The viability of post-graduate students developing a commercial ERP system of industry standards: A South African Case Study

J. F. Du Plessis

orcid.org/placeholder

Thesis submitted for the degree *Magister Scientiae* in Computer Science at the North-West University

Supervisor: Prof. C. J. Kruger

Graduation: 202?

Student number: 33642958

Acknowledgements

I would like to thank the everyone for helping me. You’re all great!

Abstract

This is the abstract. It is a summary of all the work.

Keywords: Keyword 1; Keyword 2; Keyword 3.

Contents

1 Introduction **1**

1.1 Problem Statement 2

1.2 Research Aims and Objectives 2

1.3 Hypothesis 4

1.4 Methods of Investigation 4

1.4.1 Literature Study 4

1.4.2 Methods of Investigation 5

1.5 Chapter Division 5

2 Research Design **8**

2.1 Exploring Research Methods using the Research Onion 8

2.2 Research Philosophy 11

2.2.1 Positivism 11

2.2.2 Critical Realism 12

2.2.3 Interpretivism 13

2.2.4 Pragmatism 14

2.2.5 Conclusion 14

2.3 Research Approach 16

2.3.1 Deduction 16

2.3.2 Abduction 17

2.3.3 Induction . . . 18
2.4 Research Strategy . . . 19
2.4.1 Design Science . . . 19
2.4.2 Action Research . . . 22
2.4.3 Action Design Research . . . 23
2.4.4 Experiment . . . 26
2.4.5 Survey . . . 27
2.4.6 Case Study . . . 27
2.4.7 Conclusion . . . 27
2.5 Research Design . . . 28
2.5.1 Data Collection . . . 28
2.5.2 Ethical Considerations . . . 28
2.6 Conclusion . . . 28

3 Literature Study **29**

3.1 ERP Systems . . . 30
3.1.1 Features of ERP Systems . . . 31
3.1.2 Competition within the ERP System Industry . . . 38
3.1.3 Components of Developing an Industry Approved Software System . 41
3.1.4 Components of Developing an Industry Approved ERP System . . . 43
3.2 Implementation of ERP Systems . . . 46
3.2.1 ERP Implementation Time . . . 47
3.3 The Information Technology Industry . . . 47
3.3.1 Challenges within the Information Technology Industry . . . 48
3.3.2 Challenges Exclusive to Developing Nations . . . 48
3.3.3 Recommendations for Developing Nations . . . 49
3.4 The Value Chain . . . 50

3.4.1 People . . . 50
3.4.2 Processes . . . 50
3.4.3 Technology . . . 51
3.5 Commercialization of Software . . . 52
3.5.1 Methods of Commercialization . . . 52
3.6 Students in Industry . . . 52
3.6.1 The Purpose of a University . . . 52
3.6.2 The Economic Viability of Academic and Industry Partnership . . . 52
3.6.3 Industry Offering Resources to Universities . . . 53
3.6.4 Critical Competencies for Software Development . . . 53
3.6.5 Advantages . . . 56
3.6.6 Threats . . . 56
3.6.7 Similar Projects . . . 56
3.7 Project Management Methodologies . . . 56
3.7.1 Software Development Management Methodologies . . . 56
3.7.2 Recommendations . . . 56
3.8 Artefact . . . 56
3.8.1 Methods for Measuring Artefact Success . . . 56
3.8.2 Methods for Measuring Artefact Functionality . . . 56
3.8.3 TAM Model . . . 56
3.9 Conclusion . . . 56

4 Case Study **57**

4.1 University . . . 58
4.1.1 Academic Deliverables . . . 58
4.2 Industry . . . 58
4.3 Implementation of Project Management Methodology . . . 58

4.4 Student Team and Roles 58

4.5 Project Design 58

4.5.1 Requirements Gathering 58

4.5.2 System Requirements Definition 58

4.6 Project Pipeline and Process 58

4.6.1 Project Milestones 58

4.6.2 Project Plan 58

5 Expert Reviewer **59**

6 Results and Discussion **60**

6.1 Recommendations 60

7 Conclusion **61**

List of Figures

2.1 The Saunders Research Onion 9

2.2 The Modified Research Onion 10

2.3 Comparison of four research philosophies in management research 15

2.4 Design science research process (DSRP) model 20

2.5 The ADR Method: Stages and Principles 24

3.1 ERP System Framework Hierarchy 31

3.2 Human resource management system optimization rate 33

3.3 Flow chart of the company's production and inventory business based on ERP system 35

3.4 Flow chart of the company's ERP-based procurement and payment process 36

List of Tables

2.1 The possible research design composition in IS research 10

3.1 Comparison of SAP vs. Oracle vs. Microsoft Dynamics 39

3.2 Comparison of Monday.com, Zoho, and Taskflow 40

3.3 Essential Competencies 55

Table of abbreviations

A table containing a list of abbreviations that will be used throughout text.

ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
DSR	Design Science Research
IS	Information Science
CR	Critical Realism
ADR	Action Design Research
QA	Quality Assurance
IT	Information Technology

Chapter 1

Introduction

The demand for efficient and adaptable enterprise solutions has never been greater in today's rapidly evolving technological landscape. Within the South African context, where the intersection of academia and industry holds significant promise for innovation and economic growth, the exploration of approaches to software development becomes imperative. This introduction outlines the rationale, objectives, and structure of the research study aimed at investigating the feasibility and efficacy of leveraging university students to develop a commercial-grade software solution in the form of an Enterprise Resource Planning (ERP) system in collaboration with TaskFlow a partnering North-West University-industry software development company.

Collaboration between industry and academia is essential to benefit the global technological village (Baig et al., 2018). Unfortunately, universities, especially in developing nations, often lack the resources to allow for the proper development of skills needed by students entering the software development workforce. Through collaborative initiatives, software companies possess the necessary skills, know-how, and technology to improve the skills of university students in a controlled environment that simulates real-world experience. For instance, companies specialising in software development, like TaskFlow, integrate solutions such as Contact Centre, Customer Relations Management, Dialer, and Helpdesk into a comprehensive Enterprise Resource Planning (ERP) suite. This empowers their clients to optimise time and make informed business decisions through tailor-made and integrated software that more efficiently and effectively streamlines their entire business processes.

Integrated ERP systems significantly affect the organisation's value chain in both primary and secondary operations (Amini and Abukari, 2020). By connecting business and management activities, enterprise resource planning (ERP) assists organisations in realising their full potential (Uçaktürk and Villard, 2013). Additionally, while ERP systems are pivotal to operational success, the associated expenses of non-integrated software solutions present significant barriers to cost and scalability. Consequently, many software-developing companies aim to minimise the cost of training interns on the theory behind and use of integrated software tools, project management, and software development, especially when developing complex systems such as ERP suits. By developing an industry-approved software system, this research investigates how the North-West University in South Africa utilized computer science and information systems graduate students to address challenges in the South African IT industry through collaborative university/industry-applied project management and systems development methodologies. By doing so, the study explores the viability of creating a mutually beneficial relationship between academia and industry, which can bring positive outcomes for both parties and students. As such, the research will delve into the potential of the curricula in Project management and System development at Honor's level to mitigate the challenges faced by the industry while nurturing a dynamic and symbiotic partnership between academia and industry.

1.1 Problem Statement

The intersection of academia and industry presents unique opportunities and formidable challenges. By the late 1990s, Foley (1997) argued that it is beneficial for graduate students to comprehend how the industry operates and how ideas and innovations are turned into the products and services society requires. As employers found it challenging to locate people with the knowledge and abilities needed for open positions, President Barack Obama emphasised the need for lawmakers and national higher education associations to urge educational institutions to not only set ambitious educational attainment targets but also ensure that their goals address the current gap in required industry skills (Barnett, 2011). Efficient learning and skill development in an educational setting have become increasingly important to ensure that individuals acquire the necessary expertise for success in various industries (Baig et al., 2018).

The central inquiry for this study emerges from the argument that tertiary education students need to be trained for the requirements of a professional workspace (Baig et al., 2018). The question arises: *To what extent can postgraduate students be effectively taught to develop industry-standard software solutions such as ERP systems amidst the multidimensional challenges of technical proficiency, client satisfaction, and academic rigour?*

1.2 Research Aims and Objectives

By developing an industry-approved software system, this research investigates how the North-West University in South Africa utilized computer science and information systems post-graduate students to address challenges in the South African IT industry through collaborative university/industry-applied project management and systems development methodologies. To answer the aim of the study, the first objective would be to define the essential components and prerequisites of a fully functional, versatile, and industry-approved software system. Secondly to understand the challenges the IT industry faces, especially in developing nations. Thirdly, to investigate the critical project management and system development competencies and capabilities needed to create industry-standard software solutions and to what extent Higher Education Institutions (HEIs), especially in developing nations, address these requirements in their post-graduate computer science and Information curricula. And finally, how the industry can collaborate to transfer the necessary skills to HEIs learning processes to enable students to develop industry-approved software systems.

This study aims to evaluate the feasibility of creating a commercially viable software solution in collaboration with a software development company as part of the NWU computer science and information systems department's prescribed curriculum on IT project management and software development honours program. Through collaboration with the industry, this study aims to showcase the technical abilities of computer science students by developing a software solution that meets commercial standards. This initiative also seeks to equip students with the necessary skills to become valuable contributors to the South African workforce. The project aims to conduct a pilot study to identify the key success factors in guiding postgraduate students at a Higher Education Institution (HEI) in an underdeveloped country. The goal is to provide these students with industry exposure in a formalised academic and learning environment, enabling them to develop economically viable software. The study seeks to determine how the students can be effectively mentored and trained and what support and resources are necessary to ensure their success. Ultimately, the findings of this study will be used to inform the design of future programs that seek to foster innovation and economic growth in underdeveloped regions of the world.

This study explores the possibility of integrating projects and software development techniques taught in an educational setting by teaching existing structures and practices of a typical software development company in the South African industry. The objective is, therefore, to teach computer science and information systems graduates to identify market opportunities and, by applying industry practices and sound PM and SD methods, to develop software products that are commercially viable and aligned with market demands. As such, the research seeks to uncover optimal practices and tactics that improve educational insights regarding the teaching of PM and software development procedures, integrating cutting-edge best practices from the industry. To assess the software solution's viability as a marketable product, the research will conclude through an extensive evaluation by proficient experts and professionals in the software development industry. It will also involve thoroughly examining the client's satisfaction and feedback after receiving the final system, including any concerns they may raise. This assessment will determine the critical success factors in developing commercial software, the roles of industry and academia in the learning process, and any necessary improvements to enhance the learning experience.

1.3 Hypothesis

Postgraduate computer science and information systems students can build a sophisticated software solution that is commercially viable, such as an ERP system when partnering with the expertise and resources of industry partners.

1.4 Methods of Investigation

Throughout the commencement of the research, the study will draw upon two main facets to substantiate the claims made in the paper's conclusions: the vast amount of academic literacy available and the other experiment based data collection processes referring to the development project as the main experiment. Both of these will be outlined below to provide clarity on both matters.

1.4.1 Literature Study

To enhance the credibility, accuracy, and validity of the study, a thorough and systematic review of pertinent literature will be conducted. This will involve targeting keywords such as higher education, academia, ERP systems, software success, project management, system implementation, and commercialisation. Research will be prioritised from a positivist perspective and examine peer-reviewed articles from reliable sources such as the NWU Library, Google Scholar, IEEE Xplore, and ACM Digital Library. In addition, grey literature from trusted sources like white papers and industry reports will be considered, as this research incorporates best practices from the industry. Duplicate articles or those not meeting our eligibility criteria will not be accepted. To locate important material, references from key papers will be scrutinised and analysed for citation patterns, thus seeking advice and insights from subject matter experts. The focus will be on resources published between 2015 and 2024 to ensure relevance. Historical research to be focused on include:

- Njanka, S. Q., Sandula, G., and Colomo-Palacios, R. (2021). It-business alignment: A systematic literature review. Procedia Computer Science, 181:333–340.
- Kenge, R. and Khan, Z. (2020). A research study on the erp system implementation and current trends in erp. Shanlax International Journal of Management, 8(2):34–39.
- Guo Chao Alex, P. and Chirag, G. o. C. I. S. (2014). Cloud erp: a new dilemma to modern organisations? Journal of Computer Information Systems, 54(4):22–30.
- Bagchi-Sen, S., Baines, N., and Smith, H. L. (2022). Characteristics and outputs of univer- sity spin-offs in the united kingdom. International Regional Science Review, 45(6):606– 635.

1.4.2 Methods of Investigation

As mentioned earlier, this study aims to thoroughly examine the literature on software systems and student development of software in developed nations to identify applicable lessons for developing environments like South Africa. A key objective is to equip students with the knowledge needed in PM and system development to create software that adheres to industry best practices. To achieve this, the study will assess the business case for higher educational institutions to develop commercially viable software for the industry as potential clients. Once the need is established, the next step will be identifying the technological and functional requirements for creating software that real-world clients can use to support their daily operations. Once the software is completed, the study will evaluate the usability of the software as well as the learning experience and the proficiency of postgraduate students in applying project management and systems development methodologies to complex software development. The aim is to offer significant and insightful knowledge that can be utilised by higher education institutions (HEIs) in underdeveloped countries to improve the functionality, efficiency, and acceptance of their software and systems development projects. This will help HEIs teach the most relevant and useful theories of project management and teamwork, ultimately enhancing their students employability and marketability in the industry.

As such, the researcher will serve as an integral part of the project, overseeing software development. As mentioned, according to set project management milestones, the software system will undergo testing by reviewers with experience in the information technology and ERP industries. All lessons learned and the input from reviewers will be compiled to produce a workable framework that can direct HEIs in leveraging project management and software development curricula to develop commercially viable software solutions and systems for internal and industry applications.

1.5 Chapter Division

This thesis is structured into six main chapters, each meticulously crafted to delve into specific facets of the research endeavour, elucidating methodologies, literature insights, empirical findings, and concluding reflections. The chapters are as follows:

The Research Design chapter serves as the foundational cornerstone of the thesis, this chapter provides a detailed exposition of the research methodology and paradigm guiding the study. Adopting a design science research methodology and a positivist paradigm, this section navigates the theoretical underpinnings of the research approach, outlining its epistemological and ontological foundations. Through meticulous delineation of research methods, including data collection techniques, sampling strategies, and analytical frameworks, this chapter lays the groundwork for subsequent empirical inquiry. Once this chapter has concluded, a review of the current literature in the field will be done to pinpoint previous scientific findings that will benefit the current study.

The Literature Study chapter embarks on an immersive journey through the annals of academic literature, traversing diverse domains ranging from ERP systems to implementation frameworks. The section unfolds with a lucid articulation of the problem statement, contextualising the research within the broader academic discourse. Subsequent sections delve into the intricacies of ERP systems, exploring their architectural nuances, functional capabilities, and strategic significance within organisational contexts. Furthermore, the chapter probes into the value chain framework, dissecting the interconnected dynamics of people, processes, and technology in driving organisational success. Anchored by comprehensive reviews of commercialization strategies, software implementation frameworks, and the role of students in industry, this chapter serves as a beacon of scholarly inquiry, illuminating the theoretical landscape underpinning the research endeavor. Once the theoretical landscape has been illuminated, a case study can be conducted to generate findings regarding the research question, and help the research endeavour edge closer to final conclusion.

The Case Study embarks on a voyage of empirical exploration, this chapter unfurls the intricacies of the case study conducted within the hallowed confines of TaskFlow. With a keen focus on project design and execution, this section unveils the methodological blueprint guiding the project life cycle. Through an exhaustive examination of project management methodologies, agile development frameworks, and strategic planning processes, this chapter offers unparalleled insights into the operational dynamics underpinning software development endeavours. Furthermore, by delineating the project pipeline and process, this section charts the trajectory of the project's evolution, elucidating key milestones, challenges encountered, and lessons learned along the way. Once the case study has concluded, and sufficient data is collected around the subject at hand, an expert reviewer will reveal the depth of the truth within the findings and help the project reach its final conclusion.

In pursuit of academic rigour and scholarly validation, the Expert Reviewer chapter invites the discerning gaze of domain experts to scrutinise the research methodology and findings. Through structured interviews, expert assessments, and critical appraisals, this section seeks to engender a dialectic discourse, fostering intellectual exchange and epistemic enrichment. By subjecting the research endeavour to rigorous scrutiny, this chapter endeavours to enhance the robustness and validity of the research findings, ensuring their salience and applicability within the broader academic community. The chapter that follows will discuss the results that were found throughout the study.

The Results and Discussion serves as the crucible of empirical inquiry, this chapter presents and analyses the empirical findings gleaned from the case study and expert review. Through meticulous data analysis techniques, including qualitative coding, thematic analysis, and comparative synthesis, this section unveils the rich tapestry of insights garnered from the research endeavour. Furthermore, by fostering a dialectic discourse between empirical findings and theoretical frameworks, this chapter endeavours to unravel the underlying mechanisms and causal relationships shaping the research phenomena. The chapter that follows will make the final conclusions regarding the total study and its findings; it will act as a summary for the research project and summarise the findings to draw the final conclusion.

The conclusions chapter aims to synthesise key insights, implications, and avenues for future research, this chapter offers a reflective denouement to the research endeavour. Through a judicious synthesis of empirical findings and theoretical frameworks, this section distils the essence of the research inquiry, elucidating its broader implications and relevance within academic and practical contexts. Furthermore, by delineating avenues for future research and scholarly inquiry, this chapter seeks to catalyse intellectual curiosity and foster a culture of continuous learning and innovation.

