Radboud University

MASTER THESIS

Reliable Business Information Systems

under complexity and tight coupling

Gerben Janssen van Doorn *(s4369505)*

supervised by

Dr. Matthijs MOORKAMP

second examiner

Drs. Liesbeth GULPERS

March 29, 2020

Contents

1 Introduction **3**

2 Theoretical framework **7**

2.1 Reliable Business Information System 7

2.1.1 Business Information System 7

2.1.2 Role of Information Technology 8

2.1.3 Reliability 9

2.2 Complexity, Tight Coupling and Reliability 9

2.2.1 Normal Accident Theory 10

2.2.2 High Reliability Theory 11

2.2.3 Combination 12

2.2.4 Complexity, Tight Coupling and BIS 13

2.3 Conceptual model 14

3 Methodology **15**

3.1 Data collection 15

3.2 Data analysis 16

3.3 Operationalization 17

3.4 Research ethics 18

3.5 Reliability and validity 19

4 Analysis **21**

4.1 Complexity 21

4.1.1 Size of the BIS 21

4.1.2 Issue of communication 22

4.1.3 Focus on speed 23

4.1.4 Integrations with other businesses 24

4.1.5 Architect and structure 25

4.1.6 Functionality 26

4.2 Tightly coupled 27

4.2.1 Coupling between internal, technological subsystems 27

4.2.2 Coupling between external, technological subsystems 28

4.2.3 Coupling directly involving humans 29
4.3 Normal accident occurrences 31
4.4 High reliability organization practices 32
4.4.1 Decentralized decision making 33
4.4.2 Redundancy 34
4.4.3 Conceptual slack 36
4.4.4 Constant training 37
4.5 Influence of highly reliable practices on BIS design 39

5 Conclusion **41**

6 Discussion **42**

7 References **45**

8 Attachments **48**
8.1 Respondents table 48
8.2 Operationalization table 48

1 Introduction

> It takes just the right combination of circumstances to produce a catastrophe, just as it takes the right combination of inevitable errors to produce an accident
>
> *Charles Perrow*

Driven by the invention of the integrated circuit in 1958 modern society is marked by rapid, technological developments and interconnection through global networks. Information Technologies (IT) double in speed approximately every two years since 1975 and allow humans to analyze and communicate information faster than ever before (Moore, 2006). However, society's increasing reliance on IT also introduces novel security and reliability problems. For example, a failure in the computer aided dispatch system of the London Ambulance Service (LAS) in 1992 is estimated to have led to the death of 20 to 30 people (Beynon-Davies, 1999, pp. 699–700). The Airbus A380 suffered a drastic failure because different national units turned out to be incompatible. The French unit had upgraded their version while the German unit had not (Dörfler & Baumann, 2014). In 1990 half of AT&T's network collapsed, blocking over 50 million calls in the nine hours it took to stabilize the system. The root cause was found to be a one-line bug in the recovery software of each of the 114 switches in the network (Burke, 1995). Furthermore, these failures still happen. Computer-related failures in the National Health Service of the UK are estimated to lead to hundreds of deaths per year (The Independent, 2018).

Beynon-Davies (1999) conducted a study focusing on the LAS failure in 1992 and concluded that the failure arose from a complex interaction of human and technical errors. This type of failure seems to be inherent to digital systems since they are characterized by the discontinuity of effects as a function of cause. Meaning that relatively small changes can produce an unusual large effect (MacKenzie, 1994, p. 245). For example, changing a single bit of information can result in the crash of an entire system. Even when these IT systems run smoothly, they should be "under constant development [...] like the organizations for which they are built are subject to constant adjustment and adaptation" (Truex, Baskerville, & Klein, 1999, p. 123). Logically Beynon-Davies (1999) thus point to the similarities between the LAS failure and Perrow's (1984) normal accident. Normal Accident Theory (NAT) takes a sociological approach to

accidents and "indicate[s] that some failures are not only hard or impossible to predict, but also inevitable products in complex and tightly coupled systems" (Müller, Koslowski, & Accorsi, 2013, p. 3). These specific, design-related failures are called normal accidents.

Since the development of NAT many studies have focused on complex organizations working with high-risk technologies, such as: "air traffic, marine traffic and chemical plants" (Whitney, 2003, p. 2). However, little research has been done towards reliability in (Business) Information Systems (BIS) even though their reliability is crucial in critical systems and BIS show patterns of being complex and tightly coupled. It is found that failures in BIS "have not been caused by simple breakdowns in their functioning, but by breakdowns in the larger web of computing in which the equipment residues" (Winograd & Flores, 1986). Furthermore Butler and Gray (2006, p. 217) state in their chapter "Structuring Information Systems Operations to Handle Normal Accidents" that "there is little work, either normative or empirical, related to the work practices, structures, or personnel arrangements that make reliable IS [Information System] operations possible".

Because of the lack of knowledge on reliability in BIS, this research will focus on exploring this area. Problematic however is that on its own NAT is a theory on the causation of a specific type of accidents which makes it hard to assess directly the impact of complexity and tight-coupling on overall reliability (Rijpma, 1997). To overcome this problem Rijpma (1997) combines NAT with its seemingly counterpart: High Reliability Theory (HRT). HRT states that some organizations have an outstanding safety record despite their complexity and tight coupling. These organizations are called Highly Reliable Organizations (HRO). The reason for their safety record is the application of four HRO practices: decentralized decision-making, redundancy, conceptual slack and comprehending complexity through constant training (Lekka, 2011). In combining the two Rijpma (1997) finds that NAT can also be used to explain overall reliability, while HRT can also highlight factors which contribute to an organization's proneness to normal accidents.

Therefore, this research will explore if NAT can be used to evaluate the proneness of a BIS to normal accidents, while using HRT to explore how HRO practices work with this level of proneness to aim for high reliability. This relationship is graphically depicted in a conceptual model in section 2.3.

Given the research problem the main research question becomes: '*What is the relation be-*

tween BIS system design and reliability and how is reliability subsequently developed through highly reliable practices?'. This main question can be broken down into two sub-questions: (1) '*What is the relation between BIS system design and reliability?*' and (2) '*How do highly reliable practices develop reliability given the relation between BIS system design and reliability?*'.

A qualitative, expert interview approach is used to gain in-depth knowledge about this area. This approach best fits the knowledge gap present in the current literature and allows the researcher to focus on previously unknown relationships that come up during the data collection. The interviews are conducted with (software) engineers from different organizations to give this exploratory research the opportunity to create hypotheses for BIS independent of organizational contexts.

This research is practically relevant because it offers a first step towards establishing a relationship between BIS design and BIS reliability. This gives organizations the basis to conduct further research on this relationship in their own context and work towards a better explanation of BIS reliability. Furthermore, the results of this research give organizations early indicators as to how system design (change) can influence BIS reliability and how organizational practices impact that relationship. This is relevant for design questions like: how will expanding a digital business platform influence the degree of complexity and tight coupling of the BIS? And as a followup: what organizational practices should be carried out to deal with the changes? These indicators are important as more and more critical infrastructure and profits of businesses rely on BIS.

This research is also academically relevant because this research develops hypotheses on the relationship between BIS design and BIS reliability that can be used for future research. The contribution of this research to future research has three dimensions: exploring the degree of tight coupling and complex interactions in BIS, the influence of the these design parameters in BIS on BIS reliability and how HRO practices influence the relationship of the BIS design parameters on reliability. This knowledge is currently lacking in the existing literature and where reliability of IS operations were considered the approaches taken are largely atheoretical (Butler & Gray, 2006, p. 217).

To answer the research question chapter two contains the theoretical framework. The theoretical framework is divided in two parts. The first part conceptualizes reliability in the context of BIS. The second part describes why researching the influence of complexity and tight cou-

pling on BIS reliability is relevant. To do so a combinatory approach of NAT and HRT is discussed and adopted. Chapter three focusses on the methodology used for the data collection. Choices made regarding the methodology are justified and explained. Chapter four contains the analysis of the data and chapter five and six contain the conclusion and the discussion.

2 Theoretical framework

2.1 Reliable Business Information System

2.1.1 Business Information System

A common view of IS, as summarized by Beynon-Davies (2004, p. 49), is a system "involved in the gathering, processing, distribution and use of information". These activities, like the distribution of information, are performed or facilitated by the elements in the system which commonly consist of "hardware, software, data, people, and procedures" (Silver, Markus, & Beath, 1995, p. 363). Information systems are important and common as all systems depend on the input of accurate information to perform control processes effectively. Information thus assist human activity in the sense that it enables decisions to be made about courses of action in particular circumstances (Beynon-Davies, 2013, p. 18). Given that the information system itself aids human decision making, Silver et al. (1995) use general systems theory to justify that an IS should be analyzed by determining its function in the supersystem (e.g. organizational system). In that view the information system can be seen as part of a larger system which gives it purpose and determines to which goals it contributes.

The information used and produced by a BIS, contrary to an IS, is not used for any purpose or any supersystem, its role is aiding the business and its decisions. Making this distinction is valuable since it changes the function of the system. Instead of producing information for a generic purpose, a BIS has a specific purpose namely, aiding business decisions. This purpose is for example different from societal information systems that aim to benefit society.

To give a more concrete overview of different types of BIS this research uses the study of Alter (1976) who researched 56 computerized information systems. The study divides these systems in six different categories based on what the user does with them. A short example is given for each type:

1. Retrieves isolated data items: operators submit daily reports based on which foremen juggle the information to obtain productivity data per operator.

2. Uses as a mechanism for ad hoc analysis of data files: a portfolio analysis system which aids in making authorized trading decisions by providing risk assessments.

3. Obtains prespecified aggregations of data in the form of standard reports: analyzing sales information in conjunction with proprietary data bases and models.

4. Estimates the consequences of proposed decisions: a budget system using projections of future business levels to generate projected overall cash flow by month.

5. Proposes decisions: an optimization system to solve the mathematical puzzle of choosing and balancing among various product recipes in times of shortage.

6. Makes decisions: based on coded input sheets the system calculates an insurance renewal rate using a series of standard statistical and actuarial assumptions.

(Alter, 1976)

2.1.2 Role of Information Technology

In the academic literature there is a lot of discussion on the role of IT in BIS (Orlikowski & Iacono, 2001). In the previous section, IT is not specifically mentioned however due to the scale and size of data that modern age businesses need to process, IT often plays a central role in IS systems. Lee (2001) puts it this way: "research in the information systems field examines more than just the technological system, or just the social system, or even the two side by side; in addition, it investigates the phenomena that emerge when the two interact".

