 6, it is possible to also see that software escrow operates as a kind of risk transferral, or insurance, which the licensee utilises for risk mitigation.

Table 6 - What is the likelihood of occurrence to make a withdrawal in the next 12 months

Questions	Additional Context	Scoring
% to make a withdrawal – the variables provided ranged from 7.5% down to 0%	Although withdrawal rates are very low, according to some international escrow providers, we have determined that the industry standard for withdrawal rates are 5%, 3%, 1% and lower. Do you believe you will be making a withdrawal of you escrow over the next year?	7.5% down to 0%

4.3.4 Determine magnitude of impact technical and business - Technical Impact

A number of events may be associated with the disruption of software. These items review the technical impact of software disruption, from issues such as actual access to functional software, on-going support from the service provider as well as depth of skills available. Lastly, we determine how this will impact the respondent should she be accountable for the long term access to a working software solution.

Table 7 - Determine magnitude of impact technical and business - Technical Impact

Questions	Additional Context	Scoring
Loss of access	How big is the impact if you lost access to the software and how will it impact the business technically? Will you be able to continue operating as normal without this software?	All scored as High, Medium, Low or Very Low.
Loss of support	If you lost on-going support and maintenance on the software, what will the impact be technically to the business? Will you be able to continue operating as normal?	
Loss of skills	If you lost the skills that supports and maintains this software, will you be able to operate as normal?	
Loss of accountability	Assuming you are accountable for the software what is the risk technically if you lost the application or full access to the application?	

4.3.5 Determine magnitude of impact technical and business - Business Impact

Note: The following research items are based on The Open Group's Technical Standard – Risk Taxonomy. (The Open Group, 2009)

The Open Group (2009) provides for 6 categories of loss. It has been replicated in Table 8 to help determine the business impact that could be attributed to such losses as quantified by the respondents. Finally, the overall exposure to risk associated with software development, as well as desired exposure, is calculated from both these scores and the prior values to give us the Technical Impact.

Table 8 - Determine magnitude of impact technical and business - Business Impact

Questions	Additional Context	Scoring
Productivity	If you lost access to the software what will the impact be that it will it have on the business Productivity? Will it have a visible reduction in the organization's ability to generate income from its primary revenue lines?	All scored as High, Medium, Low or Very Low.
Impact on Response	If you lost access to the software, will the business suffer issues with responding to external or internal demands, due to limited or complete unavailability, and suffer significant quantifiable expenses associated with managing such a loss event?	
Replacement, therefore the intrinsic value of an asset	If you lost access to the software, what will the impact be to typically replace it represented as the capital expense?	
Privacy violation - Fines and judgments	If you lost access to the software, will it result in any privacy breaches or violations of personally identifiable information with associated legal or regulatory actions levied against the organization?	
Competitive advantage	If you lost access to the software, will it result in any lost business or competitive advantages? Within this framework, the loss is specifically associated with assets that provide competitive differentiation between your organization and the present competition.	
Reputational losses	If you lost access to the software will it result in reputational harm and would the associated losses be contributing to external perceptions that the company leadership is contributing to mismanagement, incompetent, criminal, or unethical?	

4.3.6 Business Continuity Planning and Disaster Recovery

Table 9 contain the last two questions. Considering that the King III guidance and governance framework specifically allocates sections of responsibility to the directors it needs to be considered that any business continuity planning and disaster recovery activities must provide for the on-going access to software developed and procured from external parties. In this section we aim to determine if the respondent is aware of these guidelines and whether they are actually considered during continuity planning.

In the second of these research items the intended purpose was to determine whether the service provider followed a structured process for the development of the application. The service provider should also accommodate or facilitate a source code escrow agreement or provide for this within the standard licensing terms.

Table 9 - Business Continuity Planning and Disaster Recovery

Research Items	Additional Context	Scoring
According to King III "IT should be aligned with the performance and sustainability objectives of the company". Does your BCP and DR planning provide for maintenance and recovery of externally sourced software?	The governance and compliance of "good IT" and related processes are assessed.	A yes / no response required
Do the current Software Development Lifecycle as user by your service provider, or the licensing agreement with the service provider allow for source code escrow?	The governance and compliance of "good IT" and related processes are assessed.	A yes / no response required

After formulating these research items we can look how they fit in with some other frameworks to measure the completeness of the survey outcomes.

4.4 Industry Risk Frameworks

A number of models were reviewed, including of Factor Analysis of Information Risk (FAIR)[15] Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE).[16] Figure 11 was deemed a suitable reference item (ISACA, 2009) to proceed with, after adapting it to our content.

Figure 11 - IT Risk Scenario Components

[15] http://www.riskmanagementinsight.com/faq
[16] http://www.cert.org/resilience/products-services/octave/index.cfm

Using the guidance of the ISACA IT Risk Framework (ISACA, 2009), where they provide a sample of the IT Risk Scenario Components, we can depict the software escrow components, as seen in Figure 12.

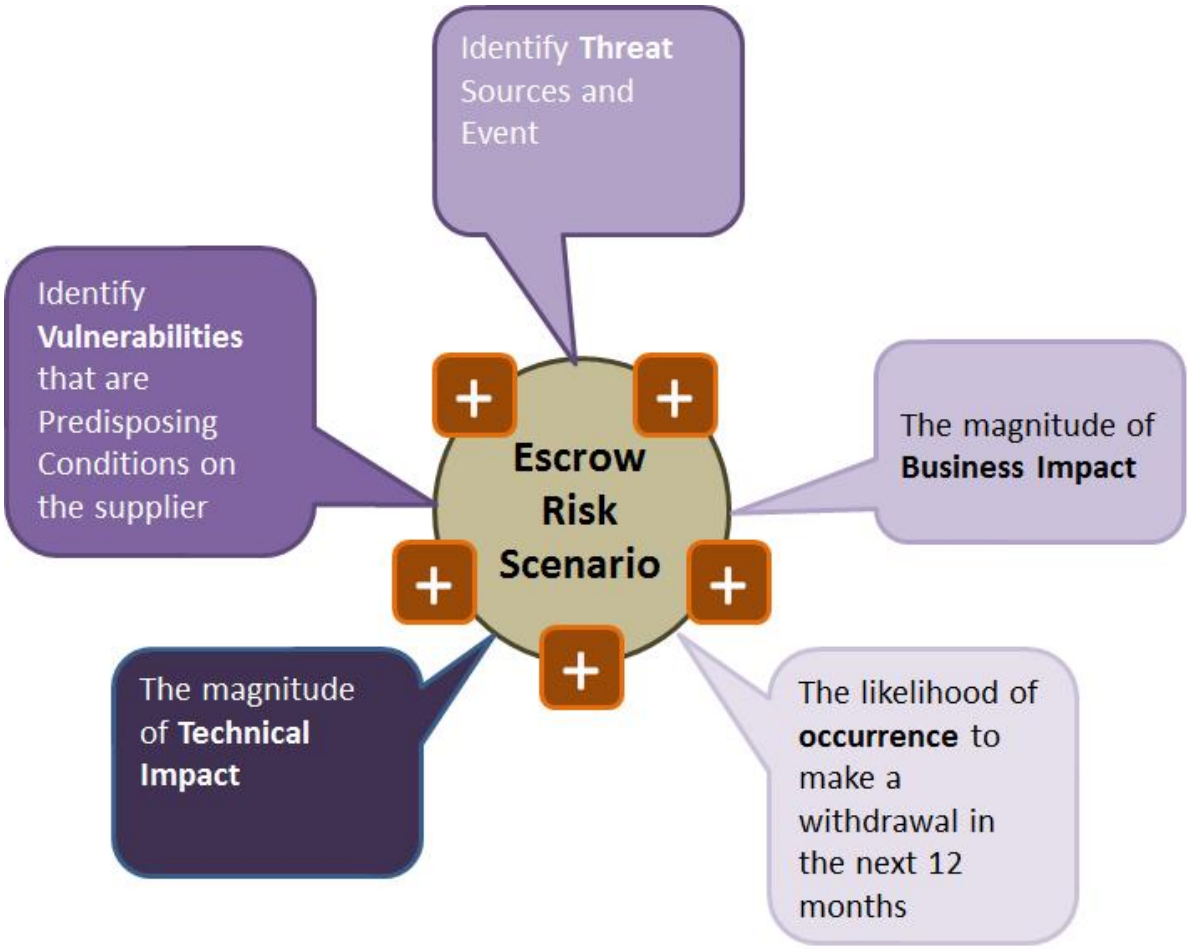


Figure 12 - Software Escrow Risk Scenario Components

Adding the described components of each of the research question sections assists to complete the Escrow Risk Scenario, mapping them as follows:

- Technical Impact to the Action
- Vulnerbilaties that are Predisposing Conditions on the suppier to Threat Type
- Threat Source and Event to Event
- Bussiness Impact to the Asset/Resource
- the Likelihood of Occurance to Time.

It is the cumulative effect of each of these actions that will determine our risk, risk mitigation strategies as well as acceptance of residual risk, if any. The headings above are broken down into the question sections. Each has been populated to reflect the research items found through the research, and have been modelled to assist the participants in determining what level of risk they are being exposed to. There are also other factors to consider pertaining especially to the escrow provider, his operation and services on offer. This will be

discussed in a later section. During the survey each participant reviewed and evaluated the place of the above-noted items within their own environment to determine their various levels based on the ranking provided by the model.

4.5 Scoring of the Items

All of the research items were given the same weight throughout, as it was viewed that no single question had a higher importance that any other; and the desire to gain responses to all of the research items on a level playing field remained important.

In all cases where a four-part answer was available, a response value was allocated (as per Table 10). An exception to this was in Sections 1, 4 and 5 where a response of High was awarded a Nil total and Very low a score of one. This metric also applies to the second section, with the exception of the very first question where Not Known replaces Very Low. Moreover, the scoring is reversed in this case, with the question also serving as a control.

Table 10 - Scoring values

High	1
Medium	0.66
Low	0.33
Very Low	0

Section 3 offered decreasing values from 7.5%, 5%, 2.5%, 1,3%, 1%, 0.5%, 0.3% and 0%, with a reference to industry comments of withdrawal rates of 5%, 3% and lower. A lower percentage than 0.5% resulted in an acceptable score, while any higher values were highlighted as elevated risk. These rates are aligned to the rates of withdrawals noted in (Out-Law.com, 2010), where 150 withdrawals were made from 8000 deposits in a one year period. At this point it is impotent to note that the graphic further on, Sample Research Results Table as per Figure 14, is a sample data set that shows an acceptable rate of 0.3% or one to two withdrawals per 100 escrow deposits over a 5 year period. This is also the level where the need for software escrow will require focus as the value proposition will be easy to support with proper business requirements documentation and motivation.

Lastly, Section 6 contains a binary question with a Yes producing a value of 1 and a No scoring a 0.

The scoring method was selected as it remains linear and does not introduce additional bias into how the responses are provided, given that equal weighting was desired throughout the scoring system.

4.6 Survey Results

Once the respondent had completed the spreadsheet of questions, a results page was sent by return email depicting the question categories, total number of research items per category, rating and score as well as baseline rating achieved for each.

The scores rolled up into a colour sensitive maturity model that reflected red for low maturity through to blue for the optimised level. The same is shown for the baseline. A five point scale was selected, as shown in Table 11, which identifies as median any score below that of "optimised;" a fact that will then provide the respondents with sufficient motivation to take further actions. The key to the resulting output document scale is:

Table 11 - Key to Software Escrow Maturity Levels

Red	Indicates unacceptable high risk. The provided data indicates that the risk is probably exceeding the acceptable appetite and would require immediate response.
Yellow	Indicates raised that is also above acceptable risk norms. Guidance in the form of company policy to mitigate such risk would influence the response.
Light Green	Indicates moderate risk, usually with immediate actions to take other that noting such in the risk register.
Green	Some risk is evident and needs to be noted in the risk register.
Blue	An indicator either the appropriate controls are in place or that the Service Provider is of such substance and stability that no further action is required.