In summation, this research endeavour represents a concerted effort to bridge the gap between academic theory and industrial practice, laying the foundation for the subsequent chapter, "Research Design." By elucidating the theoretical underpinnings and methodological frameworks guiding the study, the forthcoming chapter endeavors to translate lofty aspirations into tangible research methodologies, fostering innovation, collaboration, and economic prosperity within the South African software development landscape.

Chapter 2

Research Design

This chapter outlines the research design employed to in the study. The aim is to investigate and reveal the most optimal research methodology and paradigm that encourage the success of the case study. The methodology encompasses Design Science Research (DSR) within the positivist paradigm, emphasising the creation and evaluation of innovative artefacts to address specific challenges. Data collection will primarily involve monitoring a case study, supplemented by interviews with experts, with analysis conducted through quantitative methods.

The following chapter will explore and pinpoint areas of relevance to the study within the domain of research design. The chapter begins by examining various research methodologies relevant to the field and discussing the chosen methodology to be used in greater depth, addressing the relevant literature and the effect these findings will have on the research project at hand. The same exploration is then done on research paradigms and which will be most suitable and fitting for the current study, concurrently addressing the affect the choice will have on the research project. The chapter then continues to delve into methodologies for determining the success of completion of the artefact and makes use of that information to determine what data collection will be done so that a quantitative conclusion can be made at the end of the study. The rest of the chapter defines how the research is designed and how it will be conducted, as well as how ethical clearance is acquired and what ethical implications the project could have.

2.1 Exploring Research Methods using the Research Onion

When doing research, the research onion created by Saunders et al. (2009) is frequently used as a design guide. A growing number of researchers from different disciplines are increasingly embracing the research onion model as a basis for their study design, despite its original primary use in business research. The layers of the research onion represent several aspects of formulating a research strategy, the outer layers are more theoretical and the interior levels more practical (Saunders et al., 2009). The figure below shows the research onion that was outlined by Saunders et al. (2009).

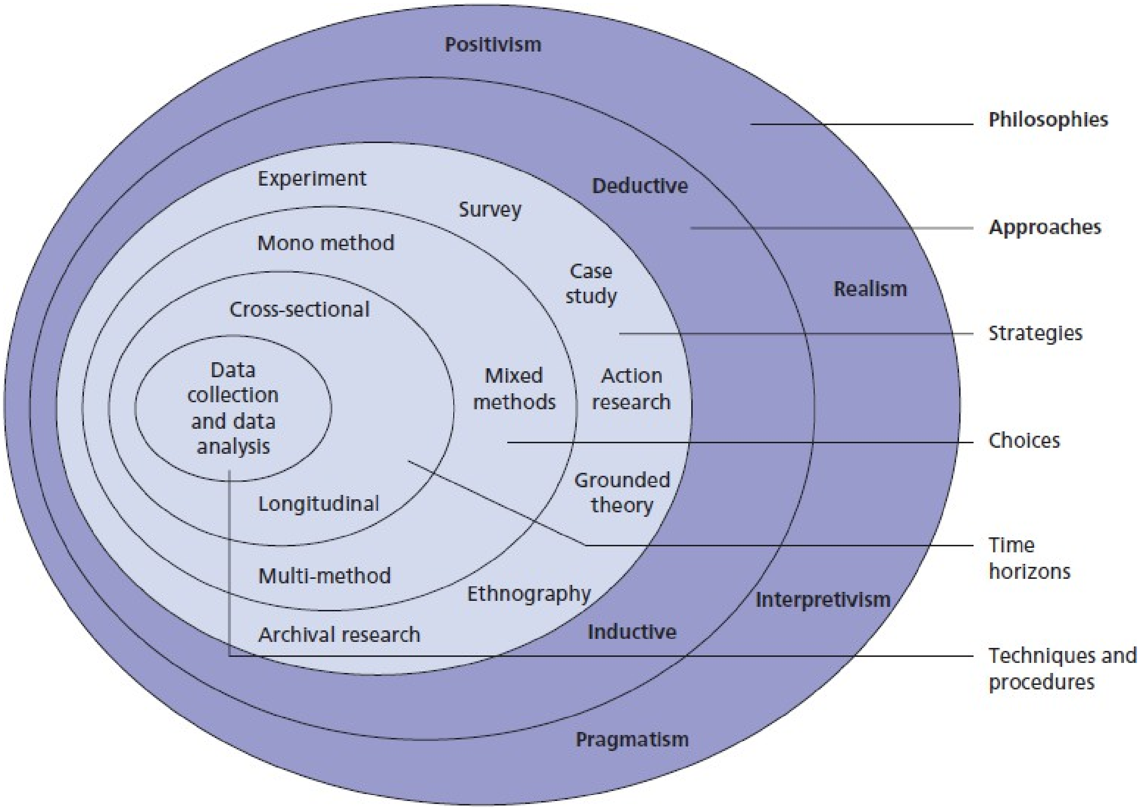

Figure 2.1: The Saunders Research Onion

Since IS is a field of study that integrates both the natural and social sciences, employing the research onion as proposed by Saunders et al. (2009) in and of itself requires some adjustments as it has not yet taken into account all the techniques and strategies used in the field of IS. Therefore, Mardiana (2020) proposed a modified version of the research onion that is more catered towards the field of information science research. The modified version of the research onion can be seen in the figure below.

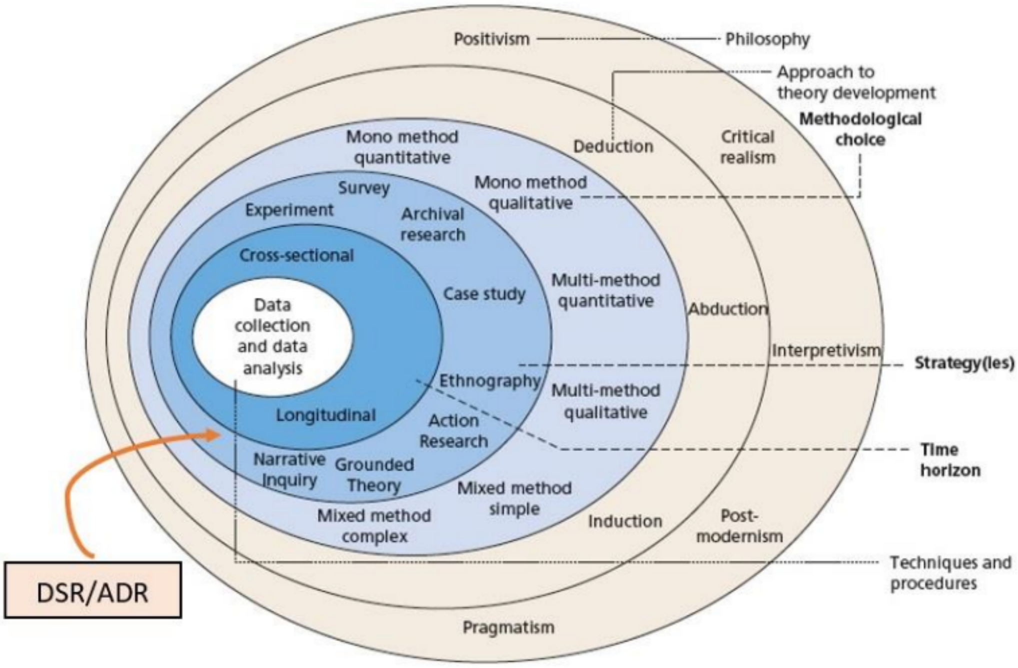

Figure 2.2: The Modified Research Onion

As it is some what difficult to understand the implications of the modified research onion, Mardiana (2020) also provides the following table showing how research design can be constructed by making use of the modified research onion.

Philosophy	**Approach**	**Methods**	**Strategy**	**Time horizon**	**Data collected**
Positivism	Deductive	Quantitative	Experiment	Cross-sectional Longitudinal	Numerical
Positivism	Deductive	Quantitative	Survey	Cross-sectional Longitudinal	Numerical
Interpretivism	Inductive	Qualitative	Archival research	Cross-sectional	Non-Numerical
Interpretivism	Inductive	Qualitative	Case study	Cross-sectional	Non-Numerical
Interpretivism	Inductive	Qualitative	Ethnography	Cross-sectional	Non-Numerical
Interpretivism	Inductive	Qualitative	Action Research	Cross-sectional	Non-Numerical
Positivism Interpretivism Pragmatism	Abductive Deductive	Quantitative Qualitative	Design research	Cross-sectional Longitudinal	Numerical Non-Numerical
Interpretivism	Inductive	Qualitative	Grounded theory	Cross-sectional	Non-Numerical
Interpretivism	Inductive	Qualitative	Narrative inquiry	Cross-sectional	Non-Numerical

Table 2.1: The possible research design composition in IS research

The research onion and its recommendations will be used to construct the research design that will be employed by this research project, as it provide guidelines for which options of the various components of research align with one another and are well suited. Each component of research, namely philosophy, approach, methods, strategy, time horizon, and date collected, will be further researched and investigated throughout the rest of this chapter in order to define the complete design of the research project.

2.2 Research Philosophy

A research philosophy explains the study's methods and the type of knowledge that was produced, as well as the researcher's viewpoint on the relationship between knowledge and development. The subsections that follow outline the various philosophies that form part of the modified research onion created byMardiana (2020). Each philosophy will be studied independently in order to determine the viability of the outlook in the context of the research being conducted.

2.2.1 Positivism

Philosophers Descartes and Locke served as major influences during the 17th and 18th century Enlightenment, which is when positivism first emerged (Park et al., 2020). Over time, logical positivism also began to emerge. The idea that there is a universal, temporal truth that all disciplines must adhere to is greatly influenced by logical positivism, which holds that although scientific laws frequently emerge through the use of intuition, this does not entail that the laws can be subjectively justified (Luczak-Rösch, 2013). Moreover, logical positivism's theory of truth consists of providing a solution to the following query: "For every given statement p, what are the conditions under which p (is true) and what are the conditions under which not-p?". Therefore, "a priori" (such as mathematical axioms) and "empirical" propositions—both of which require rigorous and open validation—are central to the logic of positivism (Luczak-Rösch, 2013). Positivism, which is frequently connected to experiments and quantitative research, is seen as an evolution or kind of empiricism (Potrac et al., 2014) Essentially, positivism comes down to the following: reliable findings can only come from occurrences that can be observed, and the researcher has no control over the findings of the investigation. Therefore, the researcher has no control over the study's outcomes.

The philosophical position of positivism is based on natural scientists' use of observed reality in society to generate generalisations. Positive thinking emphasises the value of information in general and places a stronger emphasis on considering facts and pure data free from human bias or interpretation (Saunders et al., 2009). If a researcher were to embrace an extreme positivist standpoint, several consequences would ensue (Alharahsheh and Pius, 2020):

- The researcher would regard organisations or other related social entities as tangible entities, akin to physical objects and natural phenomena.

- Epistemologically, the research focus would prioritise the identification of observable and quantifiable facts or patterns. Moreover, the phenomena subjected to observation and measurement should contribute to the establishment of credibility and significance in the collected data.

- The researcher's objective would be to uncover causal relationships among the gathered data, facilitating the formulation of law-like generalisations akin to those formulated by scientists. Additionally, the researcher would utilise and incorporate fundamental universal principles and laws to justify and elucidate the behaviour or events studied within organizations.

Based on the requirements of the research project and its nature, the positivist philosophy could apply, but may not be precisely accurate and relevant for the desired outcome and intended motivation and strategy.

2.2.2 Critical Realism

A crucial component of the formation of our natural and social world, according to CR, are the structures and mechanisms that give rise to events and discourses. The main reason for the development of critical realism was to address the positivist crisis (Carlsson, 2003). Critical realism is based on a widely liberating axiology, an inclusive realist/interpretivist epistemology, and a transcendent realist ontology. Despite being a relatively new viewpoint, critical realism is being adopted by many academic sectors such as information technology (Easton, 2010). Importantly, information sciences is the main focus of this study and Wikgren (2005) has done notable research in this field. According to CR theory, it is crucial to make the clearest possible distinction between human action and socio-cultural structuring when examining human information in context. Individuals' motivations, intentions, and plans, which determine their actions, may differ greatly from the characteristics held by the social and cultural forms (the organisations, processes, jobs, and daily circumstances) that govern information activities (Wikgren, 2005).

Critical realism emerged as a response to the crisis within positivism, with Roy Bhaskar's seminal work "A Realist Theory of Science" in 1975, which was later reiterated, introducing the concept of "transcendental realism". Bhaskar (2013) further elaborated on this foundation in "Possibility of Naturalism," where he applied his ideas specifically to the social sciences, formulating what he termed "critical naturalism." These foundational texts laid the groundwork for what would later become known as "critical realism," a term Bhaskar himself adopted. Throughout the 1980s, Bhaskar continued to refine his philosophical stance, engaging in rigorous debate and argumentation. Concurrently, other scholars within the critical realism framework, such as Archer (2013) with her work "Social Origins of Educational Systems" and Sayer (1992) with "Method in Social Science", made significant contributions, enriching the discourse. Initially directed towards critiquing positivism, critical realism evolved to encompass critiques of alternative paradigms, including postmodernism and structuration theory. As such, it stands as a comprehensive and coherent alternative to positivism and various strands of postmodern thought, offering a robust philosophical foundation for inquiry across disciplines.

Various philosophies of science hold distinct ontological perspectives. Idealism posits that reality is not independent of the mind, with different forms of idealism reflecting diverse beliefs about the nature and origin of human consciousness. In contrast, realism asserts that reality exists autonomously, irrespective of our perceptions, beliefs, or discourses. Realism, like idealism, encompasses a range of interpretations. In contemporary discourse, realism predominates among philosophies of science. According to Bhaskar (2013), the pivotal concern is not whether one subscribes to realism but rather the specific variant of realism one embraces.

This philosophy could prove to be a valuable perspective in the context of this research study. However, the remaining possibilities for research philosophies must still be investigated. Therefore, this section will proceed to conduct an investigation of the literature regarding interpretivism.

2.2.3 Interpretivism

Interpretivism shares historical roots with positivism in anthropology. But because it opposes positivism, it is sometimes referred to as anti-positivism (Flick et al., 2004). According to interpretivism, knowledge and truth are dependent on people's experiences and interpretations of them, and they are also subjective, culturally and historically placed. Since it is impossible for researchers to be totally detached from their personal values and opinions, these factors will always influence how they gather, evaluate, and analyse evidence (Ryan, 2018).

Through criticism of positivism from a subjective standpoint, interpretivism emerged. Interpretivism focuses more on the intricate details and context-related variables; it views people as distinct from physical phenomena since they are capable of generating deeper meanings and cannot be studied in the same manner as physical phenomena. As a result, research in the social sciences must be distinguished from research in the natural sciences (Alharahsheh and Pius, 2020). Here are some variations of interpretivism based on Littlejohn and Foss (2009).

- Interpretation and comprehension of philosophy are referred to as hermeneutics. Biblical sources and wisdom writings are its primary areas of focus.
- Phenomenology: This approach uses firsthand observation of phenomena to try and explain the world.
- Symbolic interactionism is a theory that views symbols as communal objects that provide meaning. In light of this, it is thought that symbols offer tools to aid in the formation of reality.

As was previously mentioned, positivist research philosophy subverts interpretivism, which is more appreciative of individual views and ideas. However, interpretive research has its detractors since it rejects and questions the validity of knowledge that has been produced as a basis and that is accepted as a universal norm. It also needs criteria that are different from those employed in positivist research (Alharahsheh and Pius, 2020). This option is not necessarily the strongest candidate for a chosen research philosophy; however, it must be investigated. The next methodology to be researched is pragmatism; its roots and ideas are discussed below.

2.2.4 Pragmatism

While interpretivism and qualitative research are often associated, there are alternative approaches as well. Along with critical research and sometimes positivism, pragmatist paradigms can also be used in information systems qualitative research. Proactive learning, intervention, and action are associated with this paradigm (Goldkuhl, 2012). According to pragmatics, the research issue is the primary factor in determining which epistemology, ontology, and axiology you choose to use—one may be more relevant than the other for addressing a certain question. Furthermore, the pragmatist's belief that it is entirely feasible to deal with variances in your epistemology, ontology, and axiology is confirmed if the research issue does not clearly suggest that either an interpretivist or positivist philosophy is adopted (Saunders et al., 2003).

The pragmatic approach is mostly used to resolve philosophical disagreements that could otherwise go on forever. Is there one world or several?–destined or unbound?—substance or mystical?These are ideas that may or may not reflect the best interests of the world, and disagreements on these ideas are endless. In these situations, the pragmatic approach is to attempt to comprehend each idea by following its corresponding practical ramifications (James, 2020).