This interplay is also the core of the earlier mentioned discussion taking place in the literature. Some IS research focusses mainly on the social aspect of technology-based systems, called the broad view. While the narrow view advocates to go back to the roots of IS and to see the IT artifact as the core part of IS (Mansour & Ghazawneh, 2009). Orlikowski and Iacono (2001) researched the current state of IS research and the role credited to IT. Based on a literature review of 188 articles they cluster IS research into five broad meta-categories:

1. Tool view: is the engineered artifact, expected to do what its designers intend it to do.

2. Proxy view: focus on one or a few key elements that are understood to represent the essential aspect, property, or value of the information technology.

3. Ensemble view: the technical artifact is a central element in how we conceive of technology, however it is only one element in a "package". A main focus in this view is the interaction between technology and people.

4. Computational view: concentrates on the computational power of information technology.

5. Nominal view: technology is only mentioned by name, but not in fact.

Furthermore, Orlikowski and Iacono (2001) present their assertion that IS research has not seriously engaged its core subject matter: the IT artifact. However, they also stress that the context and capabilities should be taken as serious as the technology. This research follows their assertion and adopts the ensemble view of IT in IS research. This choice has been made because this perspective allows the research to focus on the issues and risks that come with using an IS system and its IT artifacts, while also allowing to relate these risks to the people creating, adopting and adapting the IS system.

2.1.3 Reliability

Now that BIS and the adopted view are conceptualized it is important to define what it means for a BIS to be reliable. Leveson (1986, p. 135) defines reliability in the context of software systems: "the probability that a system will perform its intended function for a specified period of time under a set of specified environmental conditions". They further state that safety and reliability are often unjustifiably equated. Safety concerns "the probability that conditions that can lead to a mishap do not occur, whether or not the intended function is performed" (Leveson, 1986, p. 135). Reliability concerns a failure free and functional system, safety concerns a mishap free system. The two concepts can even be in conflict, sometimes the safest system is the one that does not work at all (Leveson, 1986). Reliability therefore, is related to the function of the system, where safety is not.

Having clarified this distinction this research uses the earlier given reliability definition and tailors it towards BIS. This specification is done replacing the generic system for a BIS which has been defined in section 2.1.1. The definition for a reliable BIS becomes: *the probability that a BIS will perform its intended function for a specified period of time under a set of specified environmental conditions*. This definition will be further used in the operationalization to make BIS reliability measurable.

2.2 Complexity, Tight Coupling and Reliability

It is still unclear what the relationship between a reliable BIS and system design could be. To this purpose this research uses two major theoretical works on the origin of accidents and reliability, NAT and HRT. Their origins both lie in the study of "the most serious accident in

U.S. commercial nuclear power plant operating history". The Three Mile Island reactor partially melted down due to "a combination of equipment malfunctions, design-related problems and worker errors". Its aftermath brought about sweeping changes, significantly enhancing U.S. reactor safety (United States Nuclear Regulatory Commission, 2013). One of the researchers studying this accident was Charles Perrow, whose study lead to his description of a 'normal accident'. Another researcher, Todd La Porte, also studied this accident in the context of highly reliable organizational performance under very trying conditions. Both researchers developed, in studying this accident, influential theories on accidents and the role of system design and have since often been contrasted against the other (Rijpma, 1997).

2.2.1 Normal Accident Theory

Perrow's (1984) concept of the normal accident comes from the observation that accidents in complex, tight coupled systems are seemingly normal events. He presumes that these accidents are inevitable and incomprehensible because seemingly unrelated events add up and combine into a major malfunction.

Why do normal accidents occur? To answer this question Perrow (1984) identifies two design parameters: types of interaction and types of coupling in a system. "Interactions are the reciprocal actions among elements of the system" (March & Cyert, 1992). Interactions can be either linear or complex. Linear interactions are familiar or easy to spot and complex interactions are unfamiliar and not (immediately) comprehensible (Perrow, 1984, p. 78). Complexity thus leads to unexpected interactions between unrelated events because members of the organization do not anticipate these interactions since they cannot comprehend the complex system.

These linear and complex interactions can also be coupled in two different ways: loose or tight. Loosely coupled interactions means that events in a system can occur independently from each other. Tightly coupled interactions mean that different parts in the system are highly dependent on each other hence a failure in one part of the system can easily propagate to a higher level. In this context Perrow differentiates between an incident and an accident. An incident only affects a part of a unit while an accident affects an entire (sub)system. An example of a tightly coupled interaction in the public transport system is that a bus strike will often create a shortage of taxis (Perrow, 1984, p. 8).

These types can be put in a 2x2 matrix and this is where the core of NAT as a design theory lies: an increase in interactive complexity and a tightening of the coupling lead to a system that

is more prone to normal accidents. According to Perrow (1984) only this combination leads to the occurrence of normal accidents. For example, interactive complexity and loose coupling or tight coupling and interactive linearity do not lead to the occurrence of normal accidents. The system in this case is predictable enough to see accidents coming or to trace them down (Perrow, 1984, p. 5).

2.2.2 High Reliability Theory

La Porte's study on High Reliability Theory claims they have discovered practices and strategies that have achieved outstanding safety records in organizations facing complexity and tight coupling. These organizations are named highly reliable organizations (HRO) and achieve simultaneous centralization and decentralization. People benefit from learning the lessons of previous colleagues and from their own trial-and-error processes. When errors happen people need a clear chain in command to deal with the situation. However a system in which both centralization and decentralization occur is difficult to design. Therefore before decentralizing, HRO's have to centralize so that people are socialized to use similar decision premises and assumptions so that when they operate their own units their decentralized are equivalent and coordinated (Weick, 1987, p. 124).

Secondly these organizations build in redundancy: if one component fails, another backs it up. To contain unexpected events HRO have back-up systems in place, cross check important decisions and continuously monitor safety critical activities (Lekka, 2011).

Thirdly, HROs apply a strategy of conceptual slack. As defined by Schulman (1993): "conceptual slack is a divergence in analytical perspectives among members of an organization over theories, models, or causal assumptions pertaining to its technology or production processes". Course of action is only decided after is has been discussed and negotiated thoroughly.

Finally, emphasis is put on constant training to develop an understanding of the complexities of the technology and production processes. "Trial-and-error learning is supplemented by constant training, operations and simulations in order to maintain and improve standards" (Rijpma, 1997). This allows operators to recognize emergencies and respond to unexpected problems appropriately. Furthermore training is also seen as a means of building interpersonal trust and credibility among coworkers (Lekka, 2011).

2.2.3 Combination

When comparing the two schools it seems they are complete opposites of each other. NAT describes that accidents are inevitable, HRT claims that organizations can significantly influence the prevention of these accidents by using specific strategies. However, when applying these theories to case events Rijpma (1997) found that these theories can also reach similar conclusions. This mixed view of NAT and HRO is therefore discussed and useful for gaining insight in how it might be possible to work towards a system that is better able to prevent normal accidents.

To systematically relate the two theories Rijpma (1997) first analyzes how complexity and tight coupling impact reliability by examining their impact on the four HRO practices discussed earlier. Rijpma's (1997) research shows that complexity and tight coupling have mixed effects on reliability. On the one hand they increase the need for redundancy, decentralization, conceptual slack and constant training. For example, complex systems need redundancy to keep track of all the possible interactions between the various parts of the system. Moreover, complexity creates a need for diverging perspectives making rigid perceptions less likely. On the other hand, complexity and tight coupling decrease the reliability of these strategies due to the complex and tightly coupled environment these strategies are executed in. For example the design of the Challenger space shuttle's Solid Rocket Booster's sealing was redundant. Two O-rings were fitted in that would back each other if the other would be eroded. However, both rings were dependent on the weather conditions and if one would fail the probability of the second ring failing increased (Rijpma, 1997).

Secondly, Rijpma (1997) analyzes how HRO practices affect complexity and tight coupling and thus the potential for normal accidents. Again, a mixed picture appears. First of all, HRO practices lower complexity: redundancy generates extra information; learning lowers the complexity by gaining a better understanding and conceptual slack allows an organization to better anticipate higher number of complex interactions. However, HRO practices also increase tight coupling and complexity because: an increase in redundancy induces ambiguity and opaqueness; conceptual slack could create confusion and decision premises increase the level of tight coupling.

In conclusion, what this mixed view shows is that both theories highlight the same tension but from a different perspective. NAT uses a systematic design perspective containing two design parameters which can lead to a negative effect on reliability. While HRT uses a practice

perspective containing four strategies which indicates that strategies increase the reliability of complex and tight coupled systems. As proposed by Rijpma (1997) this research will use this cross-fertilization to justify measuring the influence of complexity and tight coupling on the reliability of BIS and secondly to guard against an over-pessimistic view of accidents which can be induced by NAT.

2.2.4 Complexity, Tight Coupling and BIS

Having defined BIS and NAT still does not indicate whether a BIS can be complex and tightly coupled. If this is not the case normal accidents will not occur according to Perrow (1984) and the chosen main research question would not be relevant. When looking at well-publicized failures in large computer systems, which make up a core part of a BIS, it is found that these failures "have not been caused by simple breakdowns in their functioning, but by breakdowns in the larger web of computing in which the equipment residues" (Winograd & Flores, 1986). Furthermore, research done by MacKenzie (1994) about computer-related accidental deaths states programmable electronic devices introduce relatively novel hazards which have common features across sectors. Software-controlled systems, including BIS, tend to be logically complex. Meaning that code often interacts with other code in a manner that is not easily comprehensible. This complexity also increases the danger of these systems containing potentially risky design faults. Even changing a single bit of information can have devastating effects. Moreover, this research shows that system failures are rarely just based on technical incidents. "The fatalities in the data set resulting from human-computer interaction problems greatly outnumber those from either physical causes or software errors" (MacKenzie, 1994, p. 245).

Clearly, both examples of Winograd and Flores (1986) and MacKenzie (1994), show systems with complex interactions (e.g. relationship between causes and the system failure is difficult to spot) and interactions that are tightly coupled. Both examples indicate the destructive potential and risky design faults that seem to be inherent to these logically complex software-controlled systems. Secondly, these examples fit well with the earlier chosen ensemble view of IS because both researches indicate not merely technical incidents lead to a system failure.