A conclusion section summarises the data in a way that will be actionable and could serve as a further business motivator for taking the next steps.

In developing the research items as well as the response template a few drafts were circulated to peers for review and comments. The scoring system was tested with self-created sample data and also received comments through peer review.

The Excel spread sheet was sent to the participants with the Sample Research Results Table as per Figure 14 hidden in the actual file sent out. Once the artefact was returned it was exposed and a screenshot of it (see Figure 14), as well as a spider graph (Figure 13) were returned to the participant with a note of appreciation for the contribution to the research effort.

Figure 13 illustrates the software escrow scenario risk mapping that was developed to provide feedback to the participants in a graphical representation of their baseline. It is based on the Microsoft Office Suite and participants' response data in the following form.

Within Figure 13's radar chart the respondents can easily determine how far off they are from the desired state, or baseline. This also provides an additional metric to use for further management reporting and budget motivation, should that become a desired outcome, as the graphical representation speaks more specifically to these strengths and weaknesses that can be depicted in this graph.

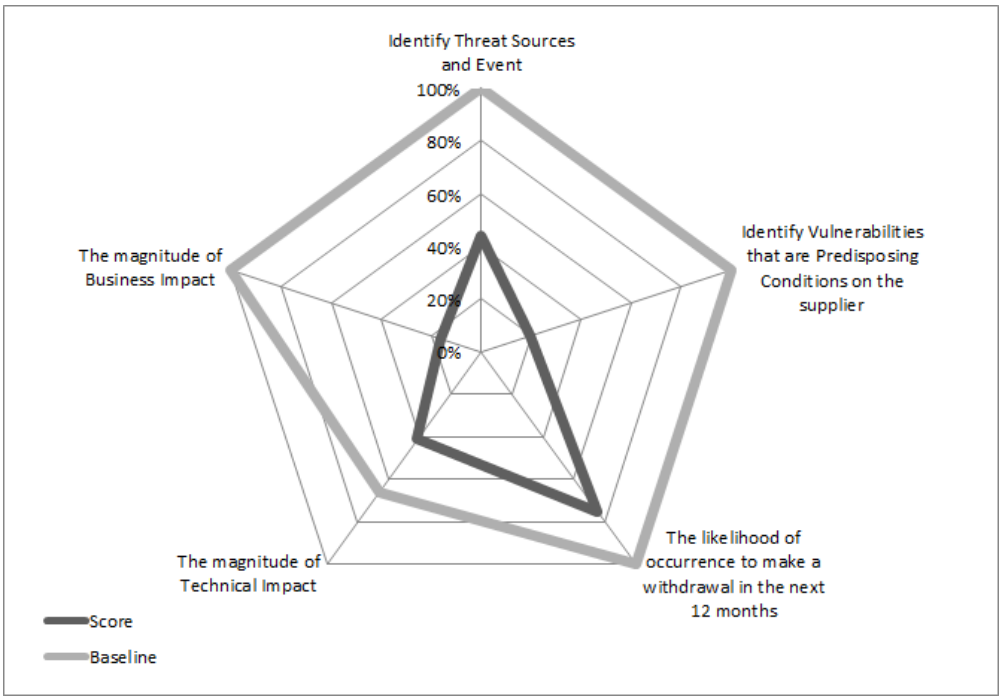


Figure 13 - Software Escrow Scenario Risk Mapping

Other supporting documentation sent with the research template, including an introduction detailing the purpose of the research, a consent form and permission to conduct research as defined by Rhodes University, are included in Appendix C, D and E.

ID	Description	Total	Rating	Score	Baseline Rating	Software Escrow Maturity					Baseline Level
						Level 1	Level 2	Level 3	Level 4	Level 5	
						Chaos	Aware	Action	Solve	Optimized	
1	Identify Threat Sources and Event	3	1.32	44%	100%	Level 2					Level 5
2	Identify Vulnerabilities that are Predisposing Conditions on the supplier	5	0.99	20%	100%	Level 1					Level 5
3	The likelihood of occurrence to make a withdrawal in the next 12 months	1	0.3%	75%	100%	Level 3					Level 5
4	The magnitude of Technical Impact	4	1.65	41%	66%	Level 2					Level 3
5	The magnitude of Business Impact	6	0.99	17%	100%	Level 1					Level 5
	Overall Rating			26%	93%	Level 1					Level 4

Conclusion:

The overall desired exposure to this kind of threat is: 0.0%

Your actual exposure to this threat can be as high as: 91.3%

Our finding based on the above data to implement Software Escrow for the evaluated application is that it is: **Highly Recomended**

The likelihood that you will call on the Escrow provider to make a withdrawal over the next three years is: 1% Industry averages are High (5%), medium (3%) or low (1.5%)

The overall exposure to risk associated with software development is 71.1% and the desired exposure is 17.0%

The company Business Continuity Planning and Disaster Recovery plans Does not Provide for the ongoing management of software not owned by the company.

The current licensing agreement with the Service provider that we use Does not Provide for a Source Code Escrow agreement.

Figure 14 - Sample Research Results

4.7 Summary

Overall the end user feedback was very constructive in providing additional guidance and input in improving the research items and supporting text. Sadly the same cannot be attributed to the feedback that was provided from the respondents as there was none received.

The next step was to look at the actual data received and to determine how the responses differed and drove the individual outcomes and baselines. It was also considered worthwhile to plot all of the user responses into one spider graph to see what points scatter and what groupings forms.

Moreover, each of the above case studies can map back to at least one of the key requirements of the research questions, as shown in Table 12 below.

Table 12 - Link of Use Cases to Research Questions

Research Questions	**Link to Case Study**
Identify Threat Sources and Event	3.4/ 3.5
Identify Vulnerabilities that are Predisposing Conditions on the supplier	3.1
The likelihood of occurrence to make a withdrawal in the next 12 months	3.2/ 3.6
The magnitude of Technical Impact	3.7
The magnitude of Business Impact	3.1

Table 12 therefore represents, as per the user feedback received, a real link to actual experiences of end users, one that also plays into the scenario of the release triggers, as discussed in Section 2.6.1 and in related mappings provided in Table 3.

Chapter 5 – Analysis

5.1 Introduction

A goal of this research was to provide the reader, CIOs as well as IT Security and Compliance Managers with a detailed overview of what software escrow is and how it functions, as per the first objective. For the third objective, a list of the advantages and benefits of implementing escrow agreements was provided, as well as a value proposition through the software escrow maturity model to make an informed decision. This will be suitable motivation should companies choose to engage with an escrow agent for the risk remediation of selected software applications.

The second objective deals with governance, compliance and the impact of King III on the duties of the directors, particularly to ensure continuity, as expanded on in Section 2.7. It also implies that software escrow is a valid option, especially if the due diligence on the software provider indicates that they may be a high-risk entity.

5.2 User responses

All of the participants in the survey were asked to provide feedback on the research items and general content as well as on the output data that was sent to them after completing the documents. The feedback was used to amend the questionnaire in some areas and resulted in the resending of the forms.

5.2.1 From an Escrow Provider Perspective

Of the four known software escrow agents that was approached, constructive feedback came from one local software escrow agent only and an interest to participate from two international agents (from the same company), although they never responded to further probing.

A first comment was passed on the flow of the questions. Here the suggestions were, generally, to move business-related and technical risk up in the research item list, and to add how the company would operate without the software in various loss situations, such as no access, no on-going support and maintenance releases or staff losses at the service provider.

Although the outcome as per the research results was shared with the escrow service provider after receiving these comments, he was of the opinion that a summary of the findings will be of high value, especially as the participants will be able to determine if they should actively pursue software escrow or accept the risk of not implementing such a control. The respondent further suggested that the nature of the model should be more active and cumulative, where the user will be able to play with the responses and find variable suggestions as to proceeding with software escrow or not. Since the research results were hidden from him on his review it is believed that this outcome is possible, and this will be the intention once it was determined that the model is fully suitable for public utilisation.

5.2.2 Feedback from Participating End Users

The total number of end users approached to complete the questionnaire totals 65. Of this sample 13 returned usable data, summarised in the following chapter. A further 15 responded that they were either not in a position to participate or that to their knowledge their company did not make use of software escrow.

5.2.3 Feedback received and general commentary

One of the first research items received from the peer review participants concerned the target audience. Hence, for clarity, a cover page was written that puts into perspective the questions, the overall purpose of the model, the structure as well as the outcome, so that all participants could clearly understand the process. More specifically, the intent was to only target end user organisations, as any licensor who takes risk management seriously would provide for software escrow within his engagement terms.

The initial research items were shared with a knowledgeable software escrow agent as well as one end user who has been actively doing software escrow for the past 10 years. Their comments were incorporated into the final surveys that were presented to the participants. At this time, additional clarity was requested on whether the research focused on inherent or residual ratings. An amendment was added to the introduction page to reflect that only inherent ratings were requested, with no reference required to residual controls, as this could reveal additional sensitive information.

These two users indicated that additional clarity would be required on the relevance of the likelihood of occurrence question in terms of a threat model. This was provided by amending the detail of the help text in the questionnaire.

> *Impact questions are dependent on the criticality of the application going into escrow. I would only submit critical code to escrow and have answered accordingly. To expand, using your model, any software that rates an overall high would be placed in escrow. Software rating below that would not be submitted to escrow.*

Generally this is the goal of the model in that users will self-determine the importance and sensitivity of the application in question, whilst rating the software provider as a viable entity to provide the levels of maintenance and support throughout the use lifecycle.

One feedback provided stated:

> *Your model assumes a rather detailed knowledge of software escrow. I think you overestimate the extent of this knowledge in the industry in general.*

There is no consensus with this statement but it is acknowledged that the selection of the participants was made with careful consideration of their roles, duties and knowledge of the subject area, so as to ensure that a suitable sample was provided. Taking advice from this statement will require an additional introduction and user guidance statements that will position the model, supporting processes for completion and use of the output that is focused on a more diverse user group.

I think your model requires a scale / size section, where the size and complexity of the organisation is considered. In an organisation the size of this large SA bank, there will be a mixture of different scenarios. The questions in your model are very specific and requires a pre-identification of what is actually in place in the organisation.

Does this model scale to small and medium sized organisations? Smaller organisations probably have a higher risk of being affected, seeing that they simply cannot afford to purchase "best in class" software.

These two points pertain again to the knowledge of the responder as well as the diverse scenarios that it aims to address, where larger and smaller companies may have very different views on what can be acceptable risk and what will be deemed just residual risk. Moreover, that actual impact of a loss event may have very diverse consequences to each and every respondent. This will need to be highlighted within the response section of the model, allowing users to be more flexible in their own conclusions and views on individual mitigation strategies.

Will your model calculate a risk weighting? What is the outcome of the model? Effectively, you have only provided an opinion based questionnaire.

The model do not actually calculates the overall risk given the rating of subcomponents.

The model does not provide any guidance on risk treatment/mitigation? What should I do with the results.

These questions were addressed by sending the participants the indicated outcomes, and they agreed that the outcome addressed the concerns.

Software escrow is more of a type of "insurance" against loosing access to critical applications. The best risk management approach would be pre-screening of the vendors and annual re-evaluations of the software vendor's financial health. You want to be in a position to negotiate with them, whilst they are still in business.

Good advice and deemed best practice to note. Aligning escrow in this context to insurance places emphasis on the risk management function that software escrow performs.

The following sections are based on user specific feedback and have been included into the relevant sections as applicable.

5.2.4 Specific Feedback

Section 1: Identify Threat Sources and Threat Events

Obviously depends on criticality of application. We have some business apps that are too expensive to replace.

> *Same as above we will proactively look to either replace the software or mothball the process. I am thinking of legacy Windows XP app not working on Windows 7.*

The above comments show an insight into the value that some applications have within organisations and how the cost to replace, as well as the cost of on-going support and long term platform compatibility, do influence choices to retain legacy applications.