2.2.5 Conclusion

To summarise the four research philosophies that were investigated, Saunders et al. (2003) provides the following figure that summarises the ontology, epistemology, axiology, and data collection techniques that are most often used. The table is as follows:

	Positivism	**Realism**	**Interpretivism**	**Pragmatism**
Ontology: *the researcher's view of the nature of reality or being*	External, objective and independent of social actors	Is objective. Exists independently of human thoughts and beliefs or knowledge of their existence (realist), but is interpreted through social conditioning (critical realist)	Socially constructed, subjective, may change, multiple	External, multiple, view chosen to best enable answering of research question
Epistemology: *the researcher's view regarding what constitutes acceptable knowledge*	Only observable phenomena can provide credible data, facts. Focus on causality and law like generalisations, reducing phenomena to simplest elements	Observable phenomena provide credible data, facts. Insufficient data means inaccuracies in sensations (direct realism). Alternatively, phenomena create sensations which are open to misinterpretation (critical realism). Focus on explaining within a context or contexts	Subjective meanings and social phenomena. Focus upon the details of situation, a reality behind these details, subjective meanings motivating actions	Either or both observable phenomena and subjective meanings can provide acceptable knowledge dependent upon the research question. Focus on practical applied research, integrating different perspectives to help interpret the data
Axiology: *the researcher's view of the role of values in research*	Research is undertaken in a value-free way, the researcher is independent of the data and maintains an objective stance	Research is value laden; the researcher is biased by world views, cultural experiences and upbringing. These will impact on the research	Research is value bound, the researcher is part of what is being researched, cannot be separated and so will be subjective	Values play a large role in interpreting results, the researcher adopting both objective and subjective points of view
Data collection techniques most often used	Highly structured, large samples, measurement, quantitative, but can use qualitative	Methods chosen must fit the subject matter, quantitative or qualitative	Small samples, in-depth investigations, qualitative	Mixed or multiple method designs, quantitative and qualitative

Figure 2.3: Comparison of four research philosophies in management research

If the above figure is studied, a separation can be made regarding which research philosophy will be appropriate for the current research study. Positivism immediately stands out because of what its epistemology holds to be true, which is that the only reliable source of information that can be drawn upon is observable phenomena. This suits the current study very well as it will be dealing with a case study to implement a system and the conclusions regarding the success of the development project will be drawn from the success of created artefact or information system. The only other philosophy that has an epistemology aligned to the needs of the research project is realism, as it also views observable phenomena as credible sources of fact, although it opposes a direct reliance on observable phenomena as it believes they can be interpreted incorrectly or with a biased view. Interpretivism and pragmatism regard subjective meaning and social phenomena as the critical source of acceptable knowledge, which does not suit the circumstances of the research project. Therefore, interpretivism and pragmatism are ruled out as options for the chosen research philosophy for this project.

The remaining two options for viable research philosophies must be compared further to find a suitable methodology for the research project. Within the positivistic approach, the research is taken from a third party point of view to remain objective and reliant on the conclusions drawn from the outcome of the research project, while realism ascertains that the researcher is biased by world views and experiences and that this factors will have an impact on the research. As one of the critical purposes of this research project is to create a replicable methodology for how students can consistently build industry level software solutions, it is of the utmost importance that the researcher remain unbiased and objective. Therefore, according to the comparison of epistemology and axiology between positivism and realism, positivism is chosen as the preferred research philosophy for this project.

The next section encompasses the following stage within the modified research onion (Mardiana, 2020), the research approach. This section will be the entry point into the second layer of the research onion.

2.3 Research Approach

Research approaches make up the second layer of the Research Onion. A research strategy aids in the structure of a study by offering direction in determining the theory that the investigation seeks to advance (Saunders et al., 2009). There are three approaches that are offered by the the modified research, namely deduction, abduction, and induction. These three approaches are each investigated in more depth in the subsections that follow. Once they have been reviewed, a conclusion will be made on which option will be taken forward as the approach for this research project.

2.3.1 Deduction

In the deductive method, a hypothesis is generated alongside an established theory in order to test the theory (Saunders et al., 2009). When assessing qualitative data, the deductive and inductive approaches offer a thorough method. To make sense of the entire collection of data and comprehend what is taking place, this procedure entails fully immersing oneself in the reading and consumption of the material (Azungah, 2018). Deductive qualitative research is distinct from other qualitative methodologies since it utilises conceptual premises generated from the review of literature as its starting point and employs them to the gathering and examination of data(Pearse, 2019).

Despite decades of exponential increases in computing capacity, the foundational ideas of the theories of computation still hold true. Thus, the importance of deductive research methods as a source of specific program knowledge has been established over time (Eden, 2007). Knuth (1968) provides the following justification for his definition of computer science as a sub field of mathematics:

> "Like mathematics, computer science will be somewhat different from other sciences in that it deals with man-made laws, which can be [deductively] proved, instead of natural laws which are never known with certainty"

If the advice of Knuth (1968) is taken into consideration alongside the fact that the current research field is computer science, the deductive approach could be very promising to achieving successful results in this project. The next subsection will outline abduction and the potential it holds for being the chosen research approach.

2.3.2 Abduction

When abduction was first used in 1597 by Julius Pacius to translate the Aristotelian apagoge, it went largely unrecognised for nearly three centuries. The term was initially used by C. S. Peirce (Smyth, 1999), who stated that it would represent the only fully knowledge-extending method of interpreting that would be categorically different from the two typical forms of logical conclusion, namely deduction and induction. In several artificial intelligence domains, including diagnosis, natural language comprehension, default reasoning, database updates, planning, and high-level vision, this type of abduction has become a common reasoning method.

Determining the most plausible explanation for a given collection of data is a challenge that can be classified as abduction (Josephson et al., 1987). Abduction is applicable to many different types of reasoning exercises (Charniak, 1985). For instance, the final diagnosis in medicine clarifies the patient's symptoms and indicators (Pople, 1973; Reggia et al., 1983). According to natural language comprehension, a sentence's intended meaning reveals why it was said (Hobbs et al., 1993). Acceptance of a hypothesis in the establishment of scientific theory depends on how well it explains the data. In abductive reasoning, assumption A is accepted if it implies some observation 0 and aligns with preexisting beliefs (Lewis and Mack, 1982).

Abduction has become a common reasoning method in various artificial intelligence domains, including diagnosis, natural language comprehension, and planning. In the context of this master's thesis in computer science, abduction offers a systematic approach to evaluating student developers' capabilities and comparing their processes with industry standards, thereby aiding in the creation of an industry-standard ERP system for TaskFlow. Through its emphasis on determining the most plausible explanation for a given collection of data, abduction provides a valuable framework for ensuring the feasibility and success of the research project. As the project progresses, abduction will serve as a guiding principle for identifying and addressing challenges, iteratively improving the system, and ultimately contributing to the advancement of systems development methodologies within both academic and industrial settings.

2.3.3 Induction

Induction is widely used in the field of computer science (CS). Data structures, philosophy of computing (Lynch, 1996; Hopcroft et al., 2001; Barwise and Etchemendy, 1998), programming languages (Pierce, 2002), program efficiency-time complexity (Cormen et al., 2022), and algorithm correctness (Cormen et al., 2022; Lynch, 1996; Page, 2003; Weiss, 1998) are only a few of the computer science fields that depend on it. Furthermore, induction can be employed as a teaching strategy to improve students' understanding and performance with CS concepts such as algorithm design, recursion, and programming languages (Manber, 1989; Wu et al., 1998). According to Bruce et al. (2003),

> "Programmers with a good understanding of mathematical induction find it much easier to write and, even more importantly, provide convincing arguments for the correctness of recursive algorithms."

However, research has shown that, even after receiving repeated training in several courses within their curriculum, students typically struggle to comprehend and execute proofs via induction (Lowenthal and Eisenberg, 1992; Movshovitz-Hadar, 1993; Baker, 1995; Dubinsky, 1989). Even after receiving repeated teaching in several courses within their curriculum, it has been reported in the literature that students generally struggle to comprehend and execute proofs by induction (Polycarpou et al., 2008).

In conclusion, induction serves as a foundational pillar within the field of computer science, playing a vital role across various domains Its significance extends beyond theoretical frameworks, as it also serves as a pedagogical tool to enhance students' understanding and proficiency in fundamental CS concepts. Despite its ubiquity and importance, empirical evidence suggests persistent challenges among students in grasping and applying inductive reasoning. Addressing these challenges requires a nuanced approach that combines theoretical instruction with practical application, leveraging insights from cognitive science and educational psychology to scaffold students' learning experiences effectively. By acknowledging the complexities inherent in mastering induction, educators and researchers can develop tailored interventions aimed at fostering deeper comprehension and proficiency in this essential aspect of computer science education that could lead to a successful outcome within this research project.

The research approach that is made use of within this research paper is deduction as it holds that with this approach a researcher begins with a hypothesis and then aims to acquire evidence to prove it, support it, or outrightly refute it. This approach is used by researchers when they want to test an existing hypothesis rigorously. Therefore, it is the most suited to what is being tested in this research project and will further be used as the research approach in this project aiming to make use of the assumption that students can build an ERP system of industry standards and put it to the test. The following section regards the next layer of the modified research onion that will be focused on and is called the research strategy, where the methods of data acquisition are investigated and the chosen method is then selected.

2.4 Research Strategy

This layer of the modified research onion refers to the research strategy that will be employed throughout this research project. The research strategies available in the modified research onion are narrative inquiry, grounded theory, action research, design science research, action design research, ethnography, case study, archival research, experiments, and surveys (Mardiana, 2020). As the deductive research approach was selected for this research project, only the strategies that make use of a deductive reasoning approach will be investigated. These research strategies include experiments, surveys, action research, design science, and action design science. They will each be examined in more depth in the subsections that follow.

2.4.1 Design Science

The Design Science Research Methodology, or DSRM, was developed by Peffers et al. (2007) with three goals in keeping in mind: "(1) offer a nominal procedure for carrying out DS study, (2) expand on earlier research on DS in IS and related fields, and (3) offer scholars a conceptual model for a framework for the results of research." The Design Science Research (DSR) paradigm is based on the artificial sciences and engineering. In essence, it is a paradigm for solving problems. DSR creates new artefacts in an attempt to advance human understanding (Hevner et al., 2010). In the past few years, a number of researchers have successfully brought design research into the realm of information science (IS) studies, such as Hevner et al. (2004) and Walls et al. (1992). They are also effective in making design a major part of research and proving the value and legitimacy of design science (DS) as an IS research paradigm. In the fifteen years or more that have elapsed since those first articles, very little DS research has been successfully published in the IS field, despite these effective breaches (Peffers et al., 2007) outlines a process that encompasses six activities that make up the DSR methodology and are arranged in a certain sequence that dictates the order of progression for research based on this methodology. This specific research methodology is investigated in great depth as it forms the basis of action design research that is discussed in a later subsection. The table below outlines these activities:

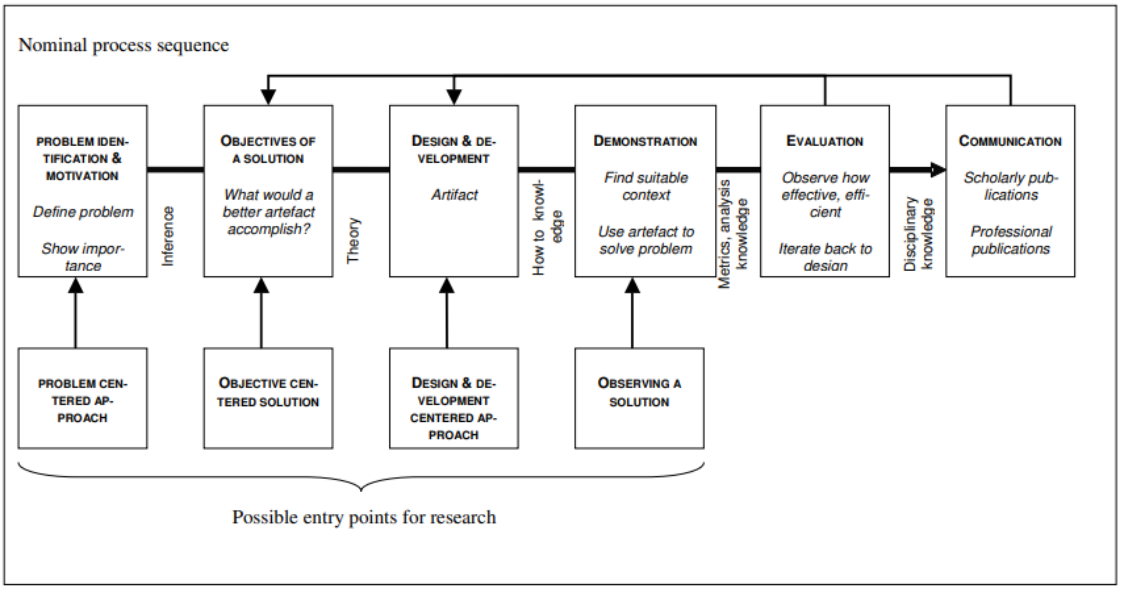

Figure 2.4: Design science research process (DSRP) model

Each step of the DSR research process that is outlined by Peffers et al. (2007) is investigated in broader context to better understand accurate execution and use of the research methodology.

1. Problem Identification and Motivation

   Before coming up with viable solutions, designers work with vague problems that need more investigation (Buchanan 1992; Rittel and Webber 1973). Prescriptive engineering design procedures frequently place a strong emphasis on characterising an issue as the first step. Problem is the driving force behind these problem-first approaches, which use design to find a solution (Dewey, 2022). There for it is of the utmost importance to correctly identify and define the problems and motivation within this research study.

   The purpose of this first activity within the research methodology is of utmost importance as it designates the entry point to the entirety of what will be done within this research. Peffers et al. (2007) specifies what has to be done to properly identify the problem and motivation related to the study. The research problem has to be specified and an explanation to why a solution is valuable must be provided. It could be helpful to conceptually atomize the problem so that the solution can adequately represent the complexity of the problem, as the problem definition will be utilised to create an effective artifactual solution. Two goals are achieved when a solution is justified: first, it encourages the researcher and the research's audience to pursue the answer and accept the findings; second, it clarifies the logic behind the researcher's comprehension of the issue. Knowledge of the problem's current condition and the significance of its solution are among the resources needed for this task.

2. Objectives of the solution

Determine a solution's goals based on the definition of the problem. The goals can be qualitative, such as when a new artefact is anticipated to enable answers to issues not previously addressed, or quantitative, such as terms in which a desirable solution would be preferable to current ones. The goals ought to be as follows: logically inferred from the problem description. Information about the current state of issues, existing solutions, and their efficacy, if any, are among the resources needed for this.

3. Design and Development

   This initial phase involves conceptualizing and creating the artifact or solution to address the identified problem or opportunity. According to Hevner et al. (2004), design science research emphasises the creation of novel artifacts, which could be systems, models, methods, or processes, to solve real-world problems. During this phase, researchers draw upon existing knowledge, theories, and best practices designing and develop the artifact, ensuring that it aligns with the objectives and requirements outlined in the problem statement.

4. Demonstration

   Once the artifact is developed, it needs to be demonstrated to showcase its functionality and effectiveness. In the demonstration phase, researchers present the artifact in action, highlighting its features and capabilities. Peffers et al. (2007) emphasise the importance of demonstrating how the artifact addresses the identified problem or improves upon existing solutions. This phase allows stakeholders to visualize the artifact's potential impact and provides an opportunity for feedback and refinement before proceeding to the evaluation stage.

5. Evaluation

   Evaluation is a critical step in the design science research process, aimed at assessing the artifact's effectiveness, efficiency, and utility. According to Hevner et al. (2004), evaluation involves rigorously testing the artifact against predefined criteria to determine its success in addressing the problem or achieving the desired outcomes. Evaluation methods can vary depending on the nature of the artifact and the research context, but commonly include usability testing, performance measurement, and user feedback. The findings from the evaluation phase inform any necessary refinements or adjustments to the artifact before its implementation.

6. Communication

   The final phase of the design science research process involves disseminating the findings and outcomes to relevant stakeholders. Communication plays a crucial role in sharing the knowledge generated through the research and facilitating its uptake and application in practice. March and Smith (1995) emphasise the importance of clear and effective communication strategies to convey the significance of the artifact, its design rationale, and its potential implications for practice or theory. Researchers may use various channels such as academic publications, conferences, presentations, and technical reports to communicate their findings to both academic and practitioner audiences.

2.4.2 Action Research

Action research integrates theory with practice, involving both researchers and practitioners, to address immediate issues within an agreed ethical framework. It's an iterative process where researchers and practitioners collaborate on activities like problem identification, taking action, and reflective learning within specific cycles (Avison et al., 1999).

Action research goes by various names, like participatory research or collaborative inquiry, but they all revolve around the same idea. Essentially, it's about "learning by doing" - a group identifies a problem, takes action to solve it, assesses the results, and adjusts if necessary. Beyond this core concept, action research involves collaborating to address immediate issues while also contributing to broader social science goals. This dual commitment requires active collaboration between researchers and those affected by the research, emphasizing co-learning as a central aspect of the process (O'Brien, 1998).

Susman (1983) provides a slightly more detailed inventory. He separates each research cycle into five separate phases. First, an issue is located, and information is gathered for a more thorough diagnosis. After then, a number of potential solutions are collectively proposed, from which a single course of action is selected and put into effect. Information on the intervention's outcomes is gathered, examined, and evaluated in relation to the degree of effectiveness of the initiative. At this stage, the issue is reevaluated, and a new cycle of the procedure is started. Until the issue is fixed, this process is continued.

The collection of guiding principles that action research follows is what gives it its own identity. Winter (1987) offers a thorough overview of six important ideas.

1. Reflexive Critique: Acknowledges subjectivity in social narratives, urging individuals to reflect on biases and assumptions underlying judgments.
2. Dialectical Critique: Emphasises understanding social phenomena through examining relationships among elements, especially those marked by instability or opposition.
3. Collaborative Resource: Treats all participants as equal contributors, aiming to mitigate biases and foster insight from diverse perspectives.
4. Risk: Recognizes the psychological fears inherent in challenging established norms, emphasizing collective participation and learning.
5. Plural Structure: Advocates for multiple viewpoints and interpretations, promoting ongoing discussion and exploration of diverse options.

6. Theory, Practice, Transformation: Highlights the iterative process where theory informs practice, and vice versa, leading to continuous transformation through critical reflection and action.

Those who use this approach are frequently practitioners looking to deepen understanding of their work, social change activists attempting to launch a campaign, or, more frequently, academics invited into an organization (or other domain) by decision-makers who recognize a problem that needs action research but lack the methodological expertise to address it (O'Brien, 1998). Although a few of the strong points of this research strategy align with the needs of the current research project, there are other strategies that would prove to be more useful in the current context.