Other factors like human-technology interaction contribute to the occurrence of a system failure as well. However, these examples are merely indicators of a relation between complex interactions in combination with tight coupling and BIS and should not be seen as proof on their own. What they do show is that the design parameters described in NAT and practices in HRT

are relevant in a BIS context and worth researching.

2.3 Conceptual model

Based on the first research question: '*What is the relation between BIS system design and reliability?*' a direct relation is researched. Based on the theory the indication is that a combination of complex interactions and tight coupling will lead to a decrease in reliability. Even though Rijpma (1997) describes mixed effects it is expected that a BIS with complex interactions and tight coupling leads to higher proneness to normal accidents which in turn will lead to decreased reliability.

Based on the second research question: '*How do highly reliable practices develop reliability given the relation between BIS system design and reliability?*' a moderated relation is researched.

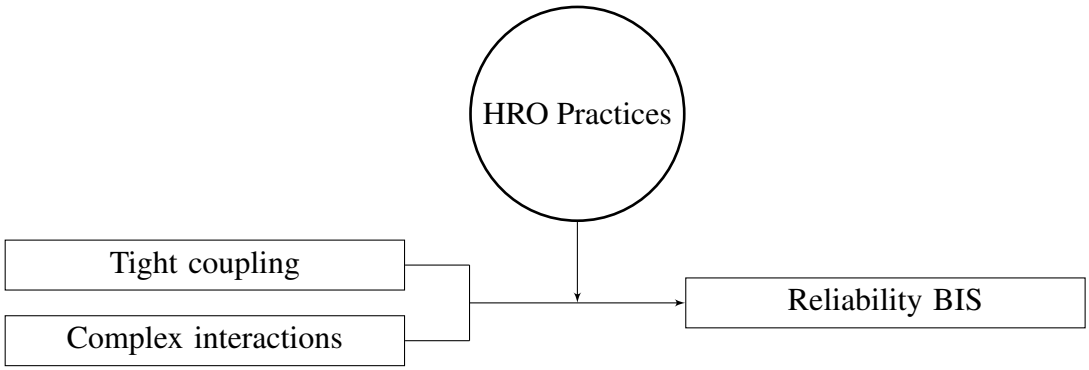

Figure 1: Conceptual model

3 Methodology

3.1 Data collection

This research uses a qualitative approach to obtain the necessary data. The reason to opt for a qualitative approach comes from the good fit it has with explorative research. A quantitative approach using surveys makes it difficult, if not impossible, to follow up on information provided by the participant which is exactly where unknown relationships or new insights might be discovered. These new insights and relationships are the knowledge that this research is trying to discover since its not available in the literature (Cassell & Symon, 2012).

More specifically the expert interview as qualitative research method is used to explore expert knowledge. An expert is defined as "a person who is responsible for the development, implementation or control of solutions/strategies/policies" (Meuser & Nagel, 1991, p. 443). In a BIS context this definition of an expert often equals to computer engineers or more specifically software engineers. These engineers have first of all, a deep understanding of the interaction between the social system and the technological system. This interaction is important for this research given the earlier selected ensemble view of IT in IS research from Orlikowski and Iacono (2001) as well as being important for researching the interactions between system design and organizational practices. Moreover, these engineers are responsible for fixing, mitigating and avoiding failures in the technological system. Finally, they are also responsible for allowing the social system to interact with the technological system in such a way that maximizes user convenience and minimizes accidents. Lay people are intentionally left out since they do not possess the knowledge about developing, implementing or maintaining a BIS. Interviewing lay people on a design theory for BIS thus would not benefit this explorative research.

Meuser and Nagel (2009, p. 31) state that an open interview based on a topic guide (section 3.3) is the appropriate format for conducting expert interviews. The reasoning is that experts reveal more information about relevances connected to their position when they are allowed to give examples and talk freely about their activities. The open interview methodology has a good fit with this behavior since it provides the room for the interviewee to unfold his own outlooks and reflections. The topic guide is included to make sure that the interview stays relevant to the topic researched. Predefined, open questions are allowed but closed questions and a prefixed guideline should be avoided (Cassell & Symon, 2012, p. 248). After agreement to the interview the interviews itself were conducted via video call or in person and recorded with permission

(see section 3.4).

Experts are selected using purposive, typical case sampling. A non-probability sampling technique where participants are chosen on the basis of judgement to provide an illustrative profile that is considered representative, albeit not statistically (Cassell & Symon, 2012, p. 42). Given that expertise is inherently subjective the judgement of the researcher is needed in order to select participants that enable the researcher to answer the research question. Furthermore, since the goal of the research is to develop general hypotheses typical cases are selected, contrary to critical or extreme cases. These experts are approached via email, various social media and forums.

Regarding the sample size for non-probability samples there are no hard and fast rules. In an overview of the literature presented in Cassell and Symon (2012, pp. 45,49) the advised minimum sample size for interviews is between 5 to 25 and will invariably depend upon whether access is granted. For this research a sample size of seven participants is used given that participants are selected from multiple organizations (see table 1 on page 48). Selecting participants from different organizations has the benefit that is allows the researcher to draw conclusions from participants working with different BIS and to guard against selection bias. The downside however is that the sample size is relatively low because approaching these experts, gaining access to different organizational contexts and analyzing the data is time consuming. A sample size of seven different experts that work on seven different BIS allows for enough data to create hypotheses on the researched relationship.

3.2 Data analysis

The data analysis of expert interviews is focused on thematic units. Thematic units are passages with comparable topics that can be found across multiple interviews. Linearity of statements in a single interview is not that important, instead passages gain meaning when analyzed in the organizational context of the expert. Organizational context is gathered from the interviews through themes like organizational operating conditions (see section 3.3) or through specific statements about the respondent's organization or system. Context therefore is taken into account from the beginning of the analysis in order to determine the meaning and significance of the expert's statements. This is needed as commonly shared context allows comparability between the different interviews (Meuser & Nagel, 2009, p. 35).

In order to allow for a systematic analysis of expert interviews Meuser and Nagel (2009)

developed a guideline which is also used in this research to analyze the data. The guideline contains the following steps:

1. Transcription: transcriptions of thematically relevant passages are a prerequisite for the analysis. Prosodic and paralinguistic elements are notated only to a certain extent.

2. Paraphrase: in order to rule out a narrowing of the thematic comparison of passages from the different interviews and to avoid to "give away reality," the paraphrase should follow the unfolding of the conversation and give account of the interviewee's opinions.

3. Coding: the next step in condensing the material is to order the paraphrased passages thematically. The interpreter keeps close to the text and adopts the terminology of the interviewee.

4. Thematic comparison: thematically comparable passages from different interviews are tied together. Category formation close to the language of data has to be maintained and theoretical abstraction should be refrained from.

5. Sociological conceptualization: features shared and features differing from interview to interview are elaborated and categorized by drawing on the theoretical knowledge base.

6. Theoretical generalization: the empirically generalized findings are framed by a theoretically inspired perspective. This is in line with analytic generalization which follows a two-step process. "The first involves a conceptual claim whereby investigators show how their case study findings bear upon a particular theory, theoretical construct, or theoretical (not just actual) sequence of events. The second involves applying the same theory to implicate other, similar situations where analogous events also might occur" (Mills, Durepos, & Wiebe, 2010, p. 20).

(Meuser & Nagel, 2009, p. 36)

3.3 Operationalization

The first topic is an introduction on the specific BIS that the participant is interviewed about. This topic at least encompasses the required function of that BIS and how long that required function should be carried out. Having defined these parts of BIS reliability allows the researcher to define when the BIS did actually fail by arguing that the BIS was not able to carry out the mentioned required function.

A second topic is about the operating conditions in which the BIS functions. Some systems, like communication systems, deal with a lot of human interaction while others, like planning systems, may be much more closed. Furthermore, this topic includes the context of the organization itself. A BIS in a chaotic startup is likely to fail in a different way than a standardized BIS in a hospital.

Having established the ambiguities in the BIS reliability definition, the topic guide can go more in-depth and explore the the relationships shown in the conceptual model. A third topic thus is about investigating whether the BIS of the participant is both complex and tightly coupled. Even though the literature discussed in section 2.2.4 indicates that BIS are inherently complex and tightly coupled, this is not a given. Due to the explorative nature of this research it is important to first establish whether this is actually the case. Vagueness around this topic could compromise this research since Perrow (1984) states that only the combination of complexity and tight coupling leads to normal accidents which in turns influences reliability. Complexity in this context refers to the complexity of interactions while tight coupling refers to the coupling of those interactions (Perrow, 1984).

The fourth topic dives further in the combination of complexity and tight coupling by exploring the influence this combination has on the reliability of the BIS. Again, to stay close to the research of Perrow (1984) these two concepts are taken together in the topic guide.

Finally, the fifth topic is about HRO practices facilitating or hindering the reliability of the BIS. The four HRO practices are individual sub-topics of this main topic and are thus discussed individually. This allows to differentiate between a case where three out of four HRO practices are deployed versus a case where only one or two are deployed. Furthermore, differences between HRO practices might be found.

From this topic guide the operationalization table 2 on page 49 is constructed.

3.4 Research ethics

As defined in the APA's Ethics Code a researcher should follow five principles of research ethics to steer clear of ethical quandaries. Three of these principles apply to the process of collecting data: be conscious of multiple roles, follow informed-consent rules and respect confidentiality & privacy (American Psychological Association, 2017).

During the data collection any relationships that could reasonably impair the professional performance of the participant or could exploit or harm others are avoided. Furthermore, the

researcher is present as a researcher when conducting the interview to distance themselves from the participant.

Secondly, participants are only selected and included when they are participating voluntarily and with full knowledge of relevant risks and benefits. Approached experts not willing to be interviewed are excluded from further contact and experts willing to participate are briefed on the contents of the research. Moreover, participants are free to withdraw from the research at any time and are informed about this. Permission to record the interview is asked before the interview. If permission is not given, permission is asked to make written notes during the interview.

Thirdly, given the sensitive topic of accidents in BIS confidentiality and privacy are of high importance. Participants and the organizations are promised anonymity in any published work. Data regarding personal identifiable information is securely stored and deleted after the research has been finalized. Findings from the research will be shared with all participants (Smith, 2003).