> *Rigorous supplier processes are in place to prevent this from happening.*

An indication that this company will go to great lengths in the vetting processes of the software provider's reputation and capabilities to maintain and support them going forward.

Section 2: Identify Vulnerabilities that are Predisposing Conditions on the Supplier

Pertaining to "Staff churn at the provider", one respondent indicated that it is a concern, but even more so with the offshoring of application development. Moreover, it is our view that with applications developed in remote countries where one has no real insight into the staffing and churn, additional precautions may be the order of the day. Likewise, a lack of understanding and insight into the legal entitlements in other countries may be a further hurdle to consider.

Regarding "3rd party obligations" it was indicated that "*we always check if any client takes more than 50% of a vendor's total business"*. This is good practice and indicates a level of maturity and insight into risk management at the large financial institution that participated.

On contemplating the financial health of a service provider one comment was:

> *This would vary from supplier to supplier - again the rigorous supplier processes will identify these issues.*

This links up with the statement above, and it is probable that the procurement process within this organisation has developed its own screening process to ensure that this particular large financial organisation can shield itself from any untoward risks.

Section 3: What is the likelihood of occurrence to make a withdrawal in the next 12 months

The following responses were provided and indicate past experiences with software escrow. They also align well with the data gathered in the literature review that suggests withdrawal rates are fairly low:

> *We have not yet made a withdrawal in 4 years (Financial Services sector)*
>
> *The likelihood of occurrence would be once every 5 to 10 years*
>
> *Don't see any Likelihood*

Section 4: Determine magnitude of impact technical and business - Technical Impact

Loss of access

> *This would vary significantly depending on the supplier and the software provided.*
>
> *This depends on the nature of the software.*
>
> *We still have copies of the code that we deploy via our code deployment process*

Finally, some divided user comments including in particularly the last one (above) indicate a significant misunderstanding of the difference between the compiled code that they have and the source code that will be retained by the software provider.

Internal Staff trained – no comments

Loss of Support

> *I would guess little but that assumes the entire system is static and the process works at the time of support cancellation*

This user feels that the loss of support will have minimal impact, on condition that the defined functionality is available at time of support termination. The lack of user insight into operating system variables and how these may introduce changes on supported applications through operating system maintenance and patching is rather disturbing.

Other Service Providers – no comments

Loss of Skills

A comment was provided related to the section "Loss of Skills", implying that this could "depend on the stage of the software development lifecycle". Although this is fair comment when passed by a service provider, this was made by a licensee, who is believed have no or little influence on the software development lifecycle, as implemented by the service provider, although they may insist on following the software's lifecycle throughout its development and supporting phases. They will then ensure that the software escrow requirements for the deposits and supporting artefacts are followed by the service provider. On understanding the importance of a software development lifecycle, the question was amended to determine if the service provider, according to the licensee, has implemented such a best practice.

Loss of Accountability

> *Depending on business process the application supports.*
>
> *This would vary significantly depending on the supplier and the software provided.*
>
> *Will need to implement other solution.*

These comments indicate awareness of the sensitivity of the application(s) in consideration, as well as the impact that it may have. It also indicates awareness of the associated risk.

Section 5: Determine magnitude of impact technical and business - Business Impact

Productivity

> *We use mostly bigger vendors but cover ourselves legally as well in contracts.*
>
> *Implement another solution. Short term outage.*

The first area shows a high level of maturity, in that contract negotiations will extend into areas that provide for risk coverage even if legal means such as supplemental contracts (presumably an escrow agreement) need to be signed.

Impact on Response

> *For some applications yes*
>
> *Not Core Business*

The indication that some applications would be impacted, compared to the other indication - that response impact will not be tolerated in the case of core business applications - shows how the respondents viewed the different impacts of the applications evaluated throughout the survey. Additional guidance was added to the informative help to ensure clarity of the question.

Replacement, therefore the intrinsic value of an asset.

> *I would imagine the integration costs would initially be almost more than software costs*

This comments show that depth of knowledge does exist in the respondents, as the supporting costs of replacing the existing solution is not only locked up in the actual software development, but extends to the peripheral costs such as integration. It is important to also consider the user training, support and deployment costs.

Privacy violation - Fines and judgments

> *In most instances we perform due diligence on providers and cover ourselves to ensure we own any client data - guess this could be ignored but then it is breach of contract.*
>
> *The contract addresses this however.*

Contractual liabilities remain high on the list of items that get attention. Reliance on procurement agreements, vendor registration processes as well as licensing contracts are seen as items that could limit liabilities where possible.

Competitive Advantage

> *Yes for sure*

Business was founded on some differentiation and this is expressed as a competitive advantage. Only one participant voiced an opinion and indicated the business losses, due to limited or complete unavailability of the application, will most definitely influence their competitive advantage. This is the view of the majority of respondents as 66% rated this as a high and a further 25% as a medium.

Reputational Losses

It depends on what the software contains, but we usually own all data.

Ownership of data versus not having full access to the software is an interesting view, as expressed. As mentioned in one of the above scenarios (SaaS,) the extraction of data from a provider may be difficult; it is therefore commendable that this respondent retains control of all data.

Section 6: Business Continuity Planning and Disaster Recovery

Most likely for critical systems only.
Not currently as BCP initiatives have just been recently initialized, and still a work-in-progress
We hold all code in software escrow.
Depending on software we have different levels of escrow and for some software we are joint owners of the software.

The key part of the question in this section was "Does your BCP and DR planning provide for maintenance and recovery of externally sourced software?" Three of the responses clearly highlight the use of escrow, although one indicates that they have some form of guidance for doing escrow or not. The others are either specific to only critical systems or to all code. It is comforting to note that continuity and recovery planning extends to the area of code escrow, and it is especially pleasing to have engaged with people within this specific security domain.

5.2.5 Other general comments

The following use case comes from a health care company that participated:

"The company was started by a financial professional and a programmer. The programmer's wife did most of the enhancements of a system developed by them. There was a disagreement between the two owners and the programmer and his wife left the company. If we had the Escrow agreement we could've used the period of inability to provide support and requested the source code to maintain the program ourselves. The financial partner is using new programmers for this system now, but the support is definitely not what it was."

Although it was possible to include the above in our escrow use cases it was left over for this section. More savvy IT practitioners are gaining insights into the value of software escrow. In this example the licensee would have been able to extract the source code from the escrow if sufficient time passed to allow the specific trigger condition to realise. Sadly, it is not known how much time passed until the support was re-initiated.

This user comment was passed on by the participant, who believed they could benefit from a software escrow agreement if it was in place. In this example, a trigger condition related to on-going support and maintenance would be called on to release the source code from the escrow provider. The fact that it was stated that the support was definitely not what it was supposed to be does not imply that the release conditions would have triggered. In fact, release conditions would only have been triggered if the actual definition of support and maintenance was well described and a baseline was established and agreed to. Even with some support still forthcoming, a release may be difficult to justify.

5.3 Discussion

The following sections provide commentary on each of the components of the questionnaire, along with the combined scores with four 'spark lines'[17] [ ] included as well. These scores was defined in Section 4.5 with these spark lines representing the total scores per section in discussion. A spark line is a graphic that shows no horizontal or vertical axis, can fit into a single text line and only shows a trend for the given data set that is represented. Some may have leading or trailing spaces, indicating an absence of data or nil value. A bar style was chosen, as it provides better visibility of the low count of values represented.

A number of key facts have come to light during this research. It is important to note the sheer diversity of personnel within the various organisations that were contacted, as well as their skills and experience to deal with this complex task. They varied from legal departments, compliance, governance, internal audit, risk management through to procurement teams. During the initial research phase, where the focus was on gathering a pool of people to approach for the research interviews, it became clear that general IT practitioners within those organisations that were targeted did not know understand escrow., In cases where they knew about it they could not immediately indicate who within their respective organisations that would be responsible for escrow agreements. Once contact was established with people within the procurement, audit and risk management sections, the engagements were a lot less onerous, and it was possible to move to the point of the discussions with relative ease.

As can be seen from the user comments in Section 5.2, the participation was lively and useful, as it was possible to amend the model through the input that was provided in the various sections. These changes were mostly to improve the readability of the research items or help text, or to provide additional context where some more clarity was required. Most of these suggestions were incorporated, with an updated survey distributed to the participants where appropriate. At no time were the calculation or selection variables changed.

5.2.6 Identify Threat Sources and Threat Events

It was necessary to determine if the comfort levels of the participants were based on good faith or formalised through licensing and support agreements, as provided by the software provider.

In general, the consensus (High, Medium, Low to Very Low) [ ] was that the software provider was financially balanced and unlikely to file for bankruptcy. Similarly, the consensus was that the software provider would

[17] http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001OR

not be terminating support [— ▬ ▬ —] for the current version of the software and that they would remain committed to the on-going development of the solution.

It was also determined that the majority of licensees believed that the current provider would not be likely to discontinue support of a version of the software that the licensee was using, with most responses (12) leaning towards medium and low [— ▬ ▬].

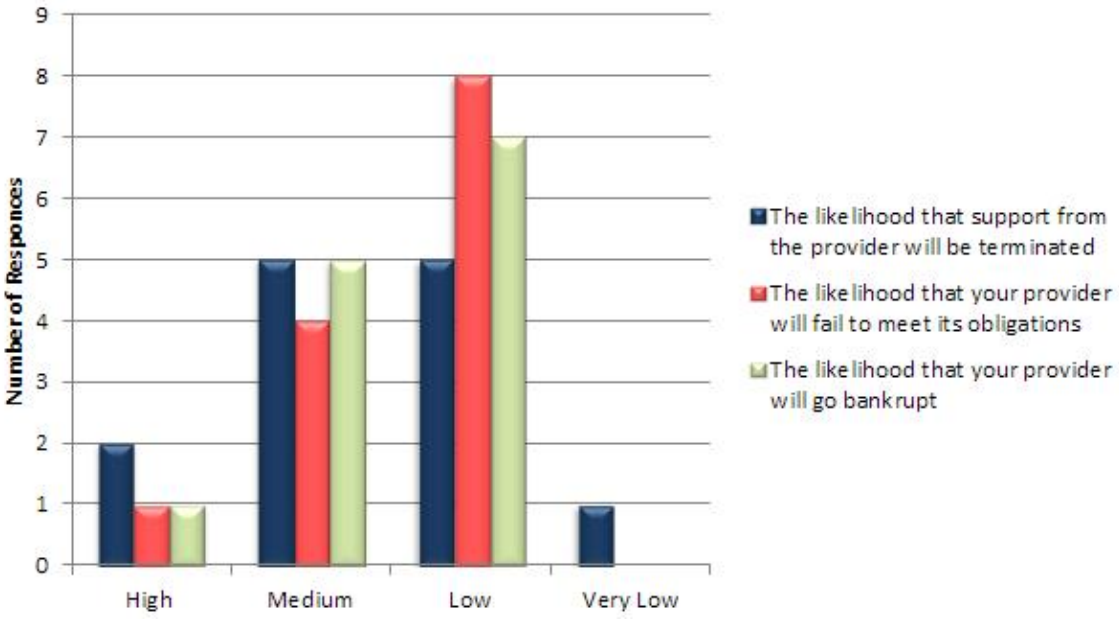


Figure 15 - Identify Threat Sources and Threat Event

Overall the majority of the participants indicated in Figure 15 that their choice of software provider has not, and will not, be introducing support issues over the next year or two, as they seem to be financially stable and keen to provide on-going support.

5.2.7 Identify Vulnerabilities that are Predisposing Conditions on the Supplier

In the second question, as discussed in Section 4.3.2, the dependencies on specialists and third party contractual obligations are the main areas of concern to the participants.

As noted in the case studies in Section 3.8, a provider in distress may be acquired by another firm and that the new firm could then indicate that the support and maintenance conditions were not triggered, even if support has been limited or non-existent for some time. This question points to the fact that knowledgeable respondents are weary of legal disputes and ramifications that will negatively influence their software escrow decisions.