2.4.3 Action Design Research

Design research (DR) places artifacts of information technology at the center of the field of information systems. Nevertheless, prevalent DR thinking regards the IT artifact from a technology standpoint and pays little heed to how the organizational context has shaped it. Therefore, the current approaches of design research concentrate on creating the item and leave evaluation for a later, distinct stage. They disregard the fact that the artifact evolves from contact with the organizational context even when its initial design is informed by the researchers' goal, placing a higher priority on technological rigor than organizational relevance (Sein et al., 2011).

Sein et al. (2011) proposes action design research (ADR) as a new DR method to navigate this issue. In order to address pressing organizational issues, AR integrates theory development with researcher intervention (Babüroglu and Ravn, 1992; Baskerville and Wood-Harper, 1998). AR therefore seeks to connect theory with application and action to thought (Susman, 1983). According to Davison et al. (2004), Susman and Evered (2023), it is usually an iterative process built on working hypotheses that are improved across several cycles of investigation. Many researches have stressed that the ADR and DS is similar, and many argue that they are not, others propose a middle ground where methodological support is given by either integrating the two (Sein et al., 2011) or including AR concepts in DR (Pries-Heje et al., 2007).

The goal of the ADR research approach is to create and evaluate a variety of IT artifacts in an organizational setting in order to produce practical design knowledge (Sein et al., 2011). This method addresses two main issues: creating and assessing IT artifacts that target a larger class of problems that are comparable to those seen, and first, intervening and evaluating in specific problem situations that are faced within an organisation. Because of this dual focus, a technique that prioritises the development, application, and assessment of artifacts that represent theoretical underpinnings and research intents is required (Sein et al., 2011).

Sein et al. (2011) presents a figure that outlines the four stages of the ADR method. Each will be described in more depth below the figure. The figure showing the stages is as follows:

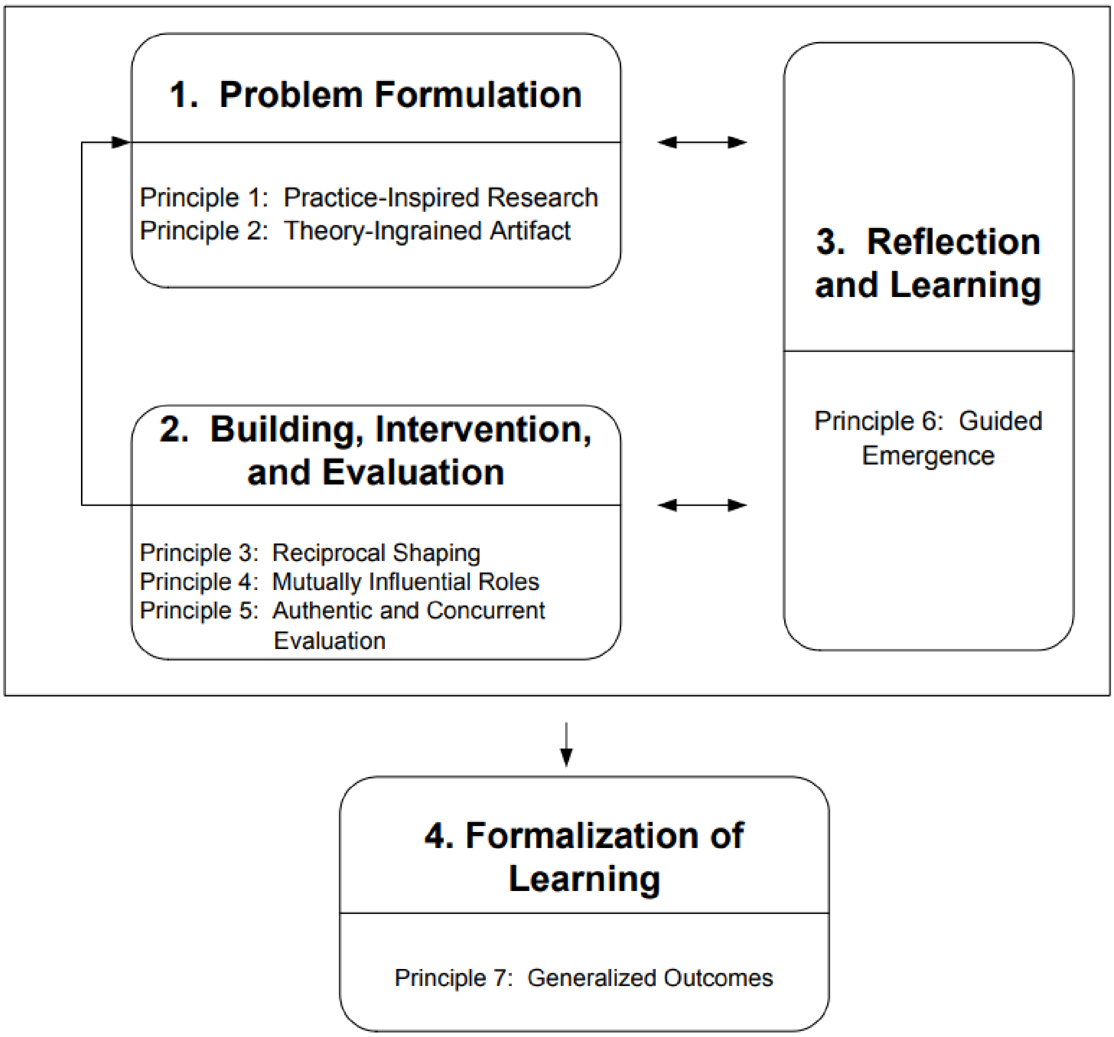

Figure 2.5: The ADR Method: Stages and Principles

Sein et al. (2011) outlines the various tasks that fall within each stage of the process as well. Therefore, the stages and their tasks are discussed and examined below:

1. Problem formulation

   The problem formulation step uses current ideas and tools to identify and conceive a research opportunity (Hevner et al., 2004). A crucial component is characterizing the issue as an example of a certain class of issues. While this definition might be a work in progress, it establishes a foundation for tackling the conflict between addressing the problem at hand and addressing a wider range of issues (Sein et al., 2011). This stage draws on two principles: practice-inspired research and theory-ingrained artifact. Practice-Inspired Research in ADR emphasises leveraging real-world organizational problems as opportunities for knowledge creation, aiming to generate insights applicable to broader problem classes. Theory-Ingrained Artifact underscores the integration of theoretical frameworks into the development and evaluation of artifacts, ensuring they are grounded in established principles for broader applicability beyond the immediate context. The tasks to be completed in this stage of the research strategy are as follows (Sein et al., 2011):

- Identify and conceptualize the research opportunity
- Formulate initial research question
- Cast the problem as an instance of a class of problems
- Identify contributing theoretical bases and prior technology advances
- Secure long-term organisation commitment
- Set up roles and responsibilities

2. Building, Intervention, and Evaluation

   The problem formulation and theoretical foundations chosen in the first stage of ADR are applied in the second stage. These presumptions offer a foundation for creating the IT artifact's initial design, which is then influenced by corporate use and later design cycles. This phase, which is implemented as an iterative process in a target context, integrates the creation of the IT artifact, organizational intervention, and evaluation (Sein et al., 2011). This stage comprises three principles, namely: Principle 3, Reciprocal Shaping, underscores the intertwined influence of IT artifacts and organizational contexts in ADR, involving iterative decision-making cycles within each domain. Principle 4, Mutually Influential Roles, stresses mutual learning between action design researchers and practitioners, where theoretical knowledge and practical insights contribute to a collaborative understanding of organizational challenges. Principle 5, Authentic and Concurrent Evaluation, emphasizes integrating evaluation throughout the research process in ADR, ensuring ongoing refinement and alignment of artifact design and organizational intervention with research objectives. The following are the tasks that are to be completed within the stage of the research strategy:

   - Discover initial knowledge-creation target
   - Select or customize BIE form
   - Execute BIE cycle(s)
   - Asses need for additional cycles, repeat

3. Reflection and learning

   The process of reflection and learning conceptually shifts from creating a solution for a specific problem to using that knowledge to address a larger class of issues. This stage is ongoing and runs concurrently to the initial two phases. The stage acknowledges that conducting research entails more than just problem-solving (Sein et al., 2011). To make sure that contributions to knowledge are recognized, careful consideration of the problem framing, the selected theories, and the developing ensemble is essential. It is imperative to modify the research methodology in light of preliminary evaluation findings to align with the evolving comprehension of the ensemble artifact (Sein et al., 2011). This stage also comprises a principle. Principle 6, Guided Emergence, blends intentional design with organic evolution in ADR. It stresses that the final outcome is shaped by both initial design and ongoing refinement influenced by organizational use, participant perspectives, and evaluation outcomes (Sein et al., 2011). The tasks to be completed in this stage are as follows:

- Reflect on the design and redesign during the project
- Evaluate adherence to principles
- Analyze intervention results according to stated goals

4. Formalization of Learning

   Formalizing the learning is the aim of the fourth phase of ADR. In accordance with Aken (2004), the situational learning gained from an ADR project must to be expanded upon to create broad concepts for a class of field problem solutions (see Principle 1). This conceptual shift is made easier by grouping the problem-instance into a class of problems (see Stage 1). To formalize the learning, researchers define the organizational results and highlight the accomplishments realized in the IT artifact. These results can be described as design principles and, after more consideration, as improvements to the theories that were part of the original design (Sein et al., 2011). This stage also comprises one principle. Principle 7, Generalized Outcomes, deals with making ADR results applicable beyond specific cases. It involves moving from unique instances to broader solutions, including generalizing problems and solutions, and deriving design principles (Sein et al., 2011). The tasks to be completed in this stage are as follows:

   - Abstract the learning into concepts for class of field problems
   - Share outcomes and assessment with practioners
   - Articulate outcomes as design principles
   - Articulate learning in light of theories selected
   - Formalize results for dissemination

2.4.4 Experiment

To ascertain the impact that changes in the independent variable have on the dependent variable in a highly controlled setting, the experiment strategy looks into the causal linkages between the independent and dependent variables (Saunders et al., 2009). A theoretical hypothesis is usually the basis of an experiment, which is intended to support or refute the hypothesis. Therefore, experimentation is employed in research investigations that use a deductive research strategy and concentrates on providing answers to the "how" and "why" research questions.

2.4.5 Survey

The survey approach entails employing questionnaires to gather a substantial amount of data from a broad sample of respondents in an affordable manner (Saunders et al., 2009). Large volumes of qualitative data that can be analyzed using a variety of statistical techniques and be helpful in determining the causes of specific variable connections can be gathered using this procedure. As a result, surveys typically address the research questions "who," "what," "where," "how many," and "how much." They are also typically employed in deductive research studies.

2.4.6 Case Study

Investigating a phenomenon inside a real-life context and gathering evidence from several sources are key components of the case study approach (Saunders et al., 2009). Case studies offer a thorough comprehension of the situation under investigation and the procedure being carried out. The case study approach could employ a variety of data gathering techniques, necessitating the triangulation of sources of information before analysis. Though they can also address the "how" and "what" issues, case studies often address the "why" questions.

2.4.7 Conclusion

Throughout this section, the relevant research strategies are investigated to attempt to unveil the best suited to the current research context. When referring outrightly to methodology and a process of approaching a project, the design science research methodology and the six activities it contains will be used to structure the research project and the research design of the current research project. The strategy is well suited to projects that require iteration and monitoring. As the students that are developing the system will need guidance, regular reviews to reiterate more successfully, and a simple development process that can happen concurrently to the research portion of this project, DSR is the best suited to the current context. Along with DSR, the positivistic paradigm will be employed to conduct research. Positivism emphasises the importance of objective, measurable observations and data. In information sciences, where quantifiable data plays a significant role, a positivistic approach can help maintain objectivity in analysis and determining the success of the created artefact in an academic and economic context.

Although DSR will be made use of to conduct the research, the project will still draw data to create conclusions from a case study being done. The case study is obviously the development project that the students are undergoing. The students will attempt to develop an ERP system to be sold in industry, their results and success will ultimately determine if the research is a success, and if the process that the students follow is reliable in replicating such projects and the project success. This establishment in the chosen philosophy and strategy concludes the useage of the research onion by Mardiana (2020) that guided the exploration of the research methods. The next section reviews the literal design of the research project, how data will be collected and the ethical considerations that should be made note of.

2.5 Research Design

A researcher's detailed approach for addressing his or her research question(s) is known as the research design (Mardiana, 2020).

2.5.1 Data Collection

2.5.2 Ethical Considerations

2.6 Conclusion

Chapter 3

Literature Study

Enterprise Resource Planning (ERP) systems have become a cornerstone in the modern business landscape, facilitating the integration and management of various organizational processes. These systems, characterized by their comprehensive features and functionalities, are designed to streamline operations, enhance efficiency, and provide real-time insights. This literature review explores the multifaceted nature of ERP systems, delving into their features, the competitive dynamics within the ERP industry, and the essential components that define an industry-approved software system.

The subsequent sections address the broader context of the information technology industry, highlighting the challenges it faces globally and within developing nations specifically. The role of universities in bridging the gap between academic knowledge and industry requirements is examined, with a comparative analysis of the unique challenges faced by developing nations versus more developed regions.

A thorough examination of the value chain in software development is provided, focusing on the critical aspects of people, processes, technology, hardware, software, and infrastructure. This section also considers the economic viability of partnerships between academia and industry, emphasizing the mutual benefits and resource-sharing opportunities.

The review further investigates the commercialization of software, discussing various methods and offering recommendations for effective commercialization strategies. The role of students in the industry is explored, considering the advantages, potential threats, and comparisons with similar projects undertaken elsewhere. An in-depth analysis of project management methodologies relevant to software development is presented, along with specific recommendations tailored to the context of ERP systems. Implementation frameworks are examined to identify critical competencies required for successful software development, followed by targeted recommendations. Finally, the review discusses the artefact of ERP systems, presenting methods for measuring the success and functionality of such systems. The Technology Acceptance Model (TAM) is utilized as a framework to evaluate user acceptance and effectiveness.

This comprehensive literature review aims to provide a detailed understanding of the complexities involved in students building ERP systems for industry, offering insights into best practices, challenges, and strategic approaches for successful implementation and commercialization.

3.1 ERP Systems

The ERP archive, which dates back to possibly 1970, was started with the intention of integrating business activities (Shields, 2004). Material Requirement Planning (MRP) systems, which were created in the 1960s and 1970s and allowed manufacturing to be planned according to projected demand rather than past information for the first time, are the ancestors of Enterprise Resource Planning (ERP) systems (Ahlawat, 2017). ERP delivers real-time data from a single centralised database and is multidisciplinary, multipurpose, and multidimensional, in contrast to MRP, which was restricted to procurement, production, and manufacturing. It links and unifies every department across the whole company. The various vendors—known as best of breed implementations—such as those from the designated "big five"—SAP, Oracle, PeopleSoft, JDE, and Baan—which together account for around 70 percent of the ERP market—can be used in tandem with one another (Light et al., 2001).

ERP was first used at the beginning of 1990, and it was named by the Gartner Group (Chang et al., 2000). The early 1990s saw the introduction of ERP by software companies like SAP. In 1992, SAP released the R/3 version once more. Customer-server hardware structure was added to the SAP R/3 so that it could operate on many stages at once (Jacobs et al., 2007). By 2000, all the main ERP software system providers had solved the Y2K challenge. By connecting business and management activities, enterprise resource planning (ERP) tools assist organisations in realising their full potential (Uçaktürk and Villard, 2013). Business patterns will shift over the next ten years as a result of modifications to a vertical market, application techniques, and the ERP cost structure. Cloud application models are stored in a lot of data. SaaS, for instance, is attracting businesses' attention. Business enterprises seeking to reduce significant capital costs through a monthly subscription model have embraced the ERP pricing model, which charges based on usage (Kenge and Khan, 2020).

Zhao and Tu (2021) propose a generalised architecture for the functional design of an ERP system that can be very helpful in understanding the purpose of these systems and what they do. The diagram that describes this functional architecture can be seen below:

The diagram shows how an ERP system interprets the defined business process and how it processes this information to generate an effective output. The services that occur during the business process are integrated with a system that monitors, optimizes, and manages these processes and in turn a completed ERP system is created.