3.5 Reliability and validity

Achieving reliability in a research with open interviews is challenging because each interview is unique in a way. Several measures have been taken in the data collection to increase reliability. First of all, the interview follows a topic guide to ensure that the relevant topics are discussed and that the interview does not divert from what it is intended to research. Secondly, experts are selected based on purposive, typical case sampling. While this does put the researcher in the position where they have to pick the interviewees it does aim to pick the right people for the research. With a random selection results might be less reliable because of the chance to pick non-experts. Thirdly, it is important to recognize the role of the interviewer in the process as "in expert interviewing both the status relation and gender relation play a prominent role" (Meuser & Nagel, 2009). Especially in the context of status relation the results of the interviews is influenced by whether the expert views the researcher as a competent conversational partner. Therefore Meuser and Nagel (2009) instruct that an interviewer should prepare the interview topic thoroughly and build up a knowledge base. In this research the researcher has prepared by investigating the relevant literature and the researcher has familiarity with the subject through practice.

Secondly, the internal validity of this research is high. In an open interview "experts do reveal a lot more about relevances and maxims connected with their positions and functions"

(Meuser & Nagel, 2009). Furthermore, interviewees are asked at the end of the interview if they want to discuss or note other events or insights of interests related to the interview. Therefore, the researched relationships within the context of this research are likely able to rule out alternative explanations. Internal validity is negatively influenced by the use of non-probability sampling which increases the possibility of systematic error. A major systematic error is confounding, which occurs when a non-measured value influences both the independent and dependent variable giving the impression a causal relation exists. To minimize this bias the topic guide does not presume this relationship and further open questions should allow the researcher to uncover unknown mediators and relationships.

Lastly related to the final topic of Meuser and Nagel (2009) guidelines 'theoretical generalization', external validity should be discussed. The concern of generalizability of qualitative research, which has often been critiqued, has been addressed by Yin (2009) in his explanation about analytical generalization as opposed to statistical generalization. "The short answer is that case studies, like experiments, are generalizable to theoretical propositions and not to populations or universes. In this sense, the case study, like the experiment, does not represent a sample, and in doing a case study, your goal will be to expand and generalize theories and not to enumerate frequencies". This is in line with this research which does not have as goal to draw definitive conclusions for the entire population but to explore reliability in BIS.

4 Analysis

Going back to the research question the first thing that must be established whether business information systems have a high degree of complexity and tight coupling. Only if the answer to this question is that they do, the second part of the research question can be answered because NAT assumes a high degree of complexity and tight coupling for normal accidents to occur.

4.1 Complexity

The interviewed experts all perceived the systems they worked on/with as complex. Based on the sociological conceptualization of complexity some similarities and differences arise which are interesting for this research to understand where complexity comes from and how it can potentially be avoided. Based on the similarities and differences the following categories were created in which these similarities or differences are discussed: size of the BIS, issue of communication (within a team or department and within the organization), focus on speed (relating to releasing new features), integrating with other businesses (and their unknowns), architect (and structure) and finally functionality (offered to the users of the BIS). This list may have some overlap because the causes often influence each other, however this was avoided as much as possible.

4.1.1 Size of the BIS

All respondents noted that the systems they worked on were large and that they could not comprehend the system by themselves (1:23, 1:27, 2:2, 3:9, 4:2, 5:1, 6:17, 7:8). A major reason given for this scale and incomprehensibility is the use of dependencies, libraries and code from third parties. Because a lot of the components used in IS come from open source repositories, the members of the organization often haven't created a large part of the system. For example, one of the respondents noted that their project uses four times as much open source code compared to code written by the people responsible for the project. Furthermore, this respondent expected this gap to be even wider for most other projects (3:22). This is further complicated by maintaining different versions or having to upgrade to a new version of such a third-party dependency (1:16). An example given here is that the newest version of a third-party dependency might be incompatible with your system or introduce multiple bugs. Because of this the dependency is simply not updated which might be fine in the short run, however is something

that could cause problems further down the line when the older dependency (which has not been updated for a long time) breaks when new features are introduced. It only works with the older system and thus holds back any new changes made. The technology was simply not made for the future (3:33, 1:4). This problem of unknown and large components integrated in the BIS touches on all three indicators of complexity and most significantly on comprehensibility of interactions. Given that large parts of the system have not been developed by people working on the system in the first place the amount of interactions increases and furthermore there is low comprehensibility of interactions given a lack of context on these parts.

A second reason why these large systems are incomprehensible by a single person is that the respondents noted that different people work on different parts of the application making it a major challenge to notify everyone what is changing or being added to the system (7:8). This problem is similar to the one described above given it relates to all three indicators of complexity where a person can no longer effectively get context on a part of the system. The difference is that here someone in the organization has context while in the previous situation the development is outside of the organization. An exception in this area was respondent 4 who was in charge of a team of five software engineers who build a business-to-business IS. Their organization consisted of five software engineers who were all hired as freelancers making the communication between this small group relatively easy and allowing them to maintain a high level of understanding of what the respondent called a "complex" system. This may at first seem contra-dictionary, while the system is described as complex the people working on the system still have high level of understanding. This can be explained by understanding that respondent four referred to the IT system as complex while the BIS itself has lower complexity because of the relatively low amount of social interactions in the BIS. The business consisted of five software engineers making it easy for them to find the person with context and communicate the complexities arisen in the IT system. This indicates that both the technical size as well as organizational size positively influence complexity.

4.1.2 Issue of communication

This issue of communication also introduces the second similarity. While it may still be easy to communicate in a system with low amount of social interactions this becomes significantly harder in bigger divisions with higher amount of interactions. Apart from respondent 4 (as noted before) not only was the size of the system perceived to increase complexity, the size of

the members of the organization was also perceived to increase complexity. Before continuing on this point, it has to be noted that a large system can have upsides as well. Respondent 7 mentioned that they have an office in Europa and the US allows them to have engineers working on the system around the clock. When a critical issue or bug is found in the system the office wherever it is daytime can quickly fix that failure and allow the company to act quickly. This does however not prevent the failure in the first place. Continuing the original point above, multiple respondents noted it being hard to find the right people in their own division to talk to when trying to discuss a certain part of the system that they were not familiar with. This problem shows close similarity with the problem related to division size in the section above. Respondents noted that talking as technical expert to other parts of the business, most notably to the management layers of the organization, about the course of the information system was often difficult (1:45, 2:7, 2:19, 3:34, 6:13). This relates to comprehensibility of interactions as developers often have a high technical understanding of the system while management has a broader view of the BIS making it difficult to either communicate very narrow, technical specifications or understand the broader view of management. Finally, a person responsible for an older feature within the organization might have left the company by the time that feature needs an update or has a failure. Contacting this person at this point often proves to be difficult and requires the current engineers to learn that feature with whatever information is available to them (3:24).

4.1.3 Focus on speed

Another point that introduces complexity in BIS is the focus on speed or equivalent time pressure. The way most interviewees rationalized this point is that to meet the demands of the customer or to outdo the competition the business has to ship new features that will create customer value. The problem with this focus on new features is given the time pressure that is behind them to release these new features, new features will often not be well tested and contain new bugs and hidden failures that are hard to predict. Thus, in a very simple way it could be said that time pressure and focus on speed to release new features creates a larger system with more potential failures which is less tested. This increases complexity mostly through increasing the amount of unexpected interactions. The amount of interactions increases but disproportionately the amount of unexpected interactions rises as the system is not tested well and therefore not understood well. To release these new features in a more reliable way would require more time

from the engineers for example to write tests or to explore the new feature manually (3:35). This time to focus on reliability and testing new features was described by some engineers, however most agreed that the focus on new features in a fast manner triumphed testing these features for reliability (1:52, 2:10, 5:15, 5:16, 6:29).

Respondent 6 said: "If an information system is not really robust, it doesn't really check me if I am declaring the right input. Suppose there is no good input validation and I have a lot of stress, I have to do a lot of work per hour, then I will do it too quickly and that results in errors. If the system is not robust enough, it can also result in unavailability, if it crashes or the data is no longer correct. Yes, the higher the stress, the more errors you get" (6:29). This statement relates to complexity as the comprehensibility of the system is low. Due to low levels of validation on the input and high level of unexpected interactions the comprehensibility decreases. Secondly, it also relates to tight coupling as introducing new, insufficiently tested features into the system increase the degree of dependability of parts. As one engineer pointed out this focus on shipping new features as fast as possible without paying your technical debt will often lead to a system which works well in the short term but has no real future in the long term. Your system becomes so complex that "you don't have enough time to develop software which works properly most of the time" (2:21). Failures and events that influence the wider system become so common that it becomes hard to change anything in the system at all.

4.1.4 Integrations with other businesses

A fourth cause of complexity was highlighted by multiple engineers to be the integrations with other businesses. While this point mainly touches on tight coupling, where it will also be discussed, it also influences complexity. Because complexity deals with the amount of understanding that can be had over the system, the amount of integrations with other businesses increases the amount of interactions in the system and therefore can make the system harder to understand. Especially when talking about business information systems where a lot of logic resides in code. When talking about integrating with other businesses it means that you are using their systems, APIs and documentation often without having access to the logic that powers that system. In such a situation it will inherently stay fuzzy to engineers from one organization what is happening in a different organization that they integrate with. System behavior can only be tested based on input and output experiences but can rarely be fully understood given that not all information is available. A given example of this is a standard that both organizations

use to integrate with each other however given different interpretations of the documentation the communication between both systems goes wrong (3:3, 5:2, 2:2, 6:2, 6:4). However, respondent 7 noted that the system that they worked on integrated with multiple other businesses without seriously impact reliability. When asked why this impact didn't affect their reliability they mentioned this was because the businesses they integrate with have standardized and well documented communication rules. This resulted in things rarely going wrong or changes happening without realizing it beforehand (e.g. Amazon APIs) (7:12).

4.1.5 Architect and structure

A lack of someone responsible for the design of the BIS, an architect, or the lack of structure in the design of the BIS in general, is perceived by the respondents to increase the complexity of the BIS. A common theme among multiple respondents is that under some of the previously mentioned perceived causes of complexity: time pressure and size of the BIS people tend to focus on their own responsibilities and in doing so significantly lack the overview of the system (1:51, 1:27, 2:15, 2:18, 5:27, 6:17). Respondent 2 describes it in the following way: "not having somebody who is the top level or the architect of the system, however you want to call it, causes the system to be really complex and this is the point where some structure would be needed. To have someone with a high level overview of the entire system, there is nobody who has that or has the high level ownership of the system and this means that we don't work enough on educating the complexity of the system. We focus on smaller pieces of the system instead of focusing on the whole" (2:15). This lack of focus on the system as a whole leads to incomprehensibility of the interactions in the BIS, people understand their own work, but lack the knowledge or context on how their part interacts with other pieces in the BIS.