The respondents questioned believed that the financial stability and profitability of their suppliers is high to medium [▬ ▬ —] (a scale of high, medium, low and very low was used and). In the following examples and no one selected very low. Therefore, a stable rating is awarded to this category.

The turnover of staff at a medium rate [▬ ▬ —] can also lead to higher dependencies on external specialists [▬ ▬]. This is a factor that can easily drive up costs for the software provider and should be monitored on an ongoing basis, as the pass through of such practices will impact ongoing support and maitanance costs.

Alternativly, if outsourced to some of the counties that offers low labour rates, additional caution needs to be taken with sample data that is supplied for development and qwality assuarance phases.

With a rating of high and medium, it is clear that the majority of software providers are deemed to be experiencing high utilisation rates of their available resources [ ]. When tied to the reliance on external specialists for on-going development and support, we could be witnessing indicators that allude to long term support issues.

Lastly, the participants did not feel that their software providers were in any financial distress, whether with external debtors or creditors [ ]. This is a positive element and can speak to good care in selecting the software provider, or to long-term relationships that have been established. Figure 16 is a summary of the above.

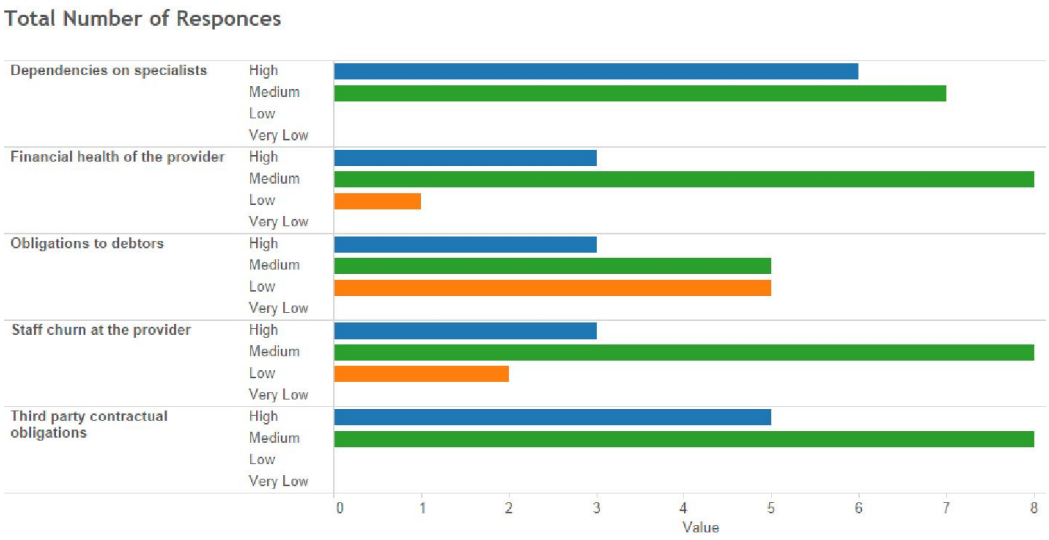


Figure 16 - Identify Vulnerabilities that are Predisposing Conditions on the Supplier

5.2.8 What is the likelihood of occurrence to make a withdrawal in the next 12 months?

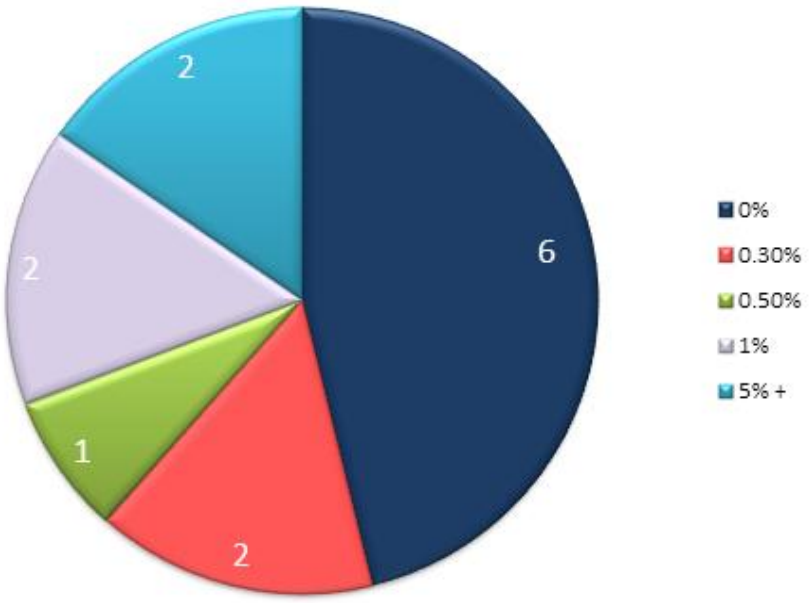


Figure 17 - What is the likelihood of occurrence to make a withdrawal in the next 12 months?

Irrespective of how people viewed the health and stability of the various software providers, they indicated that their withdrawal rates over the following 12 months would be very low to unlikely, as per \Figure 17. Only a few indicated a likelihood of making such a withdrawal and those that did also consistently rated the software providers as more risky in terms of the potential withdrawal indicators. They were also consistent in identifying more risk in the predisposing conditions. This is an indicator of end user awareness, and on further investigation shows experience in the field of software escrow that was not evident in cases where people have not dealt with the topic extensively.

Users of software escrow are very mindful of the actual impact that the loss of access and maintenance to critical applications can have on the business. The common consensus in the comments section varied and included:

1. The likelihood of occurrence would be once every 5 to 10 years.
2. We have not yet made a withdrawal in 4 years
3. Don't see any likelihood

These comments align with the numbers presented in Figure 17, where six indicated that they will be unlikely to make a withdrawal. A further five indicated a one percent or lower probability and only two suggested that they may have a five percentage probability of withdrawal. These rates are aligned to the rates of withdrawals noted in (Out-Law.com, 2010), where 150 withdrawals were made from 8000 deposits in a one year period.

5.2.9 Determine magnitude of impact technical and business - Technical Impact

As per Figure 18, the responses indicated that as a technical impact, the loss of ongoing support would be of marginal impact, whereas the loss of access to the actual software was rated as a much more serious issue. This implies that users are more concerned with the loss of or failure of the software than with actual support from the software provider.

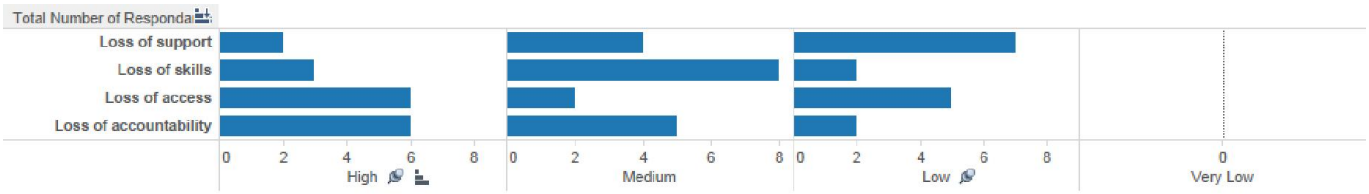


Figure 18 -Determine magnitude of impact technical and business - Technical Impact

This sentiment is echoed by the fact that on-going access to the skills pool that supports and maintains the software was a low level concern for the majority of respondents. This is a peculiar observation, one that points to either a wrongly aligned priority or the possibility that too few of the respondents have actually experienced such incidents to contribute to a differently prioritised viewpoint.

The consensus was divided when respondents were asked whether they were accountable for the software, andwhat the technical risk would be if they lost the application or full access to the application. This also implies that the actual user experiences in this area are very limited, with possibly no practical of such losses to serve as references.

5.2.10 Determine magnitude of impact technical and business - Business Impact

The competitive advantage derived from bespoke software, as well as the impact on response, stand out within Figure 19. Whilst a failure of support and maintanance, as previously discussed, can introduce reputational losses due to the

non-availabilaty of a specific software application - as well as productivity losses and supporting revenue depletion - respondents also implied that their advantage over competitors could be lost should they lose access to the software.

A crucial area of concern for respondents was the impact on responsiveness. If the software was no longer accessable, the business could suffer in responding to external or internal demands. This could accumulate quantifiable expenses associated with the management and eventual recovery from such an event.

A fairly balanced view was provided regarding the violation of customer privacy and any associated fines or judgements. Two things will impact this view over the coming years, with King III leading the charge at present while the duties of directors are gaining more and more airtime in the boardroom from an IT and bussiness continuity perspective.

Additionally, as the local data protection act, Protection of Personal Information Act (POPI) (*Protection of Personal Information Act*, 2013), was passed late in 2013, greater attention to the management and control over personally identifiable information will be required, with the further need to proceed with breach disclosure notification. This will significanly impact future responses to this specific question, as failure to comply with the Protection of Personal Information Act can bring damages ranging from fines of up to R10 million to 10 years in jail. On the subject of the financial impact of data losses through these fines, respondents indicated that subsequent reputational losses would have an impact on their companies.

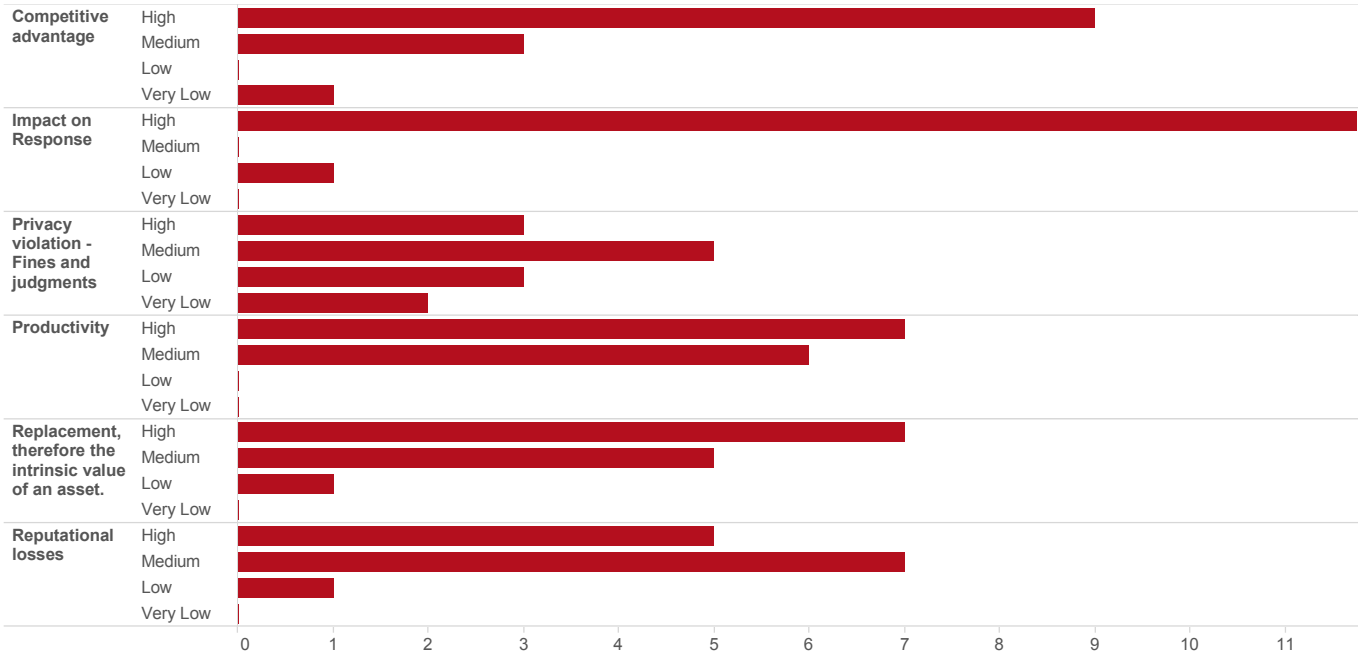


Figure 19 - Determine magnitude of impact technical and business - Business Impact

When asked about the loss of access to the software and how it would impact their business and productivity, the majority of responses indicated that it will have an impact on the organization's ability to generate income from its primary revenue sources. This indicates the pivotal importance of the software and on-going access and support thereof.