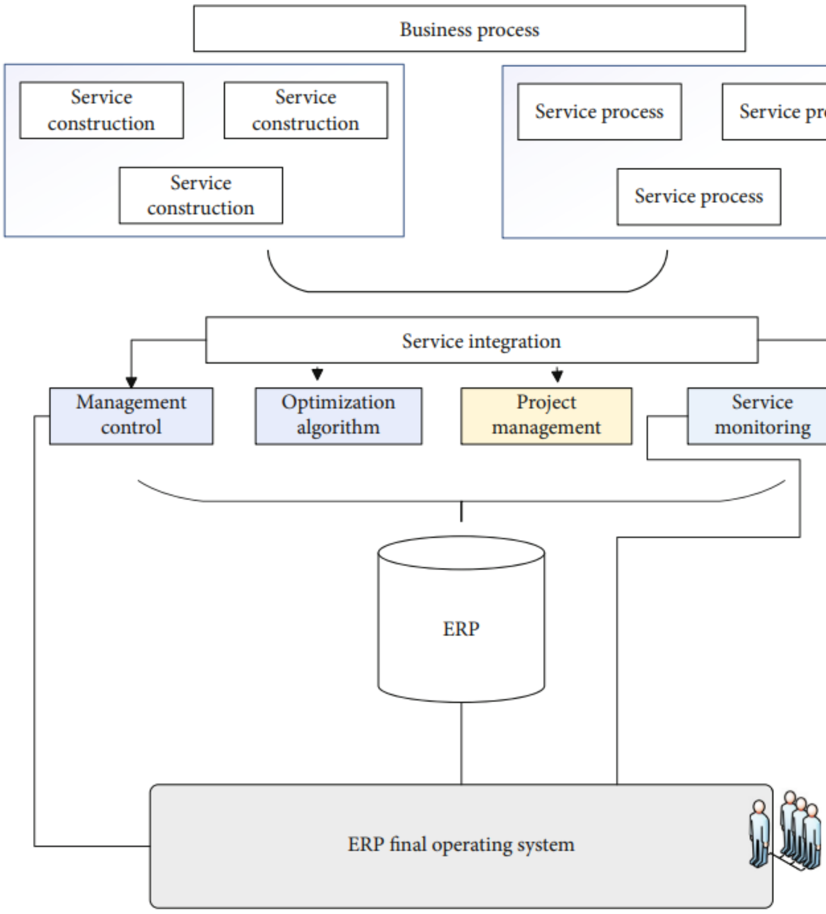

Figure 3.1: ERP System Framework Hierarchy

3.1.1 Features of ERP Systems

There are various features that make up an ERP system, however there is no set standard for what an ERP system must be able to functionally do as ERP systems are configured for the need of the company it is built for. However, generally ERP systems can potentially have any of the following features that are discussed and examined below:

Financial Management

The benefits of implementing enterprise systems for accounting have not been thoroughly studied globally. Furthermore, there are surprisingly few studies that thoroughly investigate the connection between ERP user happiness and accounting gains (Kanellou and Spathis, 2013). Nonetheless, there are studies concentrating on the relationship involving ERP systems and accounting in the pertinent literature. Spathis and Constantinides (2004) investigated the adjustments taking place in terms of accounting software as well as the factors that led businesses to decide to replace their outdated information systems with fully functional ERP systems. The findings demonstrated that the three primary drivers of ERP adoption were the growing requirement for immediate data, the requirement for integration between applications, and the production of information for decision-making. Greater data production flexibility, increased accounting application integration, better report quality (statement of accounts), better decision-making based on timely and accurate accounting information, and a shorter yearly account closure period were the main accounting benefits of ERP implementation.

It was discovered that ERP systems can support novel accounting procedures and serve as data sources for them. More precisely, ERP systems appear to help with data collecting and the organisational scope of management accounting, according to Rom and Rohde (2006) This was further supported by Spathis and Constantinides (2004) who pointed out that the use of these systems encourages the adoption of innovative management accounting practices and improves accountants' efficiency in carrying out daily tasks, managing large databases, and producing reports swiftly and adaptably.

Kanellou and Spathis (2013) concluded that there are several advantages to using an ERP in the accounting department, and the most of these are well regarded. Therefore, the case can be made that an organisation should integrate accounting with its ERP system. Lastly, it is noted that there is a positive correlation between ERP cost and ERP advantages and the degree of ERP user satisfaction. These conclusions are very encouraging and as financial management is a lacking feature within the TaskFlow system, the further development of this integral functionality will be made a priority throughout the case study project.

Human Resources Management

Enterprise managers' main concerns in modern times are increasingly intense business competition amongst enterprises and how to draw in the best talent to become part of the workforce, streamline human resources, cut personnel costs, and increase the competitiveness of enterprises; in other words, these enterprise managers believe that the integration of ERP into the HR system has expanded the system's capabilities to encompass enterprise management. The variety of HR services has expanded as well, from payroll accounting and personnel administration to a comprehensive suite of tools that support business decision-making. Planning for human resources, staff appraisal, scheduling, time management, hiring, payroll, training initiatives, and travel administration are some of these topics. They combine to create an effective and highly integrated ERP, together with the financial and production systems (Zhao and Tu, 2021).

Zhao and Tu (2021) proposes a figure showing the impact of ERP systems in increasing the efficiency of HR procedures within a business. The figure can be seen below and proves the worth and effectiveness of implementing this type of system into the HR workforce.

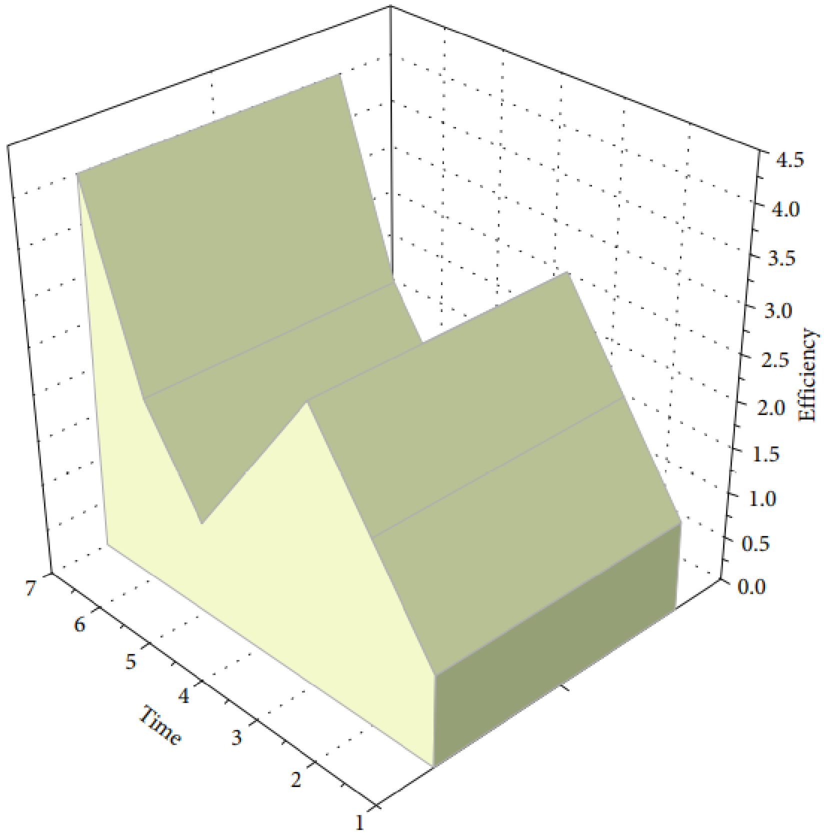

Figure 3.2: Human resource management system optimization rate

Supply Chain Management

The ERP's supply chain component manages all retail operations, including shipments, receipts, issues, and quality control. If you work for a manufacturer, wholesaler, or retailer, managing your inventory is essential to keeping expenses under control and guaranteeing the seamless running of your company. The core functions of every organisation, stock management and valuation, require a significant investment of time and money. Every item's lot-by-lot stock is kept track of, and several computerised information reports are offered to monitor stock movement (Ahlawat, 2017).

The ERP supply chain activities include determining the amount of inventory needed, establishing goals, receiving and delivering goods, maintaining materials in stock subsections, categorising every product, supplying materials to the fabrication department, and fully documenting supplier rejections. Additionally, it offers alternatives and strategies for restocking, keeps track of item usage, reconciles inventory balances, and reports the state of inventory (Ahlawat, 2017).

Customer Relationship Management (CRM)

The phrase "customer relationship management (CRM)" refers to the capacity to continuously engage with customers across a range of channels; it offers the frameworks necessary for businesses to grow and improve their customer offerings and draw in new ones. Companies are now more interested in knowing their customers and interacting with them in order to seize opportunities and overcome obstacles as a result of the growing rivalry in the business world (Kostojohn et al., 2011).

Early in the 1990s, CRM was created with the purpose of managing sales teams and direct marketing, as well as preserving consumer data and reaching their preferences from past purchases and conversations (Bygstad and Presthus, 2013). CRM was created to provide a range of essential tools that could be used by businesses of all sizes and in a variety of industries. These tools would enable them to monitor, manage, and share customer data (Smilansky, 2015), aid salespeople and marketers in examining the behaviour of their clients, and add value to the company through the use of both human and technological resources (Bibiano et al., 2014). Therefore, businesses become more competitive, maximise revenues, decrease effort duplication, improve the effectiveness of information storage, make it available to all employees, and provide a single overview to partners and customers. CRM systems comprise the techniques, tools, and capacities that assist an organisation in managing its customer relationship. A significant portion of this work is closely related to the advancement of information and communication technology, which allows businesses to gather the most data possible about their clients' behaviour, contact details, and other characteristics. It also gives them access to efficient tools and methods for managing this data. Thus, the concept of CRM refers to the ability for businesses to better manage their clientele by implementing dependable systems, processes, and procedures that enable them to obtain the most data about their clientele and interact with them for business objectives in order to obtain specific information that aids in the targeting of goods, services, and new markets (Ronchi, 2009).

Inventory Management

Real-time data on inventory levels and values, encompassing stock on order, raw materials, ongoing work, and completed products, is provided by ERP software. The ERP inventory component handles all aspect of a company's stock-related operations. An inventory management module is a tool that makes collecting information in the inventory department or warehouse quicker and can assist you in keeping the right amount of stock on hand (Ahlawat, 2017).

The supply and demand sides benefit from the application of an inventory management methodology. Inventory management can improve the speed of delivery and stabilise interactions with clients for the supply side. On the demand side, it also lessens the impact of inventory on financial resources, lowers the possibility of material deficits, and quickens the supply chain's reaction time (Zhao and Tu, 2021).

Zhao and Tu (2021) proposes two a flow charts that symbolise the functioning of production, inventory, procurement, and the payment process within an ERP system. This example can be further generalised and seen as a general flow chart of how and ERP system technically and practically functions within a given component of the system which in this case is inventory management. Both these flow charts can be studied below:

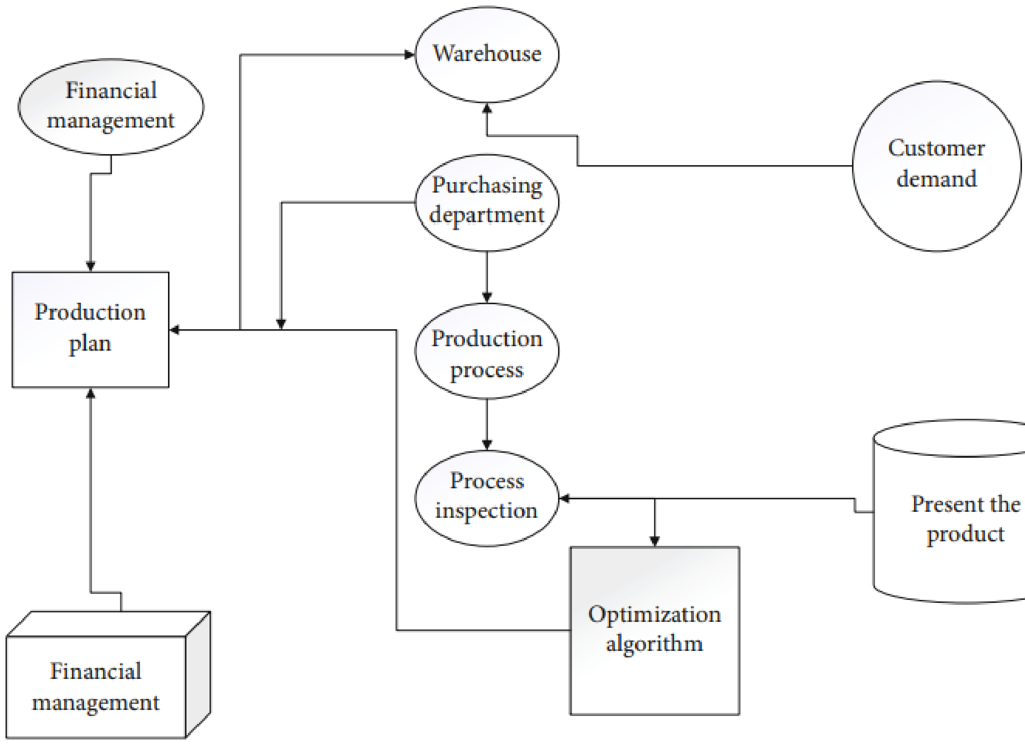

Figure 3.3: Flow chart of the company's production and inventory business based on ERP system

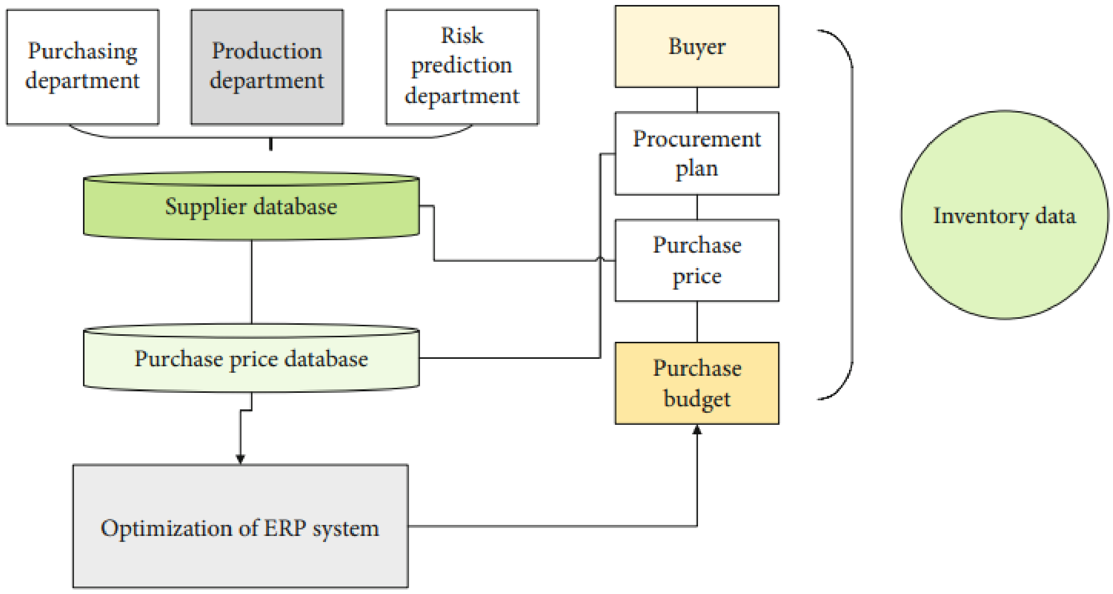

Figure 3.4: Flow chart of the company's ERP-based procurement and payment process

Within this basic examples we can see how the business process would hypothetically flow and how data moves to and from a database storing the information necessary to produce some sort of return. This concept can be applied throughout various modules of an ERP system. Each function is always user input that is processed, stored in a database, and then an output that is once again returned to the user.

Sales and Marketing

A system was created as a tool to address particular business functions beginning in 1975 (Monk and Wagner, 2013). This technology allows for the automation of business function-related tasks and transactions. Nevertheless, the organization's inter-area business operations must manually transfer data because each system is tailored to a particular business function and there is no system integration. Therefore, the likelihood of data duplication is relatively high. For instance, if the finance and supply chain functions also need data on the sales business function, the data is just being reproduced to meet the requirements. This is the primary justification for using a single database for incorporating part or all of an organization's business processes through enterprise resource planning.

It is anticipated that this technology will enable all organisational transactions to be automated and eliminate the requirement for manual data sharing amongst all business operations. Because of this, having an ERP system in place is crucial to an organization's ability to operate profitably (Tsai, 2008).

A business is a company that makes money by selling products or services to customers. Many of the transactions that take place throughout its operation involve businesspeople. The transactions typically comprise a range of organisational data, such as the quantity of inventory, the volume of revenue flowing in and going out, the quantity of goods that need to be produced. Furthermore, transactions typically result in the production of a number of documents, including sales orders, invoices, quotes, and inquiries (Xu et al., 2008).

By making proper use of an ERP system to automate and manage sales and marketing tasks, Terminanto et al. (2017) suggests the following improvements can be made:

- Employ an ERP system to save and combine data from all divisions into a single system. It is anticipated that divisions will be able to coordinate more effectively with the data recorded in the system.
- Generating a quote form through the ERP system. Sales no longer have to access different kinds of papers in order to finish filling out the offer form thanks to the ERP system. Sales can access the system's database and enter the information required to create a quote straight away. Additionally, since the quotation form is automatically prepared and kept in the system, sales do not need to save it in and external system.

Business Intelligence (BI)

Business Intelligence (BI) solutions are becoming the centre of attention for companies when it comes to information systems. The advantages of business intelligence (BI) differ greatly amongst companies. Due to their ability to combine, integrate, and analyse the massive volumes of transactional data produced by ERP systems, business intelligence (BI) solutions are increasingly utilised as extensions of ERP systems (Hawking and Sellitto, 2010).

New apps and the growth of current IT systems were brought about by the need for better information analysis as well as advancements in related technologies. Collaborative systems (CS), corporate performance management (CPM), analytics, knowledge discovery (KD), data mining (DM), and knowledge management (KM) were among them. All of the above listed systems are now frequently referred to as business intelligence (BI) (Gibson et al., 2004; Olszak and Ziemba, 2007).

In today's corporate environment, business intelligence (BI) is deemed highly priority by many firms due to its potential to significantly impact their performance. Companies who employ business intelligence (BI) properly can generate an average return on investment (ROI) of 401 percent over a three-year period, according to Power (1998). 70 percent of the 142 organisations surveyed by Herzum (2003) reported that they were putting data warehousing and business intelligence efforts into practice. Leading business analyst firm Gartner (2009) polled 1,500 Chief Information Officers globally and determined that business intelligence (BI) was the most important technology priority. Accordingly, it is predicted that by 2012, revenue from BI vendors will amount to 7.7 billion dollars (Sommer, 2008).