Respondents 1 and 6 also point out that simply having an architect is not good enough. It is a complex task in itself and when the architect fails to act in favor of the design of the BIS the complexity of the BIS is perceived to increase as well. This point touches less on comprehensibility of the system and more so the amount of interactions in the system. While it is the task of an architect to focus on the design and implementation of the larger system a system with many different interactions makes this more difficult.

Finally, respondent 5 notes that due to a lack of structure it is possible for bugs, which are possible causes for accidents, to resurface in different versions of the BIS. In their example the lack of an application as GitLab, a DevOps lifecycle tool providing wiki and issue-tracking

functionality, in the BIS can result in bug fixes not properly being documented or tracked resulting in those fixes not making it in new releases and therefore resurfacing those issues (5:27). By not documenting and tracking previous knowledge well the comprehensibility of the system lowers, because information about potential failures is not easily accessible leading to resurfacing problems.

4.1.6 Functionality

Finally, a perceived cause of complexity is the amount of functionality that users have within the BIS (2:20, 6:20, 7:2). The relationship between these two can be demonstrated using the example of a closed and open system. If a closed system, for this purpose defined as a system that does not accept any modifications from the user to the system, has an error, the cause of this error can be found relatively easy since the input and the output of the system are predeterministic. If a user requests a static information page they will always get the same page, there is little to no variation or user modification. However in the case of an open system, defined as a system that allows modifications from the user, the input in the system is (partly) unknown and therefore the output is (partly) unknown. For example in a financial system the user might be able to enter different purchase orders with different prices, the BIS takes this information, stores and processes it, to create a final report. This report is dependent on the input and non-deterministic. Any flaws in this process could be dependent on the input of the user, which can be unknown, therefore making the system more complex.

Examples of this relationship were mentioned by Respondent 2 and 6 who noted that their most error prone features were the ones that had the highest amount of user interaction and functionality. For example: “we saw this with the admin functionality, which is so knot together, that even the responsible department does not understand it anymore” (6:20). Furthermore, the decisions around functionality of the BIS can result in complex tradeoffs. “We know that some feature will add to the complexity of the system and make the system more error prone. And these comments are usually disregarded because if a feature adds lots of value then we simply accept that the system will break sometimes” (2:20). Relating this back to theory adding new functionality increases both the amount of interactions and likely the amount of unexpected interactions both of which are indicators of complexity. This further indicates a risk tradeoff between customer value by adding new features versus potentially adding complexity to the system.

4.2 Tightly coupled

Next to complexity Perrow describes the concept coupling which is the amount of "slack, buffer, or give between two items" (Perrow, 1984). As mentioned before systems can be loosely or tightly coupled where tightly coupled systems are centralized and rigid. Loosely coupled systems on the other hand are characterized by decentralized operations and flexible control procedures.

The interviewed experts described their systems as tightly coupled. Similar to the complexity characteristic similarities and differences arise when talking with the interviewee's about the coupling of their system. Based on these similarities and differences the findings have been grouped in three different categories: coupling between internal, technological subsystems, coupling between external, technological subsystems and coupling directly involving humans. These categories fit into a 2x2 matrix:

	Human	Technical
Within the org.	Internally involving humans	Internally between technologies
Outside the org.	Externally involving humans	Externally between technologies

Given the similarities between internal human interactions and external human interactions this category has been grouped together. Based on these categories the indicators: dependability of parts in the BIS and amount of events that influence other events are analyzed to examine the level of coupling in the researched BIS.Some items in the list are also present in the causes for complexity however they are mainly discussed as in how they contribute to the coupling of interactions. A degree of overlap is expected since certain design decisions would be able to both increase the complexity as well as the coupling of a system.

4.2.1 Coupling between internal, technological subsystems

The first way in which a BIS can be seen as a tightly coupled system is when we look at how internal, technological systems can have an effect on each other. In the interviews a respondent gave the following example: "it wouldn't be the first time that a whole cluster went down because the cache synchronization is the problem. We have had this problem multiple times already. Because if you break up your application in smaller parts that means that those parts will have to communicate with the database for which you need some form of synchronization between those parts. If one of those parts is facing problems it can cause so much traffic because

of the synchronization that the others can no longer keep up and just go flat. And that happens quite regularly" (3:28). This example can be further explained using Microsoft's typology of typical application layers consisting of a: user interface (UI), business logic layer (BLL) and data access layer (DAL). "Using this architecture, users make requests through the UI layer, which interacts only with the BLL. The BLL, in turn, can call the DAL for data access requests" (Microsoft, 2019). In a financial BIS the UI could be a website, the BLL would be the code that does the financial calculations that the website displays and finally the DAL would be the database in which the financial data is stored and from which it is retrieved. It is easy to imagine that without one of these layers the entire system fails. A financial report cannot be made without the data, neither without the calculations, nor without the UI for the user to create and view the report from. Here the argument can also be taken further and that is that a failure of a sub-subsystem can also lead to system wide failure. If one calculation in a set of ten that generate the financial report fails it could still be that the entire BLL fails causing the entire application to fail. This indicates that the different application layers of a BIS are highly dependent on each other. If one of the subsystems fails other systems also are negatively influenced or stop working.

This situation is also described by Respondent 1 who had a situation where an underlying library related to database control had to be replaced. This turned out to be impossible without breaking things all over the application given how many times this library was being used in different subsystems (1:16). This quote further demonstrates not only the dependability of parts but also the amount of events that influence other events. Especially in software systems where parts can easily be reused it can occur that a problematic component is used in many different subsystems leading to the possibility of a failure event to cause many other failure events in different systems.

4.2.2 Coupling between external, technological subsystems

Secondly some interviewees described their BIS as tightly coupled when talking about integrations with external partners. Showing that when the technological system of a third party, which is a subsystem in their BIS, fails their entire system can fail as well. "Lets say we got two systems: system A and system B. System A communicates with external entities and the way it communicates has changed. Because of that the external system sends data in a different format than originally received by system A. First of all system A wasn't expecting data in a

different format and because of that it couldn't process the requests. Since system A couldn't process the request, system B couldn't show the bookings of customers because the database system A wasn't responding anymore" (2:1). In this example a combination of a failure in coupling between internal and external technological subsystems is described. First of all, system A communicates with external entities and changes how it communicates. This causes the external entities to send data in a different format which system A wasn't expecting and it fails. Secondly, because system A fails system B, the customer facing booking system, also fails indicating high dependability of parts. It cannot retrieve the bookings from system A and therefore can't show the customer their bookings. Furthermore even small details in a system, like the format of passed around data can lead to the failure of the entire system.

This failure relies on a change in one system, however failures with external subsystems that are tightly coupled to internal subsystems can also occur because of lack of understanding. Imagine two BIS that want to integrate with each other via a certain communication protocol. This protocol is well documented however it is on the developers of both BIS to interpret this documentation and write the implementation. System A sends 'does not apply' for the fields it does not have information about, System B receives the information from System A and parses the field gender for which System A sent 'does not apply'. If system B only expects: male, female, other and unknown system B might fail since it cannot deal with input in the form of 'does not apply' (3:3). The difference here with the example above is that in this implementation no bug or change occurred in both systems but an uncommon feature was differently interpreted by different operators of the system causing the data to not be correctly formatted.

4.2.3 Coupling directly involving humans

Thirdly, a major component of a BIS are humans. In this section the focus will be on how under conditions of tight coupling and complex interactions human failure can results in system wide failure. While the argument can be made that even the technological (sub)systems are created by humans and that the errors in these systems are therefore human errors this is not what the focus of this section is. This is covered in the previous two sections, the aim of this section is that it still applies even if there were hypothetical, bug free, technological systems with perfect integrations.

A type of human interaction where tight coupling can lead to failures in a complex system is related to the capabilities and rights of the users of the system as noted by respondent two

and four. An obvious example is where the employee working with the BIS is either malicious or inexperienced (4:4). This could result in the employee deleting data from the database or modifying the business logic in such a way that the BIS stops functioning (4:5). Respondent two talked about internal users in more detail: "In my company we sell tickets, so the end user would be the user that buys the ticket online. The front end for these users has good validation mechanisms so it is kind of impossible for an end user to destroy something. However another type of user are superusers, internally in the company. For example business analysts. We tend to think of them as internal people of the system but they are end users. Only of a different kind when they modify data, directly on the database or change some business logic with the tools the system gives them they might break the system" (2:28). Combining these two statements it becomes clear that there is a difference in human interaction with the system depending on the rights those people have. The more rights, the more can go wrong. Furthermore, since internal systems are not customer facing they are likely less polished resulting in a higher possibility that human error is not rejected by the system causing invalid information in the system (2:28).

These examples demonstrate the amount of events that influence other events in internal tools are higher than external, customer facing tools even though they operate on the same system. This indicates that internal tools or users are therefore working on a subsystem that is ore tightly coupled than customer facing tools of a BIS. Secondly, the dependability of parts in the BIS is higher in internal tools as they are more powerful and operate on subsystems used in many other subsystems. While a customer might only be able to upload files to their own account an internal user might be able to do this for every user in the BIS.

A second type of human interaction where tight coupling can lead to failures in a complex system is related to miscommunication. An extreme example of miscommunication is unavailability of an important supplier during office hours (6:2) or a milder form where a supplier doesn't have certain goods available which are time sensitive (1:6). While these issues on their own wouldn't result in a normal accident or system failure however under complexity and tight coupling the overview and slack is lacking to make up for these failures leading to a potential system failure. Another type of potential miscommunication is mentioned by respondent 5 explaining that in complex failures where the root cause investigation takes longer communication is critical. The next person doing the night shift has to know what the others investigated during the day and what they bumped in to (5:41). In typical high pressure situations where this is little

slack or give between two items missing this communication can lead to these issues (detected by monitoring) cascading into larger failures.

Other notable examples that respondents talked about were: not informing the end user or customer about an important new feature of the BIS (5:35), stress leading to more errors (6:29) and finding the right people (1:45, 2:7, 3:24, 5:24, 7:7).

4.3 Normal accident occurrences

As examined in the previous sections the interviewed experts perceive their systems to be complex and tightly coupled. Under these conditions NAT would assume that it can be predicted that normal accidents happen. Therefore if the concept of normal accidents is relevant to BIS it is expected to see failures matching characteristics of normal accidents in these BIS given that they operate under complexity and tight coupling. This is important to establish since a relation between BIS design and complexity and tight coupling would indicate that the theory of NAT is relevant to BIS. To that end the following quotes from the experts have been highlighted and compared to failures which "are not only hard or impossible to predict, but also inevitable products in complex and tightly coupled systems" (Müller et al., 2013, p. 3).