5.2.11 Business Continuity Planning and Disaster Recovery

A final question was added to the model to determine the extent of business continuity planning and disaster recovery within a given organisation, as well as whether it addressed the maintenance and recovery of externally sourced software. The aim was to evaluate the guidance from King III, with an especial focus on the notions that IT should be aligned with the performance and sustainability objectives of its company (Stekhoven, 2010), and that business continuity planning and disaster recovery should accommodate for software sourced from external parties.

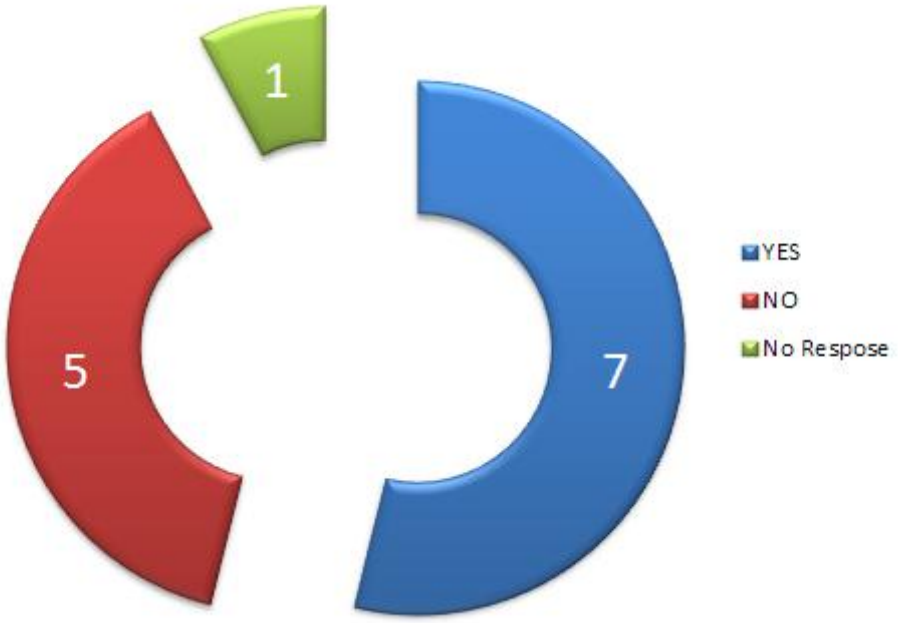


Figure 20 - Business Continuity Planning and Disaster Recovery

Figure 20 shows that the majority (7) of responses were positive, and our confidence is high that a closer to 100% 'yes' response would have been possible if additional knowledgeable resources within these responding companies were approached, as not all of these participants were likely to have sufficient knowledge of the full extent of their company's business continuity planning and disaster recovery coverage.

5.4 Summary

Figure 13 shows the individual spider diagram that was provided to each user. These are based on the score column, as per Figure 14. The aggregation of the user scores are represented in Figure 21, allowing the overall visualisation of the user scores. From these diagrams it is possible to see tight groupings around most of the items, with one anomalous outlier on 5, indicating that, in general, knowledge about the software provider is high and that the business impact of a significant loss event will at all times be more relevant as a risk than other items.

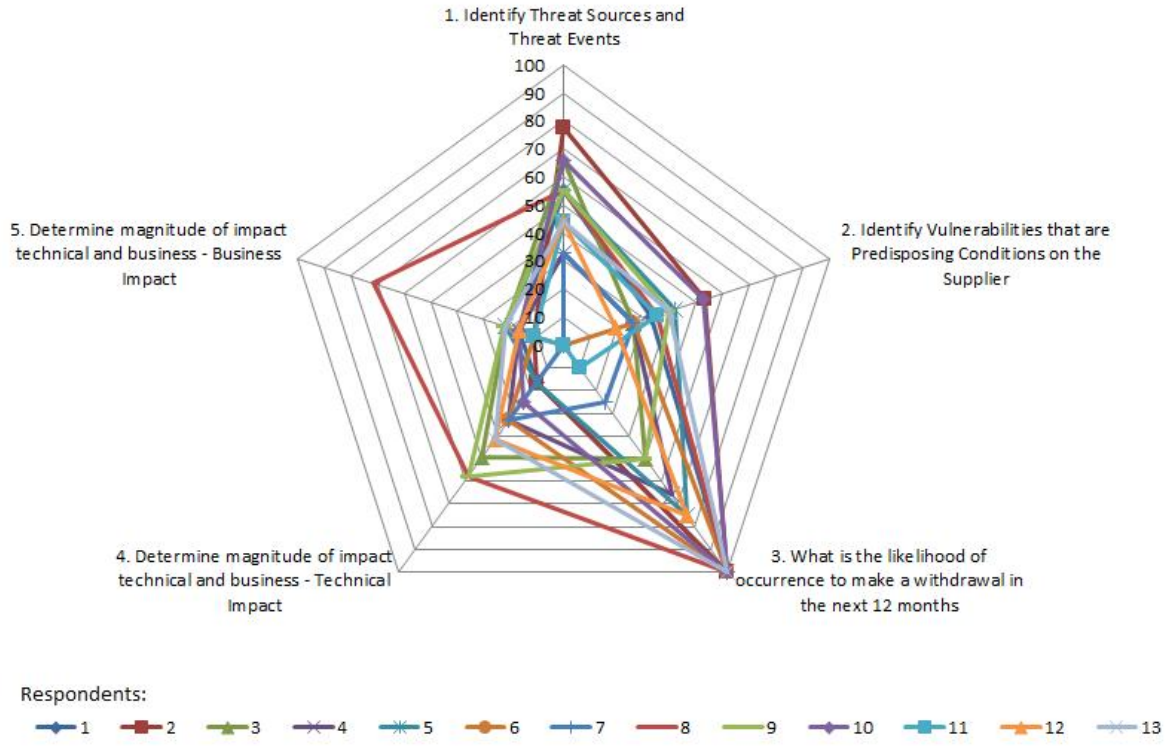


Figure 21 - Aggregation of Scores

The high indicator that a withdrawal within the next 12 months is possible (item 3) could be attributed to past experiences within the various organisations. It is more likely that the calculation of this variable was not fully understood by the majority of responders, as the one party that had experienced the release of source code in the past indicated that the provided range of choice was potentially too wide, and that it would prefer it reduced to a few very low percentages. Therefore, any future work in this area should consider this variable and aim to reduce the scope and complexity thereof. If tied to the low averages for technical and business impact (item 4) then it can be determined that this is an exaggerated anomaly.

In general, a majority of responders indicated that they were comfortable with the stability of their software providers and that they believed they would receive on-going support without the threat of the financial default of the providers. In determining if the model will aid in reducing any residual risk through the implementation of software escrow agreements, it must be concluded that all of the participants scored sufficiently low against the baseline to indicate they would be compelled to find ways of addressing a clear and well defined business risk. Moreover, when viewed as part of good corporate governance, as recommended by King III, this model of risk assessment will aid business continuity planning and disaster recovery (Kane, 2009).

The options available to South Africans when it comes to the various source code escrow providers are not just limited to specialist organisations, with some legal firms also offering escrow. To qualify by reputation only will possibly place excessive credibility on law firms, although they have superior insight into the contractual law and any related common law, experience that may impact the claims associated with an escrow agreement.

Outside the scope of legal offerings, specialist escrow providers offer a more comprehensive service inclusive of advanced inspection, user selectable technical validation services, regular deposit options as well as a user friendly web interface or other methods to track deposit activities.

Chapter 6 – Conclusion

This research was conducted primarily with input from South African participants, with the exception of an additional participant based in New Zealand as well as three representing international South African companies with head offices here in South Africa. The results and findings, however, could be extended to any country where software escrow would be required and where a legal framework exists that can support it.

Recent extended travel by the researcher to the Middle East has highlighted the fact that South Africa is much further advanced in terms of IT Services and related infrastructures. When the topic of escrow was presented at an ISC2[18] event, for example, it was met with very little knowledge and experience in the region, especially if the questions from the conference attendees were any indication of their depth and understanding of the topic.

6.1 Overview

Overall governance and requirements seem to be lacing in many companies, with items such as King III and Basil II serving as compelling motivators for business leaders to follow such guidance. In smaller companies varying interpretations of good governance, as well as a significantly different and higher appetite for risk, forms a set of behaviours that do not offer the risk mitigation that software escrow provides.

6.1.1 Primary Question

In the light of the existence of not only King III (Engelbrecht, 2009), Basel II (Bank for International Settlements, 2004) and Sarbanes Oxley (Oxley & Sarbanes, 2002) as corporate governance imperatives but also COBIT (Brand, 2007), ITIL (Allen & Westby, 2007), ISO27000 (The Open Group, 2010), it is clear that there are sufficient motivations for a business to follow the reasonable man principles of reducing risk through available products and solutions that aim to deliver against fair premium.

The ISO 38500 IT Governance Standard (International Standards Organisation, 2008) takes a top down approach and provides management with tools and views to that effect. As a standard it is focused on three main tasks:

1. Evaluate the current and future use of IT
2. Direct preparation and implementation of plans and policies to ensure that the use of IT meets business objectives
3. Monitor conformance to policies and performance against the plans

The ISO 38500 IT Governance Standard (International Standards Organisation, 2008) has six principles for good governance, namely: Responsibility, Strategy, Acquisition, Performance, Conformance and Human Behaviour. Out of these six there are three key points to highlight:

[18] https://www.isc2.org/EventDetails.aspx?id=11519&display=seminaragenda

1. Acquisition - all IT investments must be made on the basis of a business case with regular monitoring in place to assess whether the assumptions still hold.
    - Therefore all new software purchases must be evaluated, supported with a sound business case, and any risk and mitigation controls as well as residual risk should be noted and reviewed on a regular basis. This will ensure that the software provider makes regular deposits and that the escrow agreement is maintained and serviced as per its terms. Moreover, by doing regular deposits the licensor is assured that a release will provide the most complete and updated version of the code held in escrow.
2. Performance - the performance of IT systems should lead to business benefits and therefore it is necessary that IT supports the business properly.
    - No longer is it just acceptable to replace one system with another without good motivation. IT must support the business strategy (second principle) and measure success on an on-going basis, even though side agreements such as the implementation of active software escrow.
3. Conformance - IT systems should help to ensure that business processes comply with legislation and regulations; IT itself must also comply with legal requirements and agreed internal rules.
    - Although there may be no local legal mandates that compel business to effect a software escrow agreement, there may still see legal precedent in the failure of management to act diligently. This was highlighted in the case studies that were reviewed, with one case still pending as the legal debate continuous. This type of action and its related legal costs, not to mention the cost generated by the disruption of business, can be effectively mitigated through software escrow.

In Section 1.1 it was asked: Can software escrow be seen as a suitable Risk Mitigation Tool within the South African landscape? Having provided a number of case studies in Chapter 3, where the successes pursuing good guidance or failures to plan showed how graceful a recovery may be or how the pursuant disaster can linger. It is possible to tie this in as a motivator to implement software escrow and the facts from King III on corporate governance, as detailed in Section 2.7 with supporting statements in Section 2.8 from Basel II. Knowing that proper governance should be followed may not be a sufficient motivator, even after reviewing the cases studies of Chapter 3.

Therefore, a research effort was commenced and in Chapter 4 the overall end user feedback was constructive in providing additional guidance and input into improving the research items and supporting text that expands on these questions. Chapter 5 reflected on the analysis of the user-provided input where the majority of responders indicated that they were comfortable with the stability of their software providers, and that they believe that they will receive on-going support without the financial default of the providers. In determining if the model will aid in reducing risk through the implementation of software escrow agreements, it can be concluded that as all of the participants scored sufficiently low against their own baseline, they would definitely benefit from the implementation of a software escrow agreement as this will be a suitable way to reduce the determined business risk. When viewed as part of good corporate governance, as stated in Sections 2.7 and 2.8, the recommendation as per the model that was provided would be easy to follow.

