

# Startup Meets Big Systems

A Case Study of Veo Technologies

Master's Thesis 2025

MSc. In Business Administration and Digital Business

Copenhagen 15.5.2025

NUMBER OF CREDITS: 30 ECTS

REFERENCE SYSTEM: APA HARVARD

SUPERVISOR: Jonas Hedman

CHARACTERS: 168920

AUTHOR: Thomas Agerbæk Ruby (167890)

PAGES: 75

## Acknowledgements

I would like to acknowledge and express sincere gratitude towards all parts involved and their contributions to my master's thesis.

First, I would like to thank my supervisor Jonas Hedman at Copenhagen Business School for providing the valuable guidance, discussions and feedback he did throughout the process.

Secondly, I would like to thank Veo Technologies, Subscription Factory and all participants involved in my master's thesis. Also a thank you to the 6 interviewed people for taking their time to share their knowledge and experience they had from working on the Fosbury Project. Without you the thesis would not have been possible.

Lastly, I would like to thank friends, family, and colleagues who have offered support and shown interest throughout the study, this has contributed to valuable insights and perspectives.

A handwritten signature in black ink, appearing to read "Thomas Agerbæk Ruby". The signature is fluid and cursive, with "Thomas" and "Ruby" being more clearly legible than "Agerbæk".

---

Thomas Agerbæk Ruby

# Abstract

This thesis examines how a high-growth tech startup aligns internal IT systems with rapid expansion, a crucial yet complex process. Addressing *How does a tech startup balance its internal IT systems with its growth?* This thesis provides a longitudinal case study of Veo Technologies' "Fosbury Project" (a major IT transition project) to explore challenges, decision-making, governance evolution, and organizational learning.

Employing an interpretivist philosophy and inductive approach, the research uses qualitative data from semi-structured interviews and project documents. Findings are interpreted via thematic analysis, guided by a framework integrating organizational change theories (