- "You want [to replace an underlying library] because otherwise you can't actually handle the entire application and you keep messing around, but on the other hand there are a lot of people who are going to rattle your cage of what's going on here. And you did not add any new functionality and still everything breaks" (1:17).

- "But before the first error we just assume [...], once it reaches our system it malforms because it behaves in the conditions we never thought would occur" (2:3).

- "But the fact remains that mistakes always creep through. Not everything can be covered in a test. [...] Yes, I hadn't thought of that" (3:52).

- "In monitoring we go from failure to failure. If something falls over it can be simple, someone is working on it and then it is fixed again. It can also be more complex, like the chain reaction that we have just mentioned, then it takes longer to find the problem" (5:40).

- "All sorts of complexity that is so tightly tied together, there is so much in it that even the responsible department does not understand it anymore. They work very strongly with a

rollback so they try something and they know they can go back because only then they really find out what effect something has on the rest of the system. All relationships are simply not documented and are unknown" (6:21).

In these examples the characteristics of the normal accident become apparent. Human failure and change become a core issue in systems with high complexity and tight coupling. As noted systems are complex and not fully understood "even the responsible department does not understand it anymore" (6:21) and tightly coupled "[failures] can also be more complex, like [a] chain reaction" (5:40). Furthermore, these conditions of complexity and tight coupling lead to failures: "You want [to replace an underlying library] [...] and you did not add any new functionality and still everything breaks" (1:17) and "all relationships are simply not documented and are unknown" (6:21). These examples indicate that the researched BIS deal with failures similar to normal accidents which are "hard or impossible to predict". From these observations and previous sections it is possible to answer the first research question: "What is the relation between BIS system design and reliability?" with the hypothesis that *complex interactions and tight coupling have a negative influence on BIS reliability*. This relationship is depicted in figure two below.

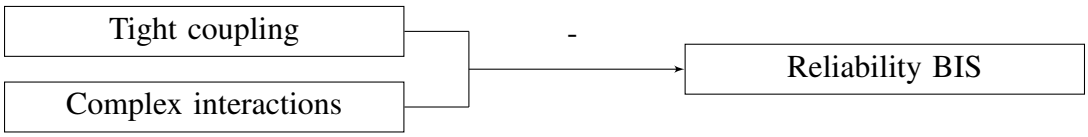

Figure 2: Research question one

4.4 High reliability organization practices

While the previous section shows that complexity and tight coupling have a negative influence on the reliability of a BIS, they are by themselves not sufficient to explain BIS reliability. As Rijpma (1997) and HRT describe, HRO practices influence the effect that complexity and tight coupling have on the reliability of a BIS. Rijpma (1997) describes a 'mixed picture' stating that HRO practices and complexity and tight coupling both influence each other in positive and negative ways, while HRT states that even under complexity and tight coupling organizations can achieve 'outstanding safety records'. Thus simply researching the first research question is not enough since theoretically the moderated relationship can be so strong that it completely

mitigates the effect on the relationship it is operating on. Therefore this chapter contains the analysis of the data in the context of HRO practices. The four practices are discussed below using the data gathered from the interviewed experts giving an insight in whether these practices might apply to complex and tightly coupled BIS and what effect they have.

4.4.1 Decentralized decision making

Decentralized decision making according to HRT leads to increased reliability because operators have the autonomy to respond to emerging problems, as long as these operators have been imbued with centrally determined goals, decision premises and assumptions (Weick, 1987). This logic shows that not only autonomy in decision making is important but also that these decisions are grounded in centrally determined goals and decision premises.

First on the topic of autonomy the respondents two and seven mentioned that they worked using an agile approach where one central principle is: "people over procedures". Based on which respondent two concludes: "so in that sense I would say organizing people in a way where they have lots of freedom is better than writing procedures" (2:17, 7:28). Furthermore, when respondent three was asked if they got the freedom from management to make sure their code was reliable, they mentioned: "definitely, I call the shots in my team" (3:36). Finally, respondent four, the manager of a BIS, mentioned their developers were self-employed and given a lot of freedom to operate (4:10). Respondents noted that in having the ability to make decentralized decisions they are able to speak up against dangerous courses of action. As respondent two describes: if we really shouldn't do it usually management will listen (2:19) and respondent three adds: "in the end I'm the one that says we do this first. We are in the IT security industry so we have to stay up to date and continuously do security audits of our code" (3:37). This seems to imply that decentralized decision making in BIS can decrease the complexity. The engineers and operators closest to the details of the BIS can use their expertise to make informed decisions about how to implement processes, like audits and updates, that decrease the amount of unexpected results.

However, these examples, describing decentralized decision making, do not show centralized design of decision premises. This is important according to Weick (1987) as only this ensures that operates "have been imbued with centrally determined goals". Decentralized decision making without centralized design premises could lead to chaotic situations in which decisions happen all over the place but there is no way in which they are coherent to each other.

A clear example of a centralized decision premise which leads to decentralized decision making is mentioned by respondent three: “In my organization we have a policy that new code needs to be tested which also means there is a continuous effort in writing tests”. So while there is a central premise to test new code it is decentralized decided what kind of tests are written and how extensive these tests are (3:36). However not all respondents described centralized design of decision premises or a culture that can be described as a ‘culture of reliability’. Instead respondent six notes: as a project leader I had to argue to postpone a deadline to operationalize a new system. All of my engineers told me don’t do it. It is walking along the abyss. And yet the CEO told me to go ahead with it (6:32). Looking back at the situation they mentioned: “afterwards I thought it was extremely reckless what happened” and “the chances of it going wrong were really larger than it working however sometimes you need a bit of luck” (6:33). A similar note was mentioned by respondent two who said that the business “usually will hear the engineers” unless it involves a feature that will bring a lot of value “then we simply accept that the system will break sometimes” (2:20). These examples further indicate that not leaving engineers in a position where they make the decisions leads to perceived higher risk of failure. Decentralized decision making therefore can lead to higher levels of tight coupling. Decision premises and assumptions are centralized and therefore influence decision making across the BIS. If these decision premises do not advocate a culture of reliability the effects of those decision premises affect all areas of the BIS.

A common theme amongst the respondents seems to be that related to decision premises reliability is in conflict with new features that generate business value. If reliability is more important than a specific new feature then reliability is prioritized, however if the new feature is that important that it generates more value than the chance of decreasing reliability looses value than the new feature is prioritized.

4.4.2 Redundancy

An interesting difference arises between the difference of non-digital system redundancy and redundancy in the context of a modern BIS. In non-digital systems this often means “physically replacing something or someone and using multiple channels to transmit warnings” (Rochlin, Porte, & Roberts, 1987). While these concepts still apply in a modern BIS the data from this research shows an overwhelming reliance on digital redundancy. The most common example respondents gave is a backup or a rollback mechanism. This means that whenever a new build

has been determined broken or failing the last stable revision is rolled back and used instead until the developers have figured out why the new build is failing. This allows the system to continue operating even when a very complex bug has been introduced that will take days to debug because the live version of the system is running on a previously determined, stable build. For example respondent one noted: "it is also becoming increasingly important, I have noticed, to have a good roll back mechanism. Okay it is hard for us to test this and it is going to take a lot of time, if we just deploy then we can test it in five minutes and if it doesn't work we can always roll back" (1:29). This loosens the coupling of the system as the dependability of parts and the amount of events that influence other events decreases. The introduced backup to the BIS is separate from the existing system and often not influenced by other events allowing the BIS to still function in a failure that would otherwise lead to complete BIS failure.

However, as respondent one and six later point out this mechanism has risks as well. Something in another dependent system might break in those five minutes and cause the rollback to be insufficient or very time consuming (1:29). Furthermore, heavy reliance on rollback can cause the developers to move too fast and not test their product in a thorough way (6:40). This also shows that the rollback mechanism is dependent on complexity. The higher the complexity of the changes in the system the higher the chance that a rollback is not sufficient to unbreak the system. This is in line with the relationship between complexity and redundancy that Rijpma (1997, p. 17) describes: "on the one hand, complex systems are often characterized by redundancy. Complex systems simply need redundancy to keep track of all the possible interactions between the various parts of the system. On the other hand, complexity may reduce the reliability of redundancy. Redundant components sometimes depend on common determinants". In being dependent on common determinants these redundant components themselves can add to the complexity of the BIS by increasing the amount of interactions.

Finally, respondent four and seven also noted the importance of redundancy in hardware. While a rollback is a good mechanism to mitigate the impact of severe software failures in a system it cannot solve a power outage or fire in the datacenter. In this context respondent four described their reliance on an N+1 strategy: "it means that you always have room for one component to fail without causing failure in the entire system. Imagine you have three of something then you can only use 66% of the maximum capacity to guarantee this principle" (4:7). Respondent seven further described that using cloud computing services (like AWS) they can always spawn new applications in different geographical regions even if one of [the

twenty-one] Amazon data centers were to burn down (7:32).

4.4.3 Conceptual slack

As described by Schulman (1993) HROs apply a strategy of conceptual slack to make sure that "complex interactions which might have been overlooked when seen from one perspective are taken into account". While in non-digital systems this is often done by reaching a decision only after intense discussion and negotiation the data in this research suggests that BIS design often comes about or evolves without the use of intense discussion, even to the point of rushed decisions. As seen in previous sections of the analysis: rollback is sometimes more relied upon than testing, operators and developers feel time pressure in releasing new features and management accepts risk of system failure in favor of products that deliver customer value. Furthermore, respondent five mentions: "the one shouting the loudest is often the person that is deemed right" (5:25). While this is a more extreme example in general the data shows that product decisions are made on the team level and direction decisions on the management level (1:32). These decisions do involve discussion but none of the respondents described what can be called 'intense' discussions.