Moreover, the risk model serves as a further test to evaluate a number of those key areas known to be accessible to the business decision makers; when compared with a known baseline the results can serve as a further motivational tool to positively motivate for any expenses associated with the software escrow agreement.

6.1.2 Secondary Questions

The secondary questions were:

I) Will the business continuity planning and disaster recovery benefit if critical business applications were covered with source code escrow agreements?

The aim of this question when posed to the respondents was to evaluate their insight into the guidance from King III particularly with respect to the idea that IT should be aligned with the performance and sustainability objectives of the company (Stekhoven, 2010) and that business continuity planning and disaster recovery should accommodate for software sourced from external parties. Although 7 of the 13 respondents replied in the positive, the view is that a much higher positive response is possible based on the profiles of the users that should be performing these activities.

Business continuity planning and disaster recovery can point the reader to the various corporate governance and compliance frameworks and guidelines where all of them urge IT to align with business and to support the business in all aspects of continuity and availability. Even with limited insight into these the reasonable man would want to limit the business exposure that could be introduced by contracting with a software developer that could suffer one of the key release triggers, namely:

- a decision by the licensor or a purchaser of the licensor to discontinue support of a version of the software that the licensee is using
- a material failure of the licensor to meet its support obligations
- or licensor is the debtor in a bankruptcy, is insolvent, or makes an assignment for the benefit of creditors.

Since these triggers are definitely not in the control of the licensee, and on-going access to the software, where support and maintenance will remain a high priority, then the only reasonable mitigation control, other that acquiring the software developer, would be a software escrow agreement between the participating parties. Moreover, where a SaaS solution comes into play the outright purchase of the developer may not be acceptable to the other participating parties (see case study in 3.2), and it will be in the interest of the developer to initiate the escrow agreement. In such a case it would be more advantageous to refer to the multi-party agreement as appended in Appendix C.

II) What makes up an acceptable source code escrow agreement for use in South Africa as there are variations in the levels of inspection and the handling of deposit content?

Although two sample agreements are available in Appendix B and C the more pressing question is what makes up a good enough software escrow agency? In researching what makes up an acceptable source code escrow agreement for use in South Africa, as there are variations in the levels of inspection and the handling of deposit content, as discussed in Section 2.6, it was confirmed that these are legal agreements between the participating parties and that a formal review by the licensee and their legal team will be advised.

A number of questions are also provided in Appendix A, Section A.1, that could be included into an industry research effort, be it through a formal request for proposal by companies considering the virtues and values of more than one escrow agent or just as a generic question set posed to a prospective escrow agent. These questions serves to evaluate the levels of services and inspection provided by the agent as well as the independence, assessing the environment for storing the escrowed content, reviewing of the liability cover provided and competitiveness of the pricing models in terms of the desired services offered. In this case pricing of the service, although a sensitive matter to all businesses should be of a lower importance as the reputation and level of services offered should take priority, if a like for like service offering is compared.

Therefore, to effectively determine the risk that a bespoke developed application has to the company through the risk model proposed, and the implementation of a software escrow agreement with an escrow provider that has a solid reputation with the depth of services that aligns with the company requirements will be the most suitable way to get to a level of manageable residual risk. From the number of case studies referenced as well as the outcomes as described in them it is only reasonable to expect that any reader of this paper will place the implementation of a software escrow agreement high on the list of priorities when in negotiation with an independent software development team.

Once a real or perceived need has been identified, through the use of the model as described in this paper or through a normal business diligence, a call to action will be required. This remains no trivial task and may require in depth knowledge of the software escrow domain. Moreover, this background information will be essential for a specific request for information and request for proposals from software escrow providers that operate and offers services within the organisation's operating geography.

This research provides sufficient information to build out the questions that such a service review will require and through the evaluation of the various risk scenarios and use cases guidance is provided as to how the evaluation of the responses should the performed. Moreover, the singular inclusion of just commercial terms and ignoring the complexities of deposit monitoring and actual inspection of any on-going deposit material will fail good governance practices as the proof of how successful the escrow agreement really is will be left to the day that it is actually called to action.

6.2 Future Research

Future research area indicated by this thesis include particularly issues that are considered highly confidential within the software escrow space. These issues were particularly problematic during initial research efforts, and include:

1. Initial deposit scope and evaluated accuracy of data provided when compared to the escrow deposit list
2. Subsequent deposits
3. Update frequency of minor releases for deposit and perceived value of incremental deposits
4. The frequency of withdrawal
5. The success level of deploying the code withdrawn
6. The additional effort to deploy successfully if applicable

7. The evolution of the escrow providers from actually storing compact digital media, disks or tapes to operating a fully digital archival systems with web front ends and end user interfaces, systems that track and report on agreed deposit objectives and escrow terms.

Insights into these matters as well as supporting statistics, especially if provided from multiple software escrow providers, will establish benchmarks and guideline that could serve the user communities with further insights into the operational features, functions and competitiveness of the escrow providers.

A further area that is closely aligned with the legal domain will be the legal enforceability of an escrow agreement on a deceased estate as well as ability to enforce succession of source code to the licensee should the liquidation of the estate become unviable (after having too many outstanding claims made against it, for example). This area is beyond this research scope and more suitable to a paper from the Faculty of Law.

6.3 Closing

This research has highlighted that many organizations takes risk seriously and will make use of software escrow as that is, within their knowledge, what the reasonable man would to reduce or mitigate this risk. Many objections inclusive of the lack of budgets and no clear senior leadership guidance on what to do can only be addressed through ongoing awareness and education. The fact that the case studies, as was used in Chapter 3, are publicly available stands as testaments to efforts by the software escrow providers to create such awareness, but sadly many companies now resist having such publicity as it could impact shareholder value.

Software escrow is still the only cost-effective way to deal with bespoke development of software, where the developer is not full time staff but a contacted third party or small company that could introduce unforeseen risk as was highlighted previously. Moreover it will provide cost-effective risk mitigation based on the licensees appetite for risk only if the various levels of inspection is fully understood by the licensee. The implications of accepting a lower level of inspection at a lower price point needs to be well communicated, as the higher levels of inspection and validation of escrow content may be seen as just additional bolt on options, if it is not properly described by the escrow provider.

References

Agin, W. E. (2000). Drafting the Intellectual Property License: Bankruptcy Considerations. *Journal of Bankruptcy Law and Practice*, *9*(6), 591–600. Retrieved from http://w.swiggartagin.com/articles/drafting.pdf

Allen, J., & Westby, J. (2007). Governing for enterprise security - implementation guide. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA472572

Andresen, A., & Salyers, J. (2008). Source Code Escrow. Retrieved from http://www.contextlaw.com/files/lbcs_source_code_escrow.pdf

Anonymous. (2011). Significant Delays and Legal Battles Often Accompany a Release. Retrieved November 04, 2013, from http://software-escrow-info.blogspot.com/2011/04/significant-delays-and-legal-battles.html

Banisar, D. (Ed.). (1996). *Cryptography & Privacy Sourcebook (1996): Documents on Wiretapping, Cryptography, the Clipper Chip, Key Escrow & Export Controls*. Diane Pub Co.

Bank for International Settlements. (2004). *Basel Committee on Banking Supervision International Convergence of Capital Measurement and Capital Standards* (p. 285).

Bankruptcy Reform Act. In the House of Representatives (1999). 106th Congress, 1st Session.

Black's Law Dictionary Free 2nd Ed. and The Law Dictionary. (2013). What is Escrow. Retrieved December 07, 2013, from http://thelawdictionary.org/escrow/

Blatt, M. G. (1998). Surviving the World of Fast-Paced Digital Visual Effects Productions with a Technology Escrow. Law Offices of Marc G. Blatt.

Blaze, M. (1994). Protocol Failure in the Escrowed Encryption Standard. Retrieved November 15, 2013, from http://www.crypto.com/papers/eesproto.pdf

Blaze, M. (2011). Key Escrow from a Safe Distance. In *Proceedings of the 27th Annual Computer Security Applications Conference* (p. 5). ACM New York, NY, USA ©2011. doi:10.1145/2076732.2076777

Botterill, L. C. (2012). *Wheat Marketing in Transition* (2012th ed., p. 150). Springer eBooks. doi:9400728034

Brand, K. (2007). *IT Governance based on Cobit 4.1* (3rd ed., p. 166). Van Haren Publishing. doi:9087531168

Bruno, F. A. (2003). Ensure business continuity- Protect investments in mission-critical software. *www.techrepublic.com*. Retrieved September 24, 2012, from http://www.techrepublic.com/article/ensure-business-continuity-protect-investments-in-mission-critical-software/5088773

Burgess, G. (2012). How robust are your escrow arrangements. Retrieved September 29, 2012, from http://computerworld.co.nz/news.nsf/news/opinion-how-robust-are-your-escrow-arrangements

Caelli, W. J. (1996). Commercial Key Escrow: An Australian perspective. In J. Dawson, E and Golic (Ed.), *Cryptography: Policy and Algorithms* (Vol. 1029, pp. 40–64). Berlin: Springer.

Cannon, D. L. (2011). *CISA Certified Information Systems Auditor Study Guide* (3rd ed., p. 696).

Cheng, A., & Helms, S. (2008). Source Code Escrow: Are You Just Following the Herd? *CIO.com*. Retrieved November 04, 2013, from http://www.cio.com/article/187450/Source_Code_Escrow_Are_You_Just_Following_the_Herd_

Cheredar, T. (2013). Nirvanix confirms abrupt shut down rumors (but doesn't apologize for royally screwing customers). VentureBeat. Retrieved October 03, 2013, from http://venturebeat.com/2013/09/28/nirvanix-confirms-abrupt-shut-down-rumors-but-doesnt-apologize-for-royally-screwing-customers/

Christiansen, J. C. (2004). Doing Software Escrows Right. *The Computer and Internet Lawyer*, *21*(6), 9.

Companies Act 71 of 2008 (2008). Retrieved from http://www.justice.gov.za/legislation/acts/2008-071amended.pdf

Covello, L., & Boruvka, J. (2010). Software Escrow : Practical Strategies for Bolstering Licensing Agreements. Retrieved November 12, 2013, from http://legalit.ca/wp-content/uploads/2010/05/lcovello-jboruvka-software-escrow-practical-strategies-for-bolstering-licensing-agreements.pdf

Creswell, J. (2003). Research design: Qualitative, quantitative, and mixed methods approaches (2nd ed., p. 26). Thousand Oaks: Sage Publications. Retrieved from http://www.stiba-malang.com/uploadbank/pustaka/RM/RESEARCH DESIGN QUA QUAN.pdf

Curtis, S. (2013). 16 most expensive apps on the App Store. 13/12/2013. *The Telegraph*. Retrieved from http://www.telegraph.co.uk/technology/apple/10255045/16-most-expensive-apps-on-the-App-Store.html

Daniels, L. M. (2007). Does Your Source Code Escrow Agreement Achieve Its Objectives? Retrieved September 13, 2012, from http://www.innovasafe.com/pdf/Does Your Source Code Escrow Agreement Achieve Its Objectives_Linda Markus Daniels.pdf

De Kock, E. (2010). Who owns software? Lessons from the Haupt case. Retrieved September 20, 2012, from http://www.dekock.co.za/who-owns-software-lessons-from-the-haupt-case/

Denson, W. D. (1998). The Source Code Escrow: A Worthwhile or Worthless Investments? Retrieved from https://acc.dau.mil/adl/en-US/33157/file/6485/#11651 Source Code Escrow.pdf

Dorrington, V. (2007). Fear of vendor bankruptcy no longer biggest driver for software escrow. *Technews Publishing (Pty) Ltd.* Retrieved July 18, 2013, from http://www.cbr.co.za/article.aspx?pklarticleid=4252

Draws, D., Euteneuer, S., Simon, D., & Simon, F. (2011). Short Term Preservation for Software Industry. In *SQS Research* (p. 9). Singapore: International Conference on Preservation of Digital Objects. Retrieved from http://daniel.cqit.de/Publikationen_files/iPres 2011 - SQS Escrow - Short Term Digital Preservation.pdf

Du Plessis, J. E., Hutchinson, D., & Pretorius, C.-J. (2011). *The Law of Contract in South Africa* (2nd ed., p. 472). Oxford University Press Southern Africa. Retrieved from http://books.google.com.kw/books?id=LNb8SAAACAAJ},

Engelbrecht, L. (2009). King III Report on Governance for South Africa. IOD. Retrieved September 27, 2013, from http://www.library.up.ac.za/law/docs/king111report.pdf

Escrow Leads Business Continuity Charge. (2010). *Gauteng Business News*. Retrieved December 14, 2013, from http://www.gbn.co.za/articles/dailynews/1066.html

EscrowTech. (2011). Understanding Software and Technology Escrows. Retrieved July 05, 2013, from https://escrowtech.com/download_u_s_e.php

EscrowTech. (2013). Levels of Technical Verification offered by EscrowTech. Retrieved December 13, 2013, from https://www.escrowtech.com/fees_technical_verification.php

European Committee for Standardization. (1999). A guide to auditing the escrow process (Part 5). Brussels. Retrieved from http://www.cen.eu/cen/Sectors/Sectors/ISSS/CWAdownload/Pages/Withdrawn CWAs.aspx

Everett-Nollkamper, P. (2003). *Fundamentals of Law Office Management* (5th ed., p. 608). Cengage Learning.