3.1.2 Competition within the ERP System Industry

The biggest players within the ERP space are SAP, Oracle, and Microsoft. These companies provide the software solutions for business process management on the largest scale. These companies are direct competitors for one another but have less impact on small and medium sized companies who rather go for more affordable and practical solutions. Within the South African space, these small and medium companies make use of Monday.com, Zoho, SalesForce, Clickup, and similar software such as what is being built and sold at Taskflow. This section will proceed to review and compare these various providers and their commercialised systems to determine what commonality there is between them and identify the base requirements of a fully functional ERP system.

SAP

In 1972, Hopp, Wellenreuther, Hector, Plattner, and Tschira developed the SAP method (O'Regan, 2015). The SAP method is made up of a number of fully integrated components that cover almost every aspect of a professional organisation. The SAP approach is the primary source of supply for the ERP system keys industry. In 2012, the SAP software system held around 25 percent of the market and industry.However, Oracle software came in second with 13 percent of the market share, followed by Microsoft Dynamics with 5 percent. SAP is a very large company that does not focus exclusively on its most lucrative ventures. Their produce is truly minor commercially approachable, catering to hundreds of customers who make up less than 1,000 society members.

When one learns the fundamentals of the SAP software system, it becomes a simple tool to utilise even though it may take some time for individuals unfamiliar with it to make the switch from Microsoft Dynamics and Oracle. SAP software systems tend to be more expensive than their competitors, even though prices can vary depending on features and benefits. This makes them less than ideal for small to medium-sized businesses with limited IT funding (Annamalai and Ramayah, 2011).

Oracle

Oracle software E-Business Collection (Barr et al., 2014) is a potent ERP programme with a wide range of features and advantages that have helped elevate it to the status of one of the most advanced ERP programmes available today. For ease of use, Oracle software system E-Business Group may smoothly merge several components into a single system. Additionally, Oracle E-Business Group enables users to automate a number of processes, eliminating the need for human data entry. In addition to increasing efficiency and productivity, this also fixes errors and guarantees that data is not lost.

The Oracle software system E-Business Collection has the ability to record the course of inventory levels at Business Company. Should an item's inventory level drop beneath a predetermined positive standard, the system will immediately provide purchase orders. Very powerful, durable, and intuitive ERP software is the Oracle software system EBusiness Collection, which can meet the needs of almost any industry. Businesses build their own individual elements since the predefined tax element and sales units on the Oracle E-Business Group are frequently insufficient (Barr et al., 2014).

Microsoft

Microsoft Dynamics Marketing is built on Microsoft infrastructure, as its name implies. It synchronises and develops flawlessly with additional Windows commercial requests, facilitating the effortless distribution of data across all methods organisations. Microsoft Dynamics synchronises with various Windows applications, as previously mentioned, simplifying data sharing and transfer. It is built on a Windows substructure and takes a lot less time to implement Microsoft Dynamics Marketing than other approaches. takes the least amount of time to apply out of the three, according to Panorama Consulting Solutions (Solutions, 2016). Large, international industries find Microsoft Dynamics ideal as it supports multiple languages and can work with a variety of currencies in close proximity to global marketing.

Mayer-Barber and Yan (2017) proposes the table below to draw comparisons between these three ERP systems:

	SAP	**Oracle**	**Microsoft Dynamics**
Store Part	19%	13%	16%
Short-list Rate	38%	18%	31%
Collection Rate When Short Recorded	38%	22%	22%
Application Period	23.1 months	24.5 months	23.6 months
Total Price of Ownership	$2.09 million	$2.38 million	$2.06 million
Reimbursement Period	30 months	29 months	12 months
Disruption at Go-live	44%	42%	41%
Realised 50%+ of Anticipated Business Benefits	34%	21%	26%

Table 3.1: Comparison of SAP vs. Oracle vs. Microsoft Dynamics

The table shows that all three of these systems perform similarly across the measurement categories, however it seems SAP remains the most popular and sits in the middle of the three systems when it comes down to cost of ownership and is the most successful in terms of realising the anticipated benefits of system implementation.

Monday.com, Zoho, and Taskflow

These are the three relevant systems when it comes to the ERP system in South Africa. While there are other competitors such as ClickUp, not all available systems will be reviewed in this research paper as there are large amounts of ERP systems available and the lines between what is classified as ERP and what is not have become blurred, therefore it is only necessary to review these three systems. As SAP, Oracle, and Microsoft have overpowered the industry in terms of landing large corporations as clients, these three systems and many others aim to fulfill the needs of small and medium-sized businesses. In this subsection the function of these three systems will be outlined and then compared to review the differences.

Monday.com is a cloud-based work operating system that enables groups to design workflow applications for managing daily tasks, projects, and procedures. For organising, monitoring, and controlling tasks and projects, it offers a versatile and graphical platform. Customisable workflows, task management, team collaboration tools, and interfaces with several third-party apps are some key features. By providing communication, file sharing, and reporting capabilities in one location, it is intended to increase efficiency and simplify project administration (Monday.com, 2024).

Zoho is a comprehensive suite of cloud-based business applications designed to help organisations manage various aspects of their operations. It includes tools for customer relationship management (CRM), finance, human resources, project management, email, and collaboration. Zoho's products are known for their integration capabilities, allowing businesses to streamline processes and improve efficiency. The platform is widely used by small to medium-sized enterprises for its affordability, scalability, and extensive range of features (Zoho, 2024).

Taskflow is a software platform designed to manage and automate workflows and tasks within an organisation. It provides tools for task management, project tracking, and process automation, enabling teams to collaborate efficiently and stay organised. Taskflow typically includes features like customisable workflows, task assignments, deadline tracking, and reporting. It is designed to adapt to various business needs, helping improve productivity and streamline operations by ensuring that tasks are completed in a structured and timely manner (TaskFlow, 2024). The table below compares these various systems.

System	**Monday.com**	**Zoho**	**TaskFlow**
Core Functionality	- Task and project management - Customisable workflows - Team collaboration tools - Integrations with third-party apps	- Customer relationship management (CRM) - Finance management - Human resources - Project management - Email and collaboration tools	- Task management - Project tracking - Process automation - Customisable workflows - Reporting
Pricing	- Varies based on plan: Basic, Standard, Pro, Enterprise - Starting at around $8/user/month	- Varies widely depending on products chosen - Free and paid tiers - Zoho One: $37/user/month for the full suite	- Custom pricing based on business needs - Typically enterprise-level pricing
Attributes	- Visual and user-friendly interface - Highly Customisable - Suitable for teams of all sizes - Strong integration ecosystem	- Comprehensive suite of business applications - Scalable for small to medium-sized businesses - Strong focus on affordability and integration - Extensive range of features	- Designed for workflow automation - Adaptable to various business needs - Focused on improving productivity and task management - Enterprise-level customization and support

Table 3.2: Comparison of Monday.com, Zoho, and Taskflow

The data in the table displays that these systems catering more towards small and medium sized businesses have each found their own business model and sales strategy to increase engagement in their target market as each of these three systems cater for a different audience and have different core functionality.

3.1.3 Components of Developing an Industry Approved Software System

This subsection and the one that follows will both review the process of creating and implementing an industry approved software system as well as an industry approved ERP system. These subsections will review the standard components of an industry systems development lifecycle and will make reference to the standards and practices that need to be upheld in order for these systems to be successful within the climate of the software development industry, not only in the South Africa, but also the rest of the world.

Requirements

Gathering, examining, and recording stakeholder demands and limitations is the process of requirements analysis, which makes sure the software system satisfies their needs (Doe, 2011). Since it establishes the framework for all upcoming development efforts, this stage is crucial. Both functional and non-functional requirements—which specify how the system should operate, such as scalability and reliability—should be included in effective requirements documentation (Sommerville, 2011).

Systems Design

A blueprint for building the software system is created by system designers using requirements. Determining the (Jacobson and Booch, 2021). Good design makes sure that the system's structure satisfies its needs and is efficient, scalable, and maintainable.

Implementation

Developing the program in accordance with the design standards is part of implementation. During this stage, code is written, reviewed, and integrated to make sure it complies with quality standards and design criteria. Maintaining code quality requires adhering to best practices in coding, which include employing version control and coding standards McConnell (2004).

Testing

Testing confirms that the programme satisfies all requirements and functions as planned. It includes a variety of testing techniques, including system testing (evaluating the complete system), integration testing (examining how components interact), and unit testing (evaluating individual elements) (ISTQB, 2011). Finding and fixing issues before release is the main objective (Myers et al., 2011).

Quality Assurance

Planned actions are taken as part of quality assurance (QA) to guarantee that software fulfils quality standards at every stage of development (Team, 2002). This include creating quality measures, carrying out evaluations, and putting procedures in place to raise the standard of quality. QA seeks to stop errors and guarantee dependable and effective software (for Standardization, 2011).

Deployment

Software must be released into the production environment in order for deployment to take place. This stage involves setting up the program, installing it, and making sure it runs properly in a live environment. Thorough preparation is necessary for a successful deployment in order to reduce disruptions and guarantee a seamless transition for users (Sommerville, 2011).

Maintenance

Software systems need to be updated and fixed on a regular basis after they are deployed. To guarantee that the program continues to satisfy the requirements of users and runs safely and effectively, maintenance entails correcting, adaptive, and preventative actions (Pressman, 2005). An essential component of the maintenance stage is managing patches and version updates.

Security

To protect the system from weaknesses, security is incorporated across the whole lifetime. According to Goodrich and Tamassia (2011), these include of restricted access, the use of encryption, secure programming techniques, and frequent inspections of security. To make sure the infrastructure is resistant to threats, penetration tests and vulnerability inspection are helpful.

Documentation

Regarding both current development and upcoming upgrades, comprehensive documentation is crucial. This includes system architecture, code, user manuals, and maintenance procedures (Doe, 2011). By acting as a communication mechanism between stakeholders, documentation guarantees the maintainability and understandability of the system.

Compliance

According to Sommerville (2011), compliance is the system's conformity to legal, regulatory, and industry standards including SOX, GDPR, and HIPAA. Compliance guarantees that the system complies with legal requirements for data security, privacy, and operational openness.

3.1.4 Components of Developing an Industry Approved ERP System

This subsection will once again review the same components as before, except making specific reference to ERP systems, how they differ from other system implementation within the software industry, and providing detail into actions within the process that result in a successful systems development process.

Requirements

Due to the cross-functional nature of ERP systems—which integrate numerous business processes—gathering requirements is one of the most important stages of the process. ERP systems must take into account the needs of different divisions, notably supply chain, customer relationship management (CRM), finance, and human resources, as opposed to normal software projects. ERP failure can result from incorrect or poorly stated requirements, which is concerning considering the high expense and complexity associated with ERP implementations (Monk and Wagner, 2013). The requirements gathering process, which makes sure the ERP system will fulfil organisational goals, consists of several steps, including discovery, stakeholder interviews, and business process modelling (Esteves and Pastor, 2001).

Systems Design

According to Monk and Wagner (2013), ERP systems are made to be modular, enabling businesses to implement just the elements they need to run the system without compromising its integrity. While one module must fluidly interface with the others, system design is made more complex by this modularity. The financial module of an ERP system, for instance, needs to interface with the modules for inventory, procurement, and manufacturing in order to guarantee correct financial reporting. Scalability is another important consideration in system design because enterprise environments typically have high transaction and user volumes (O'Leary, 2000). The ERP software that is selected has a significant impact on the design phase. Programs like SAP, Oracle ERP, or Microsoft Dynamics have predefined workflows and data structures that need to be customised to meet the unique requirements of the company. The incorporation of external systems, such as customer relationship management (CRM) software and business intelligence (BI) platforms, should also be taken into account throughout the design phase as it is common practice create intergrations between different platforms due to client requirements.

Implementation

The process of implementing an ERP involves several stages, including configuration, training, data migration, and customisation. According to Zhang et al. (2003), customisation usually refers to changing ERP modules to satisfy certain business demands, including special processes or regulatory constraints. Contrarily, configuration entails adjusting the ERP system's parameters—like tax rates or payment terms—to match the operational framework of the company. Data migration, or moving data from old systems into the ERP, is one of the trickiest parts of implementing an ERP. Due to data inconsistencies, missing data, or variations in data architecture, this might get complicated. To reduce disruption, ERP systems usually take a staged approach, rolling out modules one at a time (Shanks, 2000).

Testing

According to Ahmad and Cuenca (2013), testing in ERP systems is more complicated than testing in traditional software since ERP connects several business processes from different departments. Integration testing examines the interactions between these modules, whereas functional testing verifies that each module—for example, finance, HR, and procurement—functions as intended. For instance, adjustments to inventory levels ought to be automatically reflected in the finance module. Because it involves end users running real-world scenarios, user acceptability testing (UAT) is essential to verifying that the system satisfies business needs (Nah et al., 2001). Furthermore, especially in international organisations, performance testing is essential to guarantee that the system can manage large numbers of users and transactions. To safeguard critical data, like payroll data and financial records, ERP testing should also involve security testing.

Quality Assurance

ERP systems use quality assurance (QA) to make sure the solution satisfies technical and business objectives and is error-free (Hawari and Heeks, 2010). QA is a multi-layered process that begins with automated testing and code reviews to identify problems early in the development cycle. Validating that business processes match the ERP configuration and that workflows and transactions flow correctly between modules is another aspect of quality assurance (QA) in ERP installations. Due to the significant financial and operational risks connected with defects—which could lead to erroneous financial reporting, disruptions in the supply chain, or non-compliance with regulations—quality assurance (QA) is crucial for ERP systems (Hawking et al., 2004). ERP providers frequently offer

QA-specific solutions, like SAP's Solution Manager, which aids businesses in managing their QA procedures.

Deployment

ERP deployment is usually carried out in stages to reduce business impact. This usually entails introducing one department or module at a time, beginning with less important tasks and working your way up to more crucial ones like supply chain management or finance (Monk and Wagner, 2013). While more risky, "big bang" deployment strategies—where the entire system goes online at once—are occasionally chosen by businesses that need to make urgent system-wide modifications. Because they provide more flexibility, quicker deployment periods, and lower infrastructure costs than conventional on-premise systems, cloud-based ERP systems—like Oracle ERP Cloud or SAP S/4HANA Cloud—are becoming more and more popular (Vukovic et al., 2023). Cloud-based ERP systems, however, could run into issues with data security, compliance, and limited customisation options.

Maintenance

After going online, maintaining the ERP system becomes a constant effort that involves routine upgrades, bug patches, performance enhancements, and modifications to accommodate evolving corporate needs (Ahmad and Cuenca, 2013). Patches and upgrades are routinely released by ERP suppliers; these must be tested and implemented without interfering with regular business operations. Updating the system to account for new business procedures or regulatory changes—like entering new markets—is another aspect of maintenance. Maintaining ERP effectiveness is essential to keeping the system in line with the changing needs of the company. Reduced ROI, elevated security threats, and operational inefficiencies might result from improper ERP system maintenance (Hawking et al., 2004).

Security

ERP systems manage enormous volumes of sensitive data, including personnel records and financial data, thus security is a top priority. Access controls, encryption, and frequent audits are only a few of the layers that make up ERP system security (Tarafdar and Roy, 2003). ERP systems frequently employ role-based access control (RBAC) to make sure that users can only access information and take actions that are appropriate for their position. ERP providers help businesses manage risks and maintain regulatory compliance by offering integrated security capabilities, such as SAP's Governance, Risk, and Compliance (GRC) solutions (Hawari and Heeks, 2010). To find possible threats and reduce risks, regular penetration tests and vulnerability assessments are crucial.

3.2 Implementation of ERP Systems

The seven main steps in the ERP deployment process include business process analysis, software installation, data migration, software performance testing, user training, complete deployment, and post-implementation support (Ly, 2020). These main key steps are examined in greater depth in the sections that follow.

Software Installation

Following the first step's creation of new procedure flows, the team should have a new business process plan in place. The architecture and infrastructure for software, such as the data store, data presentation, and internet accessibility, will be installed and constructed by the software developers who have recieved the business requirements from the business analyst.

Data Migration

In this stage, all data is transferred to a new software platform. Before the data is transferred to a new site, it should all be reviewed and adjusted to ensure a smooth mapping process. Data mapping between the previous and new store locations, data transfer, and the configuration of a new data storage location are all included in this stage.

User Training

Personnel react to change management, and user training relies on the intricate nature of the ERP program. Up to 56 percent of ERP deployment cases after go live result in production halts under training.

Final Deployment

Depending on the scale of the ERP software and the assets available, the organisation might select one of the following three ways:

- Big-Bang Method: A one-day switch from the outdated to the updated software. This method is quick and inexpensive, but any inefficiency in deployment could lead to a serious issue during use.
- Phased Approach: A longer-term, function- or unit-specific phased transition.

- Parallel Operation Approach: In order to reduce risk, users use both new and old systems simultaneously. This method has greater operational expenses for the two platforms and necessitates more time for repetition of work.

Support

Performance review of ERP projects is crucial and should be done for the duration of the project. Key performance indicators that can be compared are as follows:

- Real implementation costs compared to the budgeted amount
- The return on investment for the project
- The assessment of mistakes or human errors
- The efficiency of the supply chain and production
- The satisfaction of customers and their willingness to continue working with the development company

3.2.1 ERP Implementation Time

Once the software system is launched, an ERP system implementation project may take anywhere from three months to several years to complete and fully implement to a point where both client and development company are satisfied (Sankar and Rau, 2006). The size of the organisation, the volume of data, the number of users, and the resources all affect how long the ultimate project implementation takes (Pelphrey, 2015).

3.3 The Information Technology Industry

The global economy, innovation, and industry transformation are all greatly influenced by the information technology (IT) sector. But even with its broad influence, the IT sector still faces a wide range of difficulties that differ depending on the area and industry. Talent shortages, cybersecurity issues, and the quick pace of technological advancement all pose serious problems for IT professionals and enterprises. These problems are made worse in poorer countries by poor infrastructure, restricted access to technology, and tight budgets. This section examines the global obstacles that the IT industry faces, draws attention to those that are unique to developing countries, and contrasts the issues that these countries confront with those of economies that are more developed.