Instead what the data in this research shows is reliance on testing to make sure that an oversight or missed perspective does not lead to a failure. A large part of these tests fall under automated testing which often tests a piece of code or functionality. These tests check if a predefined input leads to the expected output defined in the test, if these mismatch the test will fail. If the change in output is expected the test can be updated to account for the new use case, if the change is unexpected you fix your system so the test passes. This is further described by respondent three: "a test suite can never proof that your system doesn't contain bugs. However a test suite can proof that under normal usage the standard [information] routes in your application do what they are supposed to do. Furthermore, you can also be creative and write tests for some corner case that you might have thought of" (3:50). This means that while tests in complex systems will never be able to cover the whole system they do cover the most common flows. Since the common flows are likely the most critical for the BIS this also means that tests automatically prioritize system failure over failure in a less tested part of the BIS. Therefore tests decrease the complexity of a BIS through decreasing the amount of unexpected interactions and increasing the comprehensibility the BIS. Failures that would otherwise be unexpected might be caught by a test making sure that every new piece of code doesn't break existing functionality. Secondly,

they can loosen the coupling of a BIS by decreasing the amount of events that influence other events. A test can stop a critical failure from propagating elsewhere in the system or raising alarms for manual review after which further escalations can be prevented.

However automated testing can also lead to increased complexity through an increased amount of interactions. As respondent 1 notes: "in that case you fallback to functional tests however those have the disadvantage that they rely on dependencies because at the core of the system is a whole layer that does the provisioning and below that is an operating layer. And all of those have to work and then you get a whole chain of things that all need tests in an entirely different environment than production" (1:20). This example shows that tests sometimes have to rely on other parts of the system therefore increasing the amount of interactions in the system.

Besides automated testing respondent five also noted their BIS used used user tests to gain insight in other perspectives that they might have overlooked themselves. "Customers sometimes make comments that you never thought of. For example someone says: I would have liked to combine your product with the one from the competitor so I would have been able to watch certain sports channels. But wait a minute, we also have a plus package. We entirely forgot to inform the customer that we also offer this service" (5:34). This example nicely shows that BIS can fail in many ways that may not be as obvious as technical failures or crashes. Forgetting to see certain interactions or pieces of information for the perspective of your customer can lead the BIS to fail in ways that might be hard to notice by the organization itself. Here strategies applying conceptual slack, like user tests, can prevent these failures. These user tests can increase the comprehensibility of the system by showing an outsider view that operators and designers did not think of, therefore lowering the complexity of the system.

4.4.4 Constant training

Finally, the fourth HRO practice describes that HROs have "accomplished their extremely reliable performances only after a long [...] trial- and-error learning process [which] is supplemented by constant training" (Rijpma, 1997). Out of all HRO practices the respondents, especially engineers, seemed to practice constant training the least. Often noting that training for engineers wasn't done from an organizational level but that individual employees are responsible for staying up to date on their systems and best practices in (software) system design. One of the reasons that trainings do not seem widely used is best explained by respondent three: "I do not really believe in training with the goal of figuring out the latest developments. Some-

one that gives the training, first has to figure out how it works and develop the training which takes at least half a year" (3:39) other respondents noted similar concerns (1:33, 4:11, 5:33, 7:11). While most respondents do not participate in these kind of trainings it can be said that engineers in general train their understanding of the system by simply coding in the system. In this context respondent two describes: "so if somebody would train me in how the system internally works it might give me some high level overview but it would never be enough or replace things like good old fashioned coding in the system" (2:24). Given that engineers write new features, replace old ones or refactor parts of the system over time doing these practices can give a detailed insight in what the system is supposed to do. A minor note here is that given complexity the level of understanding can vary however it logically follows that working on the logic of the BIS increases one's understanding of what the BIS does. So, while it can be said there is little formal training there is constant training via constant development of the system which involves a "long, trying, costly and, sometimes, lethal trial-and-error learning process" that Rijpma (1997) describes in this context. As seen in previous chapters debugging certain failures can definitely be describes as a long and costly trial-and-error process from which engineers learn and gain a better understanding of the system. This better understanding of the system decreases the complexity of the BIS through increased comprehensibility leading to a lower probability of normal accidents.

While the respondents mentioned that engineers often don't formally engage in trainings they also noted that trainings can serve a valuable purpose in other domains. A good example that multiple respondents gave is to make sure that the users of your system receive training and are aware of the latest updates of your system (2:31, 6:26, 7:10). "Users" in this context is not applicable to previously referenced external users, since "they don't know how to use the system and they shouldn't know how to use the system. The UI should be as simple as possible", but more to internal users of your system. "You think you have control over internal users, you can do training with them. They have more sophisticated tools they need to do more things in the system and thats where it breaks" (2:31). In this example internal users are close to what Rijpma (1997) would describe as operators. They are not the people that designed the system or machine themselves but they are the ones that use it and have access to powerful functionality which can break the system. Given that they are, contrary to engineers, not the creators or builders of the system training, the data from this research suggests that BIS use constant training to ensure that these operators stay informed about the constant changes that

engineers introduced in the system. Similar to the trial-and-error debugging training this type of training can lower the complexity of the BIS by giving operators higher comprehensibility of the system therefore decreasing the chances of failures. If an operator is better aware of the functionality and dangers in a BIS, like operating directly on the database, the chances of failure decrease as these systems and tooling are better understood leading to lower mistakes.

4.5 Influence of highly reliable practices on BIS design

As described in section 2.2.3 Rijpma (1997) describes a mixed effect on how HRO practices affect the potential for normal accidents. “On the one hand, redundancy increases the amount of information generated; the anticipation of a higher number of complex interactions is improved when conceptual slack is maintained; and learning may reduce the level of complexity. On the other hand, redundancy increases the level of complexity by inducing ambiguity, opaqueness and the occurrence of simultaneous failures; conceptual slack may create confusion; and, finally, decision premises increase the level of tight coupling” (Rijpma, 1997, p. 21).

From the data in this research a similar, mixed effect of how HRO practices affect the potential for normal accidents appears. Operators and engineers use redundancy practices like N+1, software backups and cloud software to make sure that if one component fails another takes it place. Failures that would otherwise lead to complete system failures can often quickly be avoided by replacing the system with a stable, older version or different hardware that is not yet affected by the failure. Secondly, through decentralized decision practices the complexity of the BIS can decrease. By having people closest to the failures make the decisions better strategies and procedures against failures can be put in place. It is crucial that this is combined with centralized decision premises and goals creating a culture of reliability as otherwise risky decisions and bets against failures can nullify these practices (6:33). In the third place, engineers practice conceptual slack through (automated) tests decreasing the amount of unexpected interactions in the BIS by automatically testing old functionality against new features and alerting when things break. Finally, through training and debugging the comprehensibility of the BIS can increase leading to lower complexity.

However, HRO practices can also increase the potential for normal accidents. Decentralized decision making through decision premises can lead to increased levels of tight coupling as these centralized premises and assumptions influence business decisions everywhere in the BIS. Secondly, redundancy can increase the levels of complexity in the system through adding

additional layers and fallback systems. These fallback and N+1 systems themselves can fail or not be reliable because of common determinants with the main system adding further to the amount of interactions in the system. Finally, practicing conceptual slack through testing may increase complexity through introducing new interactions in the system. Furthermore, tests can give a false sense of reliability since not all interactions in the system can be tested. The more complex the system the harder it is to write automated tests for it.

While the data shows a mixed picture this does not mean that there is neither an overall positive or negative effect. Overall the respondents noted the different HRO practices as positive for increasing the reliability of the BIS. Furthermore, if this were not the case one could argue that automated testing or rollback system altogether would disappear. Contrary however respondents have noted that the usage of these practices has only been increasing and therefore this can be taken as an indicator for their positive effect on reliability. Based on this analysis the following hypothesis is formulated: *HRO practices negatively influence the direct relation of complex interactions and tight coupling on the reliability of a BIS*.

5 Conclusion

This thesis set out to explore if normal accident theory (NAT) can be used to evaluate the proneness of a business information system (BIS) to normal accidents and secondly explore how high reliability organization (HRO) practices influence the level of proneness to normal accidents.

To conclude, this research indicates that NAT is relevant in the context of BIS and can be used to evaluate the proneness of a BIS to normal accidents. It is hypothesized that *complex interactions and tight coupling have a negative influence on BIS reliability*. Increased size of the BIS may increase complexity through higher amount of interactions between people and code in the system. A lack of structure or efficient communication can decrease the comprehensibility of the BIS and a high focus on speed or integration with other businesses can lead to an increased amount of unexpected interactions. Tight coupling can occur because of how internal and external technological subsystems are coupled together. Multiple layers of the system are often dependent on each other and one layer failing can result in the entire BIS failing. Furthermore, human interactions in the BIS can increase the level of tight coupling through misuse of internal tooling and miscommunication. When tight coupling and complexity are both present in a BIS this research indicates they can lead to normal accidents. Respondents described multiple failures in complex and tightly coupled systems that were hard or impossible to predict and showed complex chains of events.

Secondly, this research indicates a mixed influence of HRO practices on the potential for normal accidents and the negative influence on the potential for normal accidents is hypothesized to be larger than the positive influence, therefore *HRO practices negatively influence the direct relation of complex interactions and tight coupling on the reliability of a BIS*. On the one hand, decentralized decision making can decrease complexity by having experts make the decisions closest to the problems. Redundancy, loosens the coupling of the system by introducing backup systems that can contain failures. Conceptual slack can decrease the amount of unexpected interactions and constant training increase the comprehensibility of the system. On the other hand, redundancy and conceptual slack can increase the complexity through adding new (unexpected) interactions while centralized design premises can increase the level of tight coupling.

6 Discussion

The results, as summarized in the conclusion, indicate that in the first place traditional research into system reliability could be useful when applied to newer systems centralized around IT. NAT and HRT are both theories partly based on large, nuclear system failures where systems are hardware based, have large involvement of operators and "mostly just sit there" (Perrow, 1984, p. 13). On the other hand, BIS are focused on speed, have high involvement of system designers, called engineers, and are heavily based on software. Despite these differences this research indicates that design parameters like complexity and tight coupling do influence BIS reliability and HRO practices do influence the probability of normal accidents. Furthermore, these results indicate that the combination of complexity and tight coupling in a system decrease reliability and increase the probability of normal accidents even though a mixed effect of these design parameters has been found through the increased need for HRO practices. Indicating that adding extra features to a BIS, incorporating open source components and hiring more employees in an environment where application layers influence each other and integrations with subsystems are crucial lead to lower reliability. Secondly, this research hypothesizes that HRO practices have a negative effect on the relationship between complexity and tight coupling on BIS reliability. This suggests that having backup strategies, training your system operators, engineers debugging problems and writing automated tests do increase the reliability of the BIS through decreasing the effect of the previous described relationship.