General form for Registration of Securities of Small Bussiness Issuers. (2007). Washington DC: UNITED STATES SECURITIES AND EXCHANGE COMMISSION. Retrieved from http://www.sec.gov/Archives/edgar/containers/fix300/1408057/000138575207000013/vemics10sb12g.htm

Hanse Escrow Management GmbH. (2013a). Passive and Active Escrow. Retrieved October 04, 2013, from http://www.hanse-escrow.de/en/escrow/passives-und-aktives-escrow.html

Hanse Escrow Management GmbH. (2013b). Case Study II: No Production Control Without Software. Retrieved October 04, 2013, from http://www.hanse-escrow.de/en/escrow/case-study-ii.html

Harbinger Escrow. (2013). Case Study 1: Multi-national Metailer. Retrieved October 04, 2013, from http://harbinger-escrow.com.au/content/case-studies

Ince, D. (2009). *A Dictionary of the Internet*. *Oxford University Press* (2nd ed.). Oxford University Press.

Info-Tech Research Group. (2009). Six Steps to Surviving an On-Premise Application Vendor Bankruptcy. Retrieved September 26, 2013, from http://www.infotech.com/research/six-steps-to-surviving-an-on-premise-application-vendor-bankruptcy?c=unlock4

International Standards Organisation. (2008). IEC 38500 - Corporate governance of information technology, *2008*.

Iron Mountain. (2012a). Software Escrow Service Workflow. Retrieved July 08, 2013, from http://www.ironmountain.com/Knowledge-Center/Reference-Library/View-by-Document-Type/Data-Sheets-Brochures/Brochures/Software-Escrow-Service-Workflow.aspx

Iron Mountain. (2012b). How Verification Services Fortify Your Software Escrow Solution. Retrieved September 13, 2012, from http://www.ironmountain.com/~/media/Files/Iron Mountain/Knowledge Center/Reference Library/White Paper/H/How Verification Services Fortify Your Software Escrow Solution.pdf

ISACA. (2009). The Risk IT Framework Excerp. Rolling Meadows. Retrieved October 15, 2013, from http://www.isaca.org/Knowledge-Center/Research/Documents/RiskIT-FW-Excerpt-8Jan09.pdf

ISACA. (2013). *COBIT 5 for Assurance* (p. 284). Los Angeles. doi:9871604203400

Jansen, S., & Van De Zande, T. (2010). Business Continuity with SaaS. Dept. of Information and Computing Sciences, Universiteit Utrecht. Retrieved from http://slingerjansen.files.wordpress.com/2009/04/article-1.pdf

Jennings, M. M. (2013). *Real Estate Law* (10th ed., p. 736). Cengage Learning. doi:1133586554

Jones, B. (2012). Cloud Computing Legal Risks And Best Practices. Retrieved December 14, 2013, from http://www.slideshare.net/lisaabe/cloud-computing-legal-risks-and-best-practices

Kane, R. (2009). *Software Escrow for Dummies*. *Software Escrow For Dummies* (p. 44). Wiley.

Karlyn, M. A., & Overly, M. R. (2012). *A Guide to IT Contracting: Checklists, Tools, and Techniques* (p. 448). Auerbach Publications. doi:1439876576

Kayle, A. (2003). Online fraud 101. *IT Web*. Retrieved December 14, 2013, from http://www.itweb.co.za/index.php?option=com_content&view=article&id=23076

Korelc, J., & Tittel, E. (2008). Understanding the need for Business Continuity Management and Disaster Recovery Planning. Retrieved from http://www.aicpa.org/InterestAreas/InformationTechnology/Resources/BusinessContinuityManagementandDisasterRecoveryPlanning/DownloadableDocuments/Understanding_DRP_BCM.pdf

Kshetri, N. (2010). The Economics of Click Fraud. *IEEE Security & Privacy*, *Volume 8*, Pages 45–53. Retrieved from http://libres.uncg.edu/ir/uncg/f/n_kshetri_economics_2010.pdf

McMurdo, G. (1996). Pretty good encryption. *Journal of Information Science*, *22*(2), 133–146. doi:10.1177/016555159602200207

Monnet, E. J. (2011). Protecting your Information Technology Assets. Retrieved September 13, 2012, from http://www.docstoc.com/docs/155689978/Details-of-the-Logitas-procedure-ag

Morris, J. (1988). Source code maintenance and Escrow. *Computer Law and Security Review Report*, *3*(6), 11–16. doi:10.1016/0267-3649(88)90130-6

Motta, C. (2012). Developing an Effective Software Escrow/IT Risk Mitigation Plan for Your Business. *CompTIA*. Retrieved November 04, 2013, from http://bit.ly/1eA9szu

National Software Escrow Inc. (2013). Reasons for Technology Escrow Agreements – A Licensee's Perspective. Retrieved December 13, 2013, from https://nationalsoftwareescrow.com/reasons-for-technology-escrow-agreements-a-licensees-perspective/

Nirvanix. (2013). Notice to Nirvanix Customers (wind-down of business). Retrieved October 03, 2013, from http://archive.is/www.nirvanix.com

Out-Law.com. (2010). Recession forces software escrow releases to jump by 150%. *The Register*. Retrieved July 05, 2013, from http://www.theregister.co.uk/2010/01/06/recession_escrow_boost/

Oxford Dictionaries. (2013). Definition of escrow in English. Retrieved December 07, 2013, from http://www.oxforddictionaries.com/definition/english/escrow

Oxley, M. G., & Sarbanes, P. Sarbanes Oxley Act of 2002. , Pub. L. No. 116 STAT. 745 (2002). 107th Congress. Retrieved from http://www.sec.gov/about/laws/soa2002.pdf

Peters, J., & High Court of New Zealand. (2012). TelecomNew Zealand Limited V Aldous Limited (In Recievership) HC AK CIV-2012-404-3513.

Protection of Personal Information Act (2013). South Africa. Retrieved from http://www.gov.za/documents/download.php?f=204368

Prozesky-Kuschke, B. (2006). Depositum and Escrow: The Current Application Regarding Computer Source Code in South African Law.

Raymond, J. E. (2007). Software Licenses , Source Code Escrows , and Trustee Powers Under 11 U.S.C. #365. *The Journal of Business, Entrepreneurship & the Law*, *1*(1), 25. Retrieved from http://digitalcommons.pepperdine.edu/jbel/vol1/iss1/2

Riskin, B. M. (2012). Twenty-six years later, a Lubrizol split by the Seventh Circuit. *Cadwalader Wickersham & Taft LLP*. Retrieved September 23, 2013, from http://www.lexology.com/library/detail.aspx?g=6da3a26d-145f-44fd-9ba1-e24427085352

Rivner, U. (2009). EPS going down for fraudsters. *Finextra*. Retrieved December 14, 2013, from http://www.finextra.com/blogs/fullblog.aspx?blogid=2453

Schneier, B. (2000). *Secrets & Lies* (1st ed., p. 414). Wiley & Sons. doi:0471253111

Sloan, J. (2013). Nirvanix: Hey, you, get off of our cloud! *Info-Tech Research Group*. Retrieved October 03, 2013, from http://blog.infotech.com/news-analysis/nirvanix-hey-you-get-off-of-our-cloud/

Smith, L. (2012). What should be deposited into a software source code escrow? *EscrowTech*. Retrieved December 13, 2013, from http://www.escrowtech.com/blog/

Speres, J. (2012). Source code escrow agreements and developer insolvency. Retrieved January 12, 2014, from http://www.floorswart.com/insight.aspx?id=860

Stekhoven, A. (2008). Natsure gains "peace of mind" with active software escrow. *FAnews*. Retrieved July 18, 2013, from http://www.fanews.co.za/article.asp?Technology~41,General~1204,Natsure_gains_peace_of_mind_with_active_software_escrow_~4847

Stekhoven, A. (2010). Ignore active escrow at your peril. *Enterprise Risk, Vol 3, Issue 7, Aug*. Retrieved May 13, 2012, from http://www.sabinet.co.za/abstracts/sh_eprise/sh_eprise_v3_n7_a11.html

Stekhoven, A. (2012). CF Active Software Escrows Usefulness for Companies Embracing COBIT 5. *ISACA*. Retrieved September 13, 2012, from http://www.isaca.org/Knowledge-Center/cobit/Documents/CF-Active-Software-Escrows-Usefulness-for-Companies-Embracing-COBIT-5.pdf

Stekhoven, A. (2013). Verification the basis for quality of every Escrow deposit. Retrieved December 07, 2013, from http://www.escroweurope.co.za/verifications/index.html

Steuerbrerater, S., & Wirtschaftsprufer, R. (1998). Escrow Guide to Source - Guidelines for Acquirers, Developers, Escrow Agents and Quality Assessors - Part 1 to 5. Retrieved September 24, 2012, from http://www.ncc.co.uk/files/documents/Escrow_Guide.pdf

Stulman, J. J. (2008). Technology Escrow Agreements and Software-as-a-Service. Retrieved July 18, 2013, from http://www.innovasafe.com/pdf/Technology Escrow and SaaS.pdf

The Financial Reporting Council. (2005). Revised Guidance for Directors on the Combined Code. Retrieved from https://www.frc.org.uk/Our-Work/Publications/Corporate-Governance/Turnbull-guidance-October-2005.aspx

The Open Group. (2009). Risk Taxonomy. Retrieved January 10, 2013, from www.opengroup.org/onlinepubs/9699919899/toc.pdf?

The Open Group. (2010). Technical Guide FAIR – ISO / IEC 27005 Cookbook. Retrieved July 10, 2013, from http://www.businessofsecurity.com/docs/FAIR - ISO_IEC_27005 Cookbook.pdf

VISA. (2013). ATM Cash - Out Fraud Cases. Retrieved November 15, 2013, from http://krebsonsecurity.com/wp-content/uploads/2013/01/DataSecurityAlert_ATM-CashOut_Jan2013_v1.pdf

Appendix A - Supporting Information

This paper have now examined many aspects of software escrow and consider it a value proposition that it is worth the effort to think through the questions to be asked should a licensee consider asking the service provider to place his source code in escrow. Once such an agreement is reached, you can select a suitable escrow agent for your business needs. With a licensee asking the licensor to place items in escrow, the licensee will have the choice of selecting the escrow agent. Moreover, with the licensee responsible for the escrow agent fees the choice as well and independence expected from the licensee or service provider becomes paramount.