3.3.1 Challenges within the Information Technology Industry

Many obstacles affect project success and productivity in the software development business. The fundamental complexity of software systems is one of the main obstacles; in his groundbreaking book No Silver Bullet, Brooks (1987) famously referred to this as a "essential barrier". Brooks claims that this complexity results from the diverse nature of software development, which includes managing shifting stakeholder expectations, ensuring functionality across many platforms, and responding to changing requirements. This is made worse by the difficulty of precisely projecting resource and time requirements, which McConnell (1996) identifies as one of the main causes of software project delays and scope creep. Furthermore, the swift progression of technology necessitates that developers consistently embrace novel tools and approaches, frequently without sufficient instruction, leading to a learning curve that may hinder project schedules and quality (Boehm, 1988).

Organisational and human variables can pose a serious challenge to the software development process. Project failure is often caused by dysfunctional team dynamics and poor communication within development teams, according to DeMarco and Lister (2013). Teams that are cohesively weak or have weak leadership find it difficult to reach consensus on project objectives and make important decisions, which lowers output and produces less-than-ideal results. In addition, Glass (2006) notes that the industry's emphasis on prompt delivery over careful design frequently undermines the creative aspect of software development and results in a propensity to give priority to temporary fixes over long-term solutions. Therefore, the technical skill of developers alone is not enough to overcome the larger systemic difficulties facing the software industry without addressing these human and organisational elements. In order to address these issues, complete solutions combining organisational, managerial, and technology advancements are becoming more and more necessary as the industry develops.

Additionally, a challenge that businesses in the software development industry face which is more integral to the outcome of this study, is that it takes a considerable amount of time to train employees that have left university and do not yet have technical experience as a developer. Universities are equipping students with a theoretical background on information technology and are not properly preparing the students for the work environment. The challenge for business is that it takes a considerable amount of time for an employee with no experience to become beneficial for the company.

3.3.2 Challenges Exclusive to Developing Nations

The adoption and application of international industry standards are complicated by the particular difficulties faced by software development in developing countries. The principal impediment to the constant delivery of software projects is the absence of suitable infrastructure. Unreliable electricity and Unreliable internet access are major contributors to productivity losses and project delays in many developing nations (Heeks, 2017).

The lack of cutting-edge hardware and software tools exacerbates this deficiency in infrastructure, forcing developers to depend on antiquated technologies that cannot keep up with the quick changes in software needs (HELTON, 2012). Furthermore, a lack of formal education and specialised training in software development restricts the number of qualified individuals available, ill-prepared teams for challenging projects. Because of this, software engineers in underdeveloped countries encounter both logistical and technological obstacles that severely limit their capacity to compete with their peers in more developed areas.

Software development is further limited by socio-economic and political concerns in poor countries, in addition to constraints related to infrastructure and education. For instance, a lot of developing country governments do not have strong policies in place to encourage the expansion of IT, which discourages both public and private investment in the software sector Ojo (2020). The lack of a strong legislative framework for cybersecurity and intellectual property protection deters international businesses from outsourcing projects to these areas (Heeks, 2017). Furthermore, because organisations are frequently reluctant to engage in new technologies and are resistant to change, cultural attitudes around the use of technology might impede the modernisation process (HELTON, 2012). As a result, although the global software development sector faces many obstacles, the problems in developing countries are mostly caused by socio-political, cultural, and infrastructure constraints, necessitating specific expenditures and policies to support development in these areas.

3.3.3 Recommendations for Developing Nations

With a population of one billion, India is the largest democracy in the world and is quickly becoming a major player in the global IT software market. Developing nations all over the world now look to India as an example of how to advance their technological development. (Tschang, 2001). The Indian software business developed at a remarkable 62.3 percent annual growth rate between 1995 and 2000, while its exports grew at a compound annual growth rate of almost 42 percent, according to the National Association of Software and Services Companies (NASSCOM). The Indian IT industry's earnings increased from 1.73 billion US dollars in 1994 to 1995 to 13.5 billion US dollars in 2001 to 2002. With a far smaller domestic market, about two thirds of the Indian software industry's revenue comes from exports, according to NASSCOM. Although there are more than 1250 businesses that export software services, the top 25 of them generated around 60 percent of total income in 2000 to 2001. The USA (62 percent) and Europe (24 percent) are their two main export destinations. Over 185 Fortune 500 businesses have contracted with Indian software organisations to handle their software needs (Agrawal and Thite, 2003). By making reference to the way the Indian software development industry functions, it can be seen that it will become ever more increasingly important to export the products and services created in developing nations to the rest of the world. Now more than ever, software development companies are selling their products to countries such as the USA and Europe, sometimes even doing business in Australia and Asia.

3.4 The Value Chain

The value chain, which was first proposed by Porter's (1985), describes the interconnected activities that organisations engage in to create value. The value chain, as used in the information technology sector, highlights how people, processes, and technology interact to produce and deliver services. Ensuring efficiency, competitiveness, and innovation requires effective management of these components. The main value chain components are reviewd in the subsections below.

3.4.1 People

People are at the heart of the value chain, driving innovation, strategy, and operational excellence. In the IT industry, the skills and competencies of workers significantly influence the efficiency and effectiveness of both software development and implementation processes. The role of skilled workers in the IT sector has been highlighted by Becker (2009) in human capital theory, where education and training are seen as investments that enhance productivity. In developing nations, the shortage of skilled professionals is a significant barrier to growth, particularly in advanced technological fields (Heeks, 2017). The relevant literature also underscores the importance of creating an organizational culture that fosters collaboration, continuous learning, and adaptability (Schein, 2010).

This component of the value chain will have the greatest effect on the outcome of the project that is conducted for this research paper. The largest factor affecting the success of the project is the capabilities of the university students that will be developing the software. The emphasis placed on this component of the value chain has also started to highlight the overall importance of this component within any given situation. The people involved within a process will outrightly determine the outcome of projects.

3.4.2 Processes

The organised collection of steps and actions required to create, implement, and manage IT systems is referred to as a process. Achieving efficient goal accomplishment and resource optimisation requires effective process management. Methodologies like Agile and DevOps in software development highlight the necessity for adaptable, iterative methods to handle quickly changing customer needs (Fowler et al., 2001). Lack of standardisation and access to sophisticated project management frameworks frequently leads to inefficient processes in underdeveloped countries (Boehm, 1988). Streamlining processes and increasing the pace and calibre of project outputs require process optimisation. The eventual problem with this required process automation is that process automation does not produce a profit for a business and thus it is not given a priority status. Process optimisation seems to only occur when there are avaliable resources and not under other circumstances. Client work always takes preference in the context of industry.

3.4.3 Technology

In the information technology sector, technology is the foundation of the value chain. The supply of IT services and goods is made possible by its hardware, software, and infrastructure components. How innovative and responsive a company is to market demands is largely determined by its technological skills. According to Brynjolfsson and McAfee (2014), the way organisations function has been completely transformed by the incorporation of cutting-edge technology such as cloud computing, AI, and machine learning.

Hardware

The physical foundation of the IT sector is made up of hardware, which includes networking hardware, workstations, and servers. Processing speed, storage capacity, and overall system performance are all directly impacted by the efficiency of hardware components (Hennessy and Patterson, 2017). Access to contemporary hardware might be restricted in developing countries, which has an impact on local enterprises' competitiveness and their capacity to deploy sophisticated IT solutions (Heeks, 2017).

Software

Applications and systems that operate on hardware and give organisations the capability they need to accomplish their objectives are referred to as software. The design, testing, and deployment of software have changed as a result of the quick evolution of software development frameworks like Agile and DevOps (Bass et al., 2015). Nonetheless, a major hindrance to development in underdeveloped countries is the lack of access to high-quality software and the dependence on obsolete or pirated versions to acquire skills and training within a certain technical proficiency (Ojo, 2020).

Infrastructure

The underlying mechanisms, such as internet connectivity, data centres, and cloud infrastructure, that enable the operation of hardware and software are referred to as infrastructure. Scalability and sustainability of IT services depend on a robust IT infrastructure (Ross et al., 2006). Poor infrastructure causes frequent service disruptions and inefficiency, which presents significant issues for developing countries (Ojo, 2020). Furthermore, infrastructure spending is essential to closing the digital gap that separates developed and underdeveloped countries.

3.5 Commercialization of Software

3.5.1 Methods of Commercialization

Recommendation

3.6 Students in Industry

An obvious focus of the case study associated with this research project is allowing graduate students to become involved with an industry development process guided by a company, in this case TaskFlow. This section will review the associated advantages and threats of allowing students to participate in this manner, as well as providing a background to how universities attempt to aid employers by providing industry with skilled workers and what the shortcomings of these attempts are.

3.6.1 The Purpose of a University

3.6.2 The Economic Viability of Academic and Industry Partnership

Owing to the favourable results that these kinds of collaborations frequently provide, the financial sustainability of academic-industry partnerships has attracted a great deal of interest. These kinds of collaborations are essential tools for reducing barriers to information transfer, increasing innovation, and advancing technology. Partnerships with the industry give academic institutions access to real-world data and state-of-the-art technologies, which are critical for developing research projects and applicable curricula. The innovative concepts and pool of talent generated by universities also serve companies, helping them to maintain their competitiveness in a world economy that is changing quickly. The "Triple Helix Model" developed by Etzkowitz and Leydesdorff (2000) highlights how governmental, business, and academic institutions interact to foster innovation through teamwork. They contend that organisations' cooperation promotes the generation of new information, which is essential for economic expansion.

The idea that industry-academia collaborations greatly advance economic growth is supported by empirical research, especially when it comes to enabling the commercialisation of scholarly research (Perkmann et al., 2013). Universities can expedite the transfer of research from the lab to the market by working with industry, which frequently results in the founding of spin-offs and start-up businesses. These partnerships are essential for closing the skills gap as well, particularly in high-tech fields like software development where it is difficult to locate workers with the necessary training. Thus, collaborations between academic institutions and business sectors are crucial for workforce development as well as technology improvement (Ankrah and Omar, 2015). These collaborations are especially more important in the case of developing countries, since they offer the frameworks and resources for innovation that many emerging economies do not have on their own.

3.6.3 Industry Offering Resources to Universities

A key factor in bridging the gap between academics and the real-world demands of the market is the support that industry provide to universities in the form of financing, infrastructure, and internships. This cooperation is especially important in the software development and technology fields, where universities frequently find it difficult to keep up with industry requirements for tools and infrastructure. Real world companies guarantee that universities have access to cutting-edge technologies through funding, ensuring that students have the skills required to make valuable contributions to the workforce. Companies within the industry have always been crucial in providing money for research that directly relates to their operational and innovative demands, as Mowery and Sampat (2006) stated.This interaction produces a feedback loop whereby the real-world issues that enterprises encounter influence university research objectives, resulting in research that is both academically rigorous and commercially valuable.

In addition, industry participation in academics goes beyond funding. Industry efforts that support students' professional development include guest lectures, mentorship programmes, and internships. Through these programmes, students can get practical experience and insights into the difficulties faced by businesses, which will help them in their future career endeavours. Furthermore, collaboration between academic and industrial researchers can result in the creation of collaborative research initiatives that tackle important industry issues (Wadhwani et al., 2014). Together with improving students' employability, these partnerships create an ecosystem of ongoing learning and innovation by bridging the gap between academic instruction and real-world application. This is the ultimate goal of the projects being conducted throughout this research study

3.6.4 Critical Competencies for Software Development

The success of software development is largely dependent on the knowledge and abilities of software engineers, thus Turley and Bieman (1994) sought to identify the competencies that are the most essential. During this study a total of 20 interviews were held with people of various professions and levels of educations within the software development industry. Below a table can be seen showing the various competencies and the number of individuals that have that certain competency. They were measured in three ways, derived competencies which were identified by Turley and Bieman (1994), self-described competencies which were identified by the interviewees upon self-examination, and lastly manager described competencies which were identified by the managers overseeing the interviewees at their day jobs.

The table provides a thorough examination of the necessary skills for software developers, contrasting self-reporting with managers' and experimenters' views. The resulting categories showed high ratings for critical competences including "Team Orientated," "Use of Prototypes," and "Knowledge," suggesting that external evaluators value technical and collaborative skills as essential to software development (McConnell, 1996). But there are differences between the manager's evaluation and the self-evaluation. Developers may have overestimated their collaborative and productivity-related behaviours, as evidenced by the fact that their self-reported scores for "Team Orientated" and "Pride in Quality and Productivity" are typically higher than the managerial ratings .Managers also gave lower scores to competences like "Use of New Methods or Tools" and "Schedules and Estimates Well," which may indicate that developers' perceived and real abilities to embrace new technologies and manage their time effectively differ (Brooks, 1987). This discrepancy highlights the necessity of constant feedback and skill alignment between managers and developers in order to guarantee that managers' judgements and individual views are in line with industry standards (Fagerholm et al., 2014).

Given the results from the experiment conducted by Turley and Bieman (1994), the top three most important critical competencies for software development can be identified. They are as follows:

- Team Oriented (Derived Score: 14): The emphasis placed on this competency highlights the importance of collaboration and effective teamwork within software development teams and companies.
- Use of Prototypes (Derived Score: 14): Prototyping is essential for iterative development. Within the software development industry, iterative development is essential to ensure flawless functionality and allows for early testing and refinement.
- Design Style (Derived Score: 16): This competency could be the most important of all, it describes the developers ability to create efficient, maintainable, and elegant solutions

Table 3.3: Essential Competencies

Competency	Derived	Self	Manager
1. Team Oriented	14	12	2
2. Seeks Help	11	4	
3. Helps Others	2	1	1
4. Use of Prototypes	14	3	
5. Writes/Automates Tests with Code	13		
6. Knowledge	13	12	
7. Obtains Necessary Training/Learning	12	7	
8. Leverages/Reuses Code	10		
9. Communication/Uses Structured Techniques for Communication	8	8	
10. Methodical Problem Solving	9		
11. Use of New Methods or Tools	5		
12. Schedules and Estimates Well	4	2	1
13. Uses Code Reading	4		
14. Design Style	16		
15. Focus on User or Customer Needs	11	1	
16. Response to Schedule Pressure	9		
17. Emphasizes Elegant and Simple Solutions	8	2	
18. Pride in Quality and Productivity	12	1	
19. Pro-active/Initiator/Driver	11		
20. Pro-active Role with Management	10		
21. Driven by Desire to Contribute	8	5	
22. Sense of Fun	7		
23. Sense of Mission	6		
24. Lack of Ego	4		
25. Strength of Convictions	3	4	
26. Mixes Personal and Work Goals	3		
27. Willingness to Confront Others	3		
28. Thoroughness		4	
29. Skills/Techniques	11		
30. Thinking		9	
31. Desire to Do/Bias for Action	5	1	
32. Attention to Detail		4	
33. Perseverance	13		
34. Innovation		4	
35. Experience	3		
36. Desire to Improve Things		3	
37. Quality	2		
38. Maintaining a ”big picture” view/ Breadth of View & Influence	1	3	

The Skill Gap Between University and Industry

3.6.5 Advantages

3.6.6 Threats

3.6.7 Similar Projects

3.7 Project Management Methodologies

3.7.1 Software Development Management Methodologies

3.7.2 Recommendations

3.8 Artefact

3.8.1 Methods for Measuring Artefact Success

3.8.2 Methods for Measuring Artefact Functionality

3.8.3 TAM Model

3.9 Conclusion

Chapter 4

Case Study

4.1 University

4.1.1 Academic Deliverables

4.2 Industry

4.3 Implementation of Project Management Methodology

4.4 Student Team and Roles

4.5 Project Design

4.5.1 Requirements Gathering

Internal Questioning

Abstract Requirements Summary

Functional Requirements Summary

Feature Comparison

4.5.2 System Requirements Definition

Year 1 Prototype

Year 2 Client Configuration

4.6 Project Pipeline and R

Chapter 5

Expert Reviewer

Chapter 6

Results and Discussion

6.1 Recommendations

Chapter 7

Conclusion

/

Bibliography

Agrawal, N. M. and Thite, M. (2003). Human resource issues, challenges and strategies in the indian software industry. *International Journal of Human Resources Development and Management*, 3(3):249–264.

Ahlawat, J. (2017). Role of erp in e-commerce supply chain management system. *International Journal of Advanced Education and Research*, 2(4):61–65.

Ahmad, M. M. and Cuenca, R. P. (2013). Critical success factors for erp implementation in smes. *Robotics and computer-integrated manufacturing*, 29(3):104–111.

Aken, J. E. v. (2004). Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. *Journal of management studies*, 41(2):219–246.

Alharahsheh, H. H. and Pius, A. (2020). A review of key paradigms: Positivism vs interpretivism. *Global Academic Journal of Humanities and Social Sciences*, 2(3):39–43.

Amini, M. and Abukari, A. M. (2020). Erp systems architecture for the modern age: A review of the state of the art technologies. *Journal of Applied Intelligent Systems and Information Sciences*, 1(2):70–90.

Ankrah, S. and Omar, A.-T. (2015). Universities–industry collaboration: A systematic review. *Scandinavian journal of management*, 31(3):387–408.

Annamalai, C. and Ramayah, T. (2011). Enterprise resource planning (erp) benefits survey of indian manufacturing firms: An empirical analysis of sap versus oracle package. *Business Process Management Journal*, 17(3):495–509.

Archer, M. (2013). *Social origins of educational systems*. Routledge.