These interpretations share similarities with organizational mindfulness theory applied to BIS by Butler and Gray (2006). They state that response teams and ad hoc crisis resolution are a range of structures and practices for managing day-to-day failures and catastrophes that arise when working with complex BIS. Organizational mindfulness theory implies that these structures and practices underlie a firm's ability to make effective use information technologies (Butler & Gray, 2006, p. 217). Our research indicates similarly that HRO practices, which can take the form of response teams through redundancy and ad hoc crisis resolution through decentralized decision making, is essential for effectively using BIS.

Given the little research previously on work practices and structures to make reliable BIS possible, this explorative research is significant. While it does not draw definitive conclusion it contributes hypotheses that can be used for further research to draw on and as early indicator for possible relationships. Furthermore, research in the area of BIS reliability is important. BIS are responsible for people's life and safety (The Independent, 2018) and "companies are

increasingly looking to technology to drive their revenue" (Bureau of Labor Statistics Labor Deptartment, 2008). First of all this shows that BIS are a large part of society and growing. Secondly, it shows that they have catastrophic potential, similar to the system describes by Perrow (1984), "the ability to take the lives of hundreds of people in one blow". Although in a BIS it is likely not a blow but change in "a single bit of information (whether in a program or data) [which] can have devastating effects" (MacKenzie, 1994, p. 245). This research contributes to a larger body of knowledge, indicating possible new relationships, which aims to increase our understanding on how the design of these systems influence its reliability.

Because of the explorative nature of this research these results should only be taken as hypothesis and should not be used to draw definitive conclusions. As previously mentioned in the methodology section these results are not generalizable to an entire population but to theoretical propositions. Therefore based on this research it should not be said that BIS reliability is influenced by complexity, tight coupling or HRO practices. It also should not be said based on this research in which direction the relationship is influenced. It can only be said that the findings are indications for these relationships and handles for future research. Furthermore, since all respondents noted that their systems were complex to work with little analysis has been spend on BIS that do not have complex or tightly coupled interactions.

Our academic recommendations based on these results, research state and limitations are for future research to use the hypothesis generated by this research and explore these relations further. Both qualitative and quantitative research is needed about the, positive and negative, effects of complex interactions and tight coupling on BIS reliability. Quantitative research could test the hypotheses by conducting a more deductive approach on a large set of data to see if the formulated assumptions still show. For example complexity (e.g. number lines of code in a BIS) and tightly coupled (e.g. number of estimated dependencies) interactions could be made measurable to quantitatively link them to number of accidents in a BIS. Furthermore, qualitative research could contribute by researching larger failures in a case study, similar to the Three Mile Island accident analysis. This could research the hypotheses in greater detail by leveraging a larger and more detailed case.

Finally our practical recommendation for organizations using BIS are to take these results as early indicators that the design of a BIS influences the reliability of the system. Moreover, they suggest that when increasing the complexity and/or tight coupling of the BIS a tradeoff should be considered between increasing functionality, capability or other desired effects and

the negative effect of these changes on the reliability of the system. Lastly, these results indicate that if a BIS has a high degree of complexity and tight coupling the organization should evaluate the level of which the HRO practices are utilized and where improvements can be made in order to increase reliability.

7 References

Alter, S. L. (1976). How effective managers use information systems. *Harvard Business Review*, *54*(6), 97–104.

American Psychological Association. (2017). *Ethical Principles of Psychologists and Code of Conduct.* Retrieved from `https://www.apa.org/ethics/code/`

Beynon-Davies, P. (1999). Human error and information systems failure: the case of the london ambulance service computer-aided despatch system project. *Interacting with Computers*, *11*(6), 699–720.

Beynon-Davies, P. (2004). *Database systems*. Springer.

Beynon-Davies, P. (2013). *Business information systems*. Macmillan International Higher Education.

Bureau of Labor Statistics Labor Deptartment. (2008). *Occupational outlook handbook*. Bureau of Labor Statistics.

Burke, D. (1995). All circuits are busy now: The 1990 at&t long distance network collapse. *California Polytechnic State University*.

Butler, B. S., & Gray, P. H. (2006). Reliability, mindfulness, and information systems. *MIS Quarterly*, *30*(2), 211–224.

Cassell, C., & Symon, G. (2012). *Qualitative organizational research: core methods and current challenges*. Sage.

Dörfler, I., & Baumann, O. (2014). Learning from a drastic failure: the case of the airbus a380 program. *Industry and Innovation*, *21*(3), 197–214.

Lee, A. S. (2001). Editorial. *MIS Quarterly*, *25*(1), 3–7.

Lekka, C. (2011). *High reliability organisations: A review of the literature.* Retrieved from `http://www.hse.gov.uk/research/rrpdf/rr899.pdf`

Leveson, N. G. (1986, June). Software safety: Why, what, and how. *ACM Comput. Surv.*, *18*(2), 125–163. doi: 10.1145/7474.7528

MacKenzie, D. (1994). Computer-related accidental death: an empirical exploration. *Science and Public Policy*, *21*(4), 233–248.

Mansour, O., & Ghazawneh, A. (2009, 01). Research in information systems: Implications of the constant changing nature of IT in the social computing era. In *32nd information systems research seminar in scandinavia.* IRIS32.

March, J., & Cyert, R. M. (1992). *A behavioral theory of the firm* (Vol. 2). Blackwell Business.

Meuser, M., & Nagel, U. (1991). Expertinneninterviews - vielfach erprobt, wenig bedacht : ein beitrag zur qualitativen methodendiskussion. In D. Garz & K. Kraimer (Eds.), (pp. 441–471). Westdt. Verl.

Meuser, M., & Nagel, U. (2009). The expert interview and changes in knowledge production. In A. Bogner, B. Littig, & W. Menz (Eds.), *Interviewing experts* (pp. 17–42). London: Palgrave Macmillan UK.

Microsoft. (2019). *Common web application architectures.* Retrieved 2019-05-12, from `https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/common-web-application-architectures`

Mills, A., Durepos, G., & Wiebe, E. (2010). *Encyclopedia of case study research*. SAGE.

Moore, G. E. (2006). Progress in digital integrated electronics [technical literaiture, copyright 1975 ieee. reprinted, with permission. technical digest. international electron devices meeting, ieee, 1975, pp. 11-13.]. *IEEE Solid-State Circuits Society Newsletter*, *20*(3).

Müller, G., Koslowski, T., & Accorsi, R. (2013). Resilience - a new research field in business information systems? *Business Information Systems Workshops*, *160*, 3–14.

Orlikowski, W. J., & Iacono, S. (2001). Research commentary: Desperately seeking the "IT" in IT research – a call to theorizing the it artifact. *Information Systems Research*, *12*(2), 121–134.

Perrow, C. (1984). *Normal accidents: Living with high risk technologies*. Basic Books.

Rijpma, J. A. (1997). Complexity, tight-coupling and reliability: Connecting normal accidents theory and high reliability theory. *Journal of Contingencies and Crisis Management*, *5*(1), 15–23.

Rochlin, G. I., Porte, T. R. L., & Roberts, K. H. (1987). The self-designing high-reliability organization: Aircraft carrier flight operations at sea. *Naval War College Review*, *40*(4), 76–90.

Schulman, P. (1993). The negotiated order of organizational reliability. *Administration and Society*, *25*(3), 353-372.

Silver, M. S., Markus, M. L., & Beath, C. M. (1995). The information technology interaction model: A foundation for the mba core course. *MIS quarterly*, 361–390.

Smith, D. (2003). *Five principles for research ethics: cover your bases with these ethical strategies.* Retrieved from `https://www.apa.org/monitor/jan03/principles.aspx`

The Independent. (2018). *NHS computer problems could be to blame for 'hundreds of deaths', academics claim.* Retrieved from `http://www.independent.co.uk/news/health/nhs-computer-problems-blame-hundreds-deaths-harold-thimbleby-martyn-thomas-gresham-college-a8197986.html`

Truex, D., Baskerville, R., & Klein, H. (1999, 08). Growing systems in emergent organizations. *Communications of the ACM*, *42*, 117-123. doi: 10.1145/310930.310984

United States Nuclear Regulatory Commission. (2013). *Backgrounder on the Three Mile Island Accident.* Retrieved from `https://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html`

Weick, K. E. (1987). Organizational culture as a source of high reliability. *California Management Review*, *29*(2), 112–127.

Whitney, D. E. (2003). *"normal accidents" by charles perrow.* Massachusetts Institute of Technology Engineering Systems Division.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition*. Ablex Publishing.

Yin, R. K. (2009). *Case study research: Design and methods*. SAGE.

8 Attachments

8.1 Respondents table

Respondent	Type of organization	Type of system
Respondent 1	Internet service provider	Network management between different systems
Respondent 2	Travel agency	Sell travel tickets
Respondent 3	Online education	Access management & security
Respondent 4	Enterprise resource planning	Enterprise resource planning
Respondent 5	Communciation provider	Legal intercept
Respondent 6	Consulting	(multiple)
Respondent 7	Crowd source	Crowd source platform

Table 1: Respondents table

8.2 Operationalization table

Concepts	Dimensions	Indicators	Topic
Type of BIS	Required function	The objective of the BIS	1
		The requirements of the BIS	1
	Organizational operating conditions	The (cost) constraints of the BIS	2
		Amount of people maintaining/developing the BIS.	2
		Amount of people working with the BIS in the organization.	2
		Organization culture	2
		Type of Organization	2
	Environmental operating conditions	Amount of people working with the BIS outside of the organization.	2
		Amount of user interaction with the BIS.	2
Proneness to normal accidents	Complexity	Amount of interactions in the system.	3
		Comprehensibility of interactions.	3
		Amount of unexpected interactions.	3
	Tight coupling	Dependability of parts in the BIS.	3
		Amount of events/failures that influence other events.	3
BIS reliability	Performing intended function	Amount of times required function was not met (system accident).	4
		(Amount of) changes in the required function	4
		Amount of trust in BIS	4
	Operating conditions	(Amount of) changes in the operating conditions	4
HRO practices	Decentralized decision making	Centralized design of decision premises.	5a
		Amount of process involvement of the engineers.	5a
	Redundancy	Amount of single points of failure.	5b
		Presence of backup strategies if BIS fails.	5b
	Conceptual Slack	Thoroughness of decisions	5c
		Number of previously detected blind spots.	5c
		Amount of work stress	5c
	Constant training	Number of trainings/workshops	5d
		Amount of understanding complexity in own system.	5d

Table 2: Operationalization table