A.1 Qualifying questions

The preceding sections provided some insights into successes and failures with and without software escrow. It also covered user provided comments and a review of the respondents' inputs to a model that has been designed and evaluated. When moving to the next step in selecting an escrow agent a fair review should be undertaken. Specific questions should be asked of the escrow agent when selecting one from the available escrow service providers. You may also choose to proceed with a request for information or proposal or even a public tender. The maturity, levels of services, type of access available (physical or via a web interface) to the licensee and licensor as well as ease of dealing with the escrow provider needs to be assessed.

The following is therefore a non-exhaustive list of items to consider when qualifying your escrow provider and these items are based on information gathered during the research phase as well as an audit document from the European Committee for Standardisation (1999):

I. Pricing

- How is the service priced?
- Do you require a contract initiation fee to register and create the escrow account?
- What are the annual subscription fees and what do they cover in terms of inspection and verification?
- How many deposits are allowed per year?
- Will more fees be levied for additional deposits, and will they be inspected and verified in the same way as normal deposits?

II. Independence

- Is the escrow agent associated with another firm?
- If they are a legal firm what technical capabilities do they offer that can assist with inspection and validation?
- If they are a bank, what additional ties do that have with the software provider that will/will not introduce any bias?

III. Environment

- How does the escrow provider store the deposits?

- If it is physical media, do the facility provide for authenticated access control and does it have sufficient fire suppression systems, heating and ventilation?
- If all interaction is through digital media and the Internet, how will user access and accounts be provided, authenticated and reported on, and how will the escrow agent inform the licensee of the various deposits and verifications?

IV. Submitting the deposit

- How is the deposit collected? This may include CD or DVD or some other type of electronic submission.
- What verification and validation is done of the items that are submitted for escrow?
- What level of personal involvement does the escrow provider give in terms of working with the software service provider to ensure the quality of the deposit(s)?

V. Inspection levels

- What level of inspection will be suitable for your organisation and available budget and does the escrow service provider provide for this level of inspection?
  - Level 1 - File Listing Verification Report
  - Level 2 - Technical Verification - Deposit Analysis
  - Level 3 - Technical Verification - Build and Compile
  - Level 4 - Technical Verification - Binary Comparison
  - Level 5 - Technical Verification - Test Plan

VI. Liability

- The escrow agent is liable for damage to the material in deposit. What do they specifically exclude contractually and are you willing to accept those binding terms?

VII. Release conditions

- Are the release conditions stipulated and acceptable to you, the licensee?
- Can you amend the release conditions prior to contract conclusion?

VIII. Arbitration

- Has your chosen escrow provider been involved in prior contact arbitration and will they be willing to appoint an independent arbiter when required?

IX. Other Questions

Other than the above questions one can also ask, what makes up an acceptable source code escrow agreement for use in South Africa? The attributes to look for will probably be local presence and expertise. Having to deal with an escrow provider that works in a different time zone may be problematic, but any consideration of the complexities of legal jurisdiction quickly motivates in favour of dealing locally. Fortunately South Africa is blessed with enough entrepreneurship and local expertise to have breadth of choice in escrow providers. There are four local software escrow agents and possibly more than six legal firms that provide software escrow with a further three offshore specialist software escrow agents actively offering services.

Moreover, from engaging with the local escrow providers -those available through internet research as well as legal firms that indicate that they do offer such services - it is clear that a manual submission process still exists and that digital submissions via secure internet transfers are still not a practical and cost effective way for these business to provide these services. This could be blamed on the low volume of work available in the country and therefore it may not be worthwhile to develop a fully digitally enabled solution. Another factor could be the number of software escrow providers currently operating and competing for that business, as the small volume of deposits, requests and other administrative functions can still be addressed through manual processes.

The physical inspection of facilities, side agreements and the inspection of other contractual agreements with landlords, safe custody providers as well as guarding and environmental control providers is critical and needs specialist attention, as dependencies or the failure to maintain such services could impact the contracting parties.

Finally, two sample escrow agreements are provided in Appendix B. The first is for one to one agreements and the second for multiple licensees. The legal validity of these agreements needs to be validated by in-house legal counsel and any amendments need to be accepted by all participating parties.

A.2 Summary

Acting on the need to implement software escrow, if driven to that decision through a risk model as the licensee, or as a result of a standard practice in your company, it will still leave you exposed to the elements of risk that an escrow provider may introduce due to lack of oversight, cost cutting or external malicious acts.

Detailed questioning and active participation in reviewing the capabilities and capacity to serve your needs remains the order of the day when selecting an escrow provider. Moreover, the questions detailed above as well as the contents of this thesis serves as a knowledge base to guide companies of all sorts and sizes towards the realisation of their software escrow needs as well as to what levels of software escrow services would suit their needs and budgets.

This additional section serves as a useful value add for those that are seriously considering engaging a software escrow agent, be it a software developer that sees the additional value of such an activity or a licensee that recognises the risks of engaging with a service provider.

Appendix B - Sample Escrow Agreement - Three Party Agreements

The following is a suitable sample agreement to establish between a service provider (licensor) and the beneficiary (licensee). At the end of the page is the agreement description, in this case a Three Party Agreement, with the three parties being the beneficiary (End User Co), licensor (Vendor Co Software) and the escrow agent (Escrow Europe).

Active Escrow Agreement

Entered into by and between

End-User Co (Pty) Ltd,

Vendor Co Software (Pty) Ltd

and

Escrow Europe (Pty) Ltd.

Agreement Description:	Three Party Agreement
Agreement Reference:	EscrowEurope_3Party_rv1May13

Appendix C - Sample Escrow Agreement - Two Party Frame – Multi Beneficiary

The Two Party Frame – Multi Beneficiary agreement only involved the licensor (Vendor Co Software) and the escrow agent (Escrow Europe). As the licensor takes the initiative to commence an escrow agreement for software that he resells to multiple end user companies and licensees, he may seek a comparative advantage through this type of agreement, or may benefit by having this agreement in place as it will reduce his administrative burden of having to deal with multiple escrow agents and agreements. A suitable addendum to this escrow agreement may exist listing all of the parties that the specified software has been sold to.

escrowREADY

Software Escrow Agreement

Two Party Frame – Multi Beneficiary

between

Software Vendor Co
as Licensor

and

Escrow Europe (Pty) Ltd
as Escrow Service Provider

Agreement Reference: | EscrowEurope_escrowREADY_2PartyFrameB_rv1May13

Appendix D

Other Supporting information

In sending out the above questions the following Instructions page was presented to the recipient on opening the attached spread sheet:

Dear Participant,

Thank you for taking the time to evaluate and comment on the enclosed model.
The questions within the tab marked "Questions" have been designed to determine through a qualitative method how you view Software Escrow. The research is focused on inherent ratings only and will ignore residual risk or mitigation controls, as this could reveal additional sensitive information not required.

The questions were divided into the following categories with supporting help text to guide you through it, should you required additional clarity:

- Identify Threat Sources and Threat Events
- Identify Vulnerabilities that are Predisposing Conditions on the supplier
- What is the likelihood of occurrence to make a withdrawal in the next 12 months
- Determine magnitude of impact technical and business - Technical Impact
- Determine magnitude of impact technical and business - Business Impact
- Business Continuity Planning and Disaster Recovery

Once each of the questions, as well as a supporting baseline, is completed a detailed matrix will be presented to you via return email, providing context and a guiding conclusion.

A baseline application, for comparison purposes was selected and we used the Microsoft Office Suite. Since 99% of respondents will be familiar with Microsoft as an organization as well as be in possession of a moderate to good working knowledge of the Office Suite of products, it was assumed that you would be able to provide a qualitative assessment of the Microsoft offering as a comparative to the in house product in consideration.

Thank you for your time in assisting with this research.

Please return the completed forms via email to me, Karel Rode at g12r6359@campus.ru.ac.za or 1cissp@gmail.com

Kind regards

Karel Rode (CSE, CISSP)
+27829414840

Appendix E

The standard Rhodes University informed consent form

Consent Form:

Masters Research - software escrow as a Risk Mitigation Control

Research Questions:

The aim of this research is to determine how you are applying software escrow as a risk mitigation and risk management tool within your business. We have prepared a few questions that will assist is to reach conclusions in this segment of the research. These questions are divided into two segments, where one will focus on the software developers as one response group and the other on end user organizations with some overlapping questions to serve as control questions.

Deciding whether to participate

Taking part in the research is voluntary. If you do decide to take part you will be given this information sheet to keep and be asked to sign a consent form. If you decide to take part you are still free to withdraw at any time and without giving a reason.

There are no risks in participating in this interview although you may be inconvenienced by taking time out of your busy schedule to be interviewed. There will be no direct monetary benefit to you for your participation. However, the study may have several beneficial outcomes. In particular, it will further our understanding of the topic and contribute to the knowledge in the field.

Confidentiality

Any personal information collected about you will be kept strictly confidential. Identifiers will be removed from the data when the research findings are consolidated into a report and will not be included in any subsequent publications. The anonymised data generated in the course of the research will be kept securely in paper or electronic form for a period of 36 months after the completion of a research project. It may be used for further research and analysis.

Research Ethics

If you have concerns about the research, its risks and benefits or about your rights as a research participant in this study, you may contact Prof B Irwin at b.irwin@ru.ac.za .

Please initial box

1. I confirm that I have read and understand the information sheet for the above study and have had the opportunity to ask questions. ☐
2. I understand that my participation is voluntary and that I am free to withdraw at any time, without giving reason. ☐
3. I understand that the researcher will not identify me by name in any reports using information obtained from this interview, and that my confidentiality as a participant in this study will remain secure. ☐

Please tick box	Yes	No
4. I agree to the interview being audio recorded.	☐	☐
5. I agree to the use of anonymised quotes in publications.	☐	☐
6. I agree that my data gathered in this study may be stored (after it has been anonymised) in a specialist data center and may be used	☐	☐

for future research.

Name and Surname Signature

---------- ----------

Please return the completed forms via email to me, Karel Rode at g12r6359@campus.ru.ac.za

All participant information and consent forms has been kept.

Appendix F

Permission to Conduct Research

RHODES UNIVERSITY

Grahamstown (6140 (

DEPARTMENT OF COMPUTER SCIENCE
Tel: [+27] 046 603 8291
Fax: [+27] 046 636 1915
E-mail: g12r6359@campus.ru.ac.za

21 June 2013

To Whom It May Concern

Re: Permission to conduct research at your institution

Karel Rode (under the supervision of Prof B Irwin) is a Computer Science post graduate student (Masters student) at Rhodes University carrying out research on 'Software Escrow as a Risk Mitigation Tool in a South African Context'. The aim of this research will be to determine the use of software escrow, also sometimes referred to as code or source code escrow. In doing this research a view will be provided on how pervasive its use is, how it is implemented by the various parties involved and how it functions, should the terms of the agreement be called to action to mitigate and reduce risk. The participation and cooperation of your institution is important so that the results of the research are accurately portrayed.

The research will be undertaken through a personal interview with respondents that agree to it. The questions in the interview relate to how Software Escrow is used in a day to day manner and includes questions such as:

- Software code in escrow may need to be released on what conditions
- Have you in the past experienced any of these conditions
- During the contractual phase with the client, do you;
- Ask them if they have an escrow agent (yes / no)
- I/we have my own escrow partner. Please expand on how often the client makes use of your escrow agent versus their own
- When using escrow, who pays for the service
- What have been the most pertinent reasons (benefits) for making use of a software escrow agreement
- When considering the use of Escrow, have you ever contemplated registering a patent for your software

The identity of your organisation will be treated with complete confidentiality. The interview should take about 20-50 minutes to complete.

We look to you for guidance in identifying someone at your organisation that would be suitable to submit the survey and agree to a possible interview at a time and date that suites them.

Thank you for your time and I hope that you will find our request favourable. If you have questions or wish to verify the research, please feel free to contact me at g12r6359@campus.ru.ac.za or my research supervisor at b.irwin@ru.ac.za .

Yours sincerely,

Karel Rode and Prof B Irwin