Avison, D. E., Lau, F., Myers, M. D., and Nielsen, P. A. (1999). Action research. *Communications of the ACM*, 42(1):94–97.

Azungah, T. (2018). Qualitative research: deductive and inductive approaches to data analysis. *Qualitative research journal*, 18(4):383–400.

Babüroglu, O. N. and Ravn, I. (1992). Normative action research. *Organization Studies*, 13(1):019–34.

Baig, M. Z., Haq, M. U. U., Surkhail, H. M. U., Iqbal, R., and Sheikh, M. M. (2018). Bridging the industry-academia collaboration gap a focus towards final year projects. In *Proceedings of the 2nd International Conference on High Performance Compilation, Computing and Communications*, pages 40–44.

Baker, J. D. (1995). *Characterizing students' difficulty with proof by mathematical induction*. Indiana University.

Barnett, D. R. (2011). Partnering industry and education for curricular enhancement: A response for greater educational achievement. *Online Journal for Workforce Education and Development*, 5(2):5.

Barr, E. T., Harman, M., McMinn, P., Shahbaz, M., and Yoo, S. (2014). The oracle problem in software testing: A survey. *IEEE transactions on software engineering*, 41(5):507–525.

Barwise, J. and Etchemendy, J. (1998). *Computers, visualization, and the nature of reasoning*. na.

Baskerville, R. and Wood-Harper, A. T. (1998). Diversity in information systems action research methods. *European Journal of information systems*, 7:90–107.

Bass, L., Weber, I., and Zhu, L. (2015). *DevOps: A software architect's perspective*. Addison-Wesley Professional.

Becker, G. S. (2009). *Human capital: A theoretical and empirical analysis, with special reference to education*. University of Chicago press.

Bhaskar, R. (2013). *A realist theory of science*. Routledge.

Bibiano, L. H., Marco-Simó, J. M., and Pastor, J. A. (2014). An initial approach for improving crm systems implementation projects. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5):61–72.

Brooks, F. P. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE computer*, 20(4):10–19.

Bruce, K. B., Drysdale, R. L. S., Kelemen, C., and Tucker, A. (2003). Why math? *Communications of the ACM*, 46(9):40–44.

Brynjolfsson, E. and McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company.

Buchanan, R. (1992). Wicked problems in design thinking. *Design issues*, 8(2):5–21.

Bygstad, B. and Presthus, W. (2013). Social media as crm? how two airline companies used facebook during the "ash crisis" in 2010. *Scandinavian Journal of Information Systems*, 25(1):3.

Carlsson, S. A. (2003). Critical realism: a way forward in is research. In *ECIS*, pages 348–362.

Chang, S.-I., Gable, G., Smythe, E., and Timbrell, G. (2000). A delphi examination of public sector erp implementation issues.

Charniak, E. (1985). *Introduction to artificial intelligence*. Pearson Education India.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to algorithms*. MIT press.

Davison, R., Martinsons, M. G., and Kock, N. (2004). Principles of canonical action research. *Information systems journal*, 14(1):65–86.

DeMarco, T. and Lister, T. (2013). *Peopleware: productive projects and teams*. Addison-Wesley.

Dewey, J. (2022). *How we think*. DigiCat.

Doe, J. (2011). Recommended practice for software requirements specifications (ieee). *IEEE, New York*.

Dubinsky, E. (1989). Teaching mathematical induction ii. *Journal of Mathematical Behavior*, 8(3):285–304.

Easton, G. (2010). Critical realism in case study research. *Industrial marketing management*, 39(1):118–128.

Eden, A. H. (2007). Three paradigms of computer science. *Minds and machines*, 17:135–167.

Esteves, J. and Pastor, J. (2001). Enterprise resource planning systems research: an annotated bibliography. *Communications of the association for information systems*, 7(1):8.

Etzkowitz, H. and Leydesdorff, L. (2000). The dynamics of innovation: from national systems and "mode 2" to a triple helix of university–industry–government relations. *Research policy*, 29(2):109–123.

Fagerholm, F., Guinea, A. S., Münch, J., and Borenstein, J. (2014). The role of mentoring and project characteristics for onboarding in open source software projects. In *Proceedings of the 8th ACM/IEEE international symposium on empirical software engineering and measurement*, pages 1–10.

Flick, U., Von Kardorff, E., and Steinke, I. (2004). What is qualitative research? an introduction to the field. *A companion to qualitative research*, 1:3–11.

Foley, J. (1997). Technology transfer from university to industry. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 3–4.

for Standardization, I. O. (2011). *Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models*. ISO.

Fowler, M., Highsmith, J., et al. (2001). The agile manifesto. *Software development*, 9(8):28–35.

Gartner, G. (2009). Gartner exp worldwide survey of more than 1,500 cios shows it spending to be flat in 2009. *Disponible on line: http://www. gartner. com/newsroom/id/855612*.

Gibson, M., Arnott, D., Jagielska, I., and Melbourne, A. (2004). Evaluating the intangible benefits of business intelligence: Review & research agenda. In *Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain and Complex World*, pages 295–305. Prato, Italy.

Glass, R. L. (2006). *Software Creativity 2.0*. developer.* Books.

Goldkuhl, G. (2012). Pragmatism vs interpretivism in qualitative information systems research. *European journal of information systems*, 21:135–146.

Goodrich, M. T. and Tamassia, R. (2011). *Introduction to computer security*. Pearson London, UK.

Hawari, A. and Heeks, R. (2010). Explaining erp failure in developing countries. Technical report, Working Paper Series.

Hawking, P. and Sellitto, C. (2010). Business intelligence (bi) critical success factors.

Hawking, P., Stein, A., and Foster, S. (2004). Revisiting erp systems: benefit realization. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pages 8–pp. IEEE.

Heeks, R. (2017). *Information and communication technology for development (ICT4D)*. Routledge.

HELTON, D. A. (2012). Bridging the digital divide in developing nations through mobile phone transaction systems. *B¿ Quest*.

Hennessy, J. L. and Patterson, D. A. (2017). *Computer architecture: a quantitative approach*. Morgan kaufmann.

Herzum, P. (2003). Cutter consortium report on corporate use of bi and data warehousing technologies. *Cutter Consortium, Executive Report*, 4.

Hevner, A., Chatterjee, S., Hevner, A., and Chatterjee, S. (2010). Design science research in information systems. *Design research in information systems: theory and practice*, pages 9–22.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, pages 75–105.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. (1993). Interpretation as abduction. *Artificial intelligence*, 63(1-2):69–142.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65.

ISTQB, F. (2011). Foundation level syllabus version 2011. *International Software Testing Qualifications Board*.

Jacobs, F. R. et al. (2007). Enterprise resource planning (erp)—a brief history. *Journal of operations management*, 25(2):357–363.

Jacobson, L. and Booch, J. R. G. (2021). The unified modeling language reference manual.

James, W. (2020). Pragmatism. In *Pragmatism*, pages 53–75. Routledge.

Josephson, J. R., Chandrasekaran, B., Smith, J. W., and Tanner, M. C. (1987). A mechanism for forming composite explanatory hypotheses. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):445–454.

Kanellou, A. and Spathis, C. (2013). Accounting benefits and satisfaction in an erp environment. *International Journal of Accounting Information Systems*, 14(3):209–234.

Kenge, R. and Khan, Z. (2020). A research study on the erp system implementation and current trends in erp. *Shanlax International Journal of Management*, 8(2):34–39.

Knuth, D. E. (1968). Semantics of context-free languages. *Mathematical systems theory*, 2(2):127–145.

Kostojohn, S., Paulen, B., and Johnson, M. (2011). *CRM fundamentals*. Springer.

Lewis, C. and Mack, R. (1982). The role of abduction in learning to use a computer system.

Light, B., Holland, C. P., and Wills, K. (2001). Erp and best of breed: a comparative analysis. *Business Process Management Journal*, 7(3):216–224.

Littlejohn, S. W. and Foss, K. A. (2009). *Encyclopedia of communication theory*, volume 1. Sage.

Lowenthal, F. and Eisenberg, T. (1992). Mathematical induction in school: An illusion of rigor? *School Science and mathematics*, 92(5):233–238.

Luczak-Rösch, M. (2013). Beyond positivism in computer science.

Ly, A. (2020). The definitive guide to erp implementation.

Lynch, N. A. (1996). *Distributed algorithms*. Elsevier.

Manber, U. (1989). *Introduction to algorithms: a creative approach*. Addison-Wesley Longman Publishing Co., Inc.

March, S. T. and Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, 15(4):251–266.

Mardiana, S. (2020). Modifying research onion for information systems research. *Solid State Technology*, 63(4):5304–5313.

Mayer-Barber, K. D. and Yan, B. (2017). Clash of the cytokine titans: counter-regulation of interleukin-1 and type i interferon-mediated inflammatory responses. *Cellular & molecular immunology*, 14(1):22–35.

McConnell, S. (1996). *Rapid development: taming wild software schedules*. Pearson Education.

McConnell, S. (2004). *Code complete*. Pearson Education.

Monday.com (2024). Monday.com. Accessed on July 22, 2024.

Monk, E. F. and Wagner, B. J. (2013). *Concepts in enterprise resource planning*. Course Technology, Cengage Learning.

Movshovitz-Hadar, N. (1993). Mathematical induction: A focus on the conceptual framework. *School Science and Mathematics*, 93:408–408.

Mowery, D. C. and Sampat, B. N. (2006). Universities in national innovation systems.

Myers, G. J., Sandler, C., and Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.

Nah, F. F.-H., Lau, J. L.-S., and Kuang, J. (2001). Critical factors for successful implementation of enterprise systems. *Business process management journal*, 7(3):285–296.

Ojo, T. (2020). Political economy of ict4d and africa. In *Handbook of communication for development and social change*, pages 1243–1255. Springer.

O'Leary, D. E. (2000). *Enterprise resource planning systems: systems, life cycle, electronic commerce, and risk*. Cambridge university press.

Olszak, C. M. and Ziemba, E. (2007). Approach to building and implementing business intelligence systems. *Interdisciplinary Journal of Information, Knowledge, and Management*, 2(1):135–148.

O'Brien, R. (1998). An overview of the methodological approach of action research.

O'Regan, G. (2015). Sap se. In *Pillars of Computing: A Compendium of Select, Pivotal Technology Firms*, pages 189–194. Springer.

Page, R. L. (2003). Software is discrete mathematics. *ACM SIGPLAN Notices*, 38(9):79–86.

Park, Y. S., Konge, L., and Artino Jr, A. R. (2020). The positivism paradigm of research. *Academic medicine*, 95(5):690–694.

Pearse, N. (2019). An illustration of deductive analysis in qualitative research. In *18th European conference on research methodology for business and management studies*, page 264.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Pelphrey, M. W. (2015). *Directing the ERP implementation: A best practice guide to avoiding program failure traps while tuning system performance*. CRC Press.

Perkmann, M., Tartari, V., McKelvey, M., Autio, E., Broström, A., D'este, P., Fini, R., Geuna, A., Grimaldi, R., Hughes, A., et al. (2013). Academic engagement and commercialisation: A review of the literature on university–industry relations. *Research policy*, 42(2):423–442.

Pierce, B. C. (2002). *Types and programming languages*. MIT press.

Polycarpou, I., Pasztor, A., and Adjouadi, M. (2008). A conceptual approach to teaching induction for computer science. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pages 9–13.

Pople, H. E. (1973). On the mechanization of abductive logic. In *IJCAI*, volume 73, pages 147–152. Citeseer.

Porter's, V. C. M. (1985). What is value chain. *E-Commer.*, pages 1–13.

Potrac, P., Jones, R. L., and Nelson, L. (2014). Interpretivism. In *Research methods in sports coaching*, pages 31–41. Routledge.

Power, D. (1998). Justifying a data warehouse project. *Part I and II. DS*, The On-Line Executive Journal for Data-Intensive Decision Support*, 2(5).

Pressman, R. S. (2005). Software engineering: a practitioner's approach. *Pressman and Associates.*

Pries-Heje, J., Baskerville, R., and Venable, J. (2007). Soft design science research: Extending the boundaries of evaluation in design science research. In *Proceedings from the 2nd International Conference on Design Science Research in IT (DESRIST)*, pages 18–38.

Reggia, J. A., Nau, D. S., and Wang, P. Y. (1983). Diagnostic expert systems based on a set covering model. *International journal of man-machine studies*, 19(5):437–460.

Rittel, H. W. and Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169.

Rom, A. and Rohde, C. (2006). Enterprise resource planning systems, strategic enterprise management systems and management accounting: A danish study. *Journal of enterprise information management*, 19(1):50–66.

Ronchi, A. M. (2009). *eCulture: cultural content in the digital age.* Springer Science & Business Media.

Ross, J. W., Weill, P., and Robertson, D. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution.* Harvard business press.

Ryan, G. (2018). Introduction to positivism, interpretivism and critical theory. *Nurse researcher*, 25(4):41–49.

Sankar, C. and Rau, K.-H. (2006). *Implementation Strategies for SAP R/3 in a Multinational Organization: Lessons from a Real-World Case Study: Lessons from a Real-World Case Study.* IGI Global.

Saunders, M., Lewis, P., and Thornhill, A. (2003). Research methods forbusiness students. *Essex: Prentice Hall: Financial Times.*

Saunders, M., Lewis, P., and Thornhill, A. (2009). *Research methods for business students.* Pearson education.

Sayer, R. A. (1992). *Method in social science: A realist approach.* Psychology Press.

Schein, E. H. (2010). *Organizational culture and leadership*, volume 2. John Wiley & Sons.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. (2011). Action design research. *MIS quarterly*, pages 37–56.

Shanks, G. (2000). A model of erp project implementation. *Journal of information Technology*, 15(4):289–303.

Shields, M. G. (2004). *E-business and ERP: Rapid implementation and project planning.* John Wiley & Sons.

Smilansky, O. (2015). How to select a crm system: whether it's your first solution or your fourth, these tips will help make the vendor selection process easier. *CRM Magazine*, 19(4):28–32.

Smyth, R. (1999). Peirce, pragmatism and the logic of scripture.

Solutions, P. (2016). report on erp systems and enterprise software. 1-32.

Sommer, D. (2008). Gartner says worldwide business intelligence spending to grow 11 percent in 2008. *located at www. gartner. com, accessed February.*

Sommerville, I. (2011). Software engineering (ed.). *America: Pearson Education Inc.*

Spathis, C. and Constantinides, S. (2004). Enterprise resource planning systems' impact on accounting processes. *Business Process management journal*, 10(2):234–247.

Susman, G. I. (1983). Action research: a sociotechnical systems perspective. *Beyond method: Strategies for social research*, 95(113):95.

Susman, G. I. and Evered, R. D. (2023). An assessment of the scientific merits of action research. *STUDI ORGANIZZATIVI*, (2022/2).

Tarafdar, M. and Roy, R. K. (2003). Analyzing the adoption of enterprise resource planning systems in indian organizations: a process framework. *Journal of Global Information Technology Management*, 6(1):21–51.

TaskFlow (2024). Taskflow.co.za. Accessed on July 22, 2024.

Team, C. (2002). Capability maturity model® integration (cmmi), version 1.1–continuous representation.

Terminanto, A., Hidayat, R., and Hidayanto, A. N. (2017). Implementation of enterprise resource planning using odoo module sales and crm. case study: Pt ecosains hayati. In *IOP Conference Series: Materials Science and Engineering*, volume 277, page 012034. IOP Publishing.

Tsai, B.-H. (2008). The impact of enterprise resource planning systems on the efficiency of taiwanese firms. In *2008 IEEE Asia-Pacific Services Computing Conference*, pages 1441–1446. IEEE.

Tschang, T. (2001). The basic characteristics of skills and organizational capabilities in the indian software industry. Technical report, ADBI Research Paper Series.

Turley, R. T. and Bieman, J. M. (1994). Identifying essential competencies of software engineers. In *ACM Conference on computer science*, volume 10.

Uçaktürk, A. and Villard, M. (2013). The effects of management information and erp systems on strategic knowledge management and decision-making. *Procedia-Social and Behavioral Sciences*, 99:1035–1043.

Vukovic, V., Gagić, N., Raković, L., and Marić, S. (2023). Erp systems for small and medium-sized enterprises from the sap and microsoft perspective. In *International Scientific Conference Strategic Management and decision Support systems in Strategic Management*, pages 333–339.

Wadhwani, R. D., Bucheli, M., et al. (2014). The future of the past in management and organization studies. *Organizations in time: History, theory, methods*, 4.

Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. (1992). Building an information system design theory for vigilant eis. *Information systems research*, 3(1):36–59.

Weiss, M. A. (1998). Data structures and problem solving using java. *ACM SIGACT News*, 29(2):42–49.

Wikgren, M. (2005). Critical realism as a philosophy and social theory in information science? *Journal of documentation*, 61(1):11–22.

Winter, R. (1987). Action-research and the nature of social inquiry: Professional innovation and educational work. *(No Title)*.

Wu, C.-C., Dale, N. B., and Bethel, L. J. (1998). Conceptual models and cognitive learning styles in teaching recursion. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, pages 292–296.

Xu, Y., Rahmati, N., and Lee, V. C. (2008). A review of literature on enterprise resource planning systems. In *2008 International Conference on Service Systems and Service Management*, pages 1–6. IEEE.

Zhang, L., Lee, M. K., Zhang, Z., and Banerjee, P. (2003). Critical success factors of enterprise resource planning systems implementation success in china. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pages 10–pp. IEEE.

Zhao, B. and Tu, C. (2021). Research and development of inventory management and human resource management in erp. *Wireless Communications and Mobile Computing*, 2021(1):3132062.

Zoho (2024). Zoho.com. Accessed on July 22, 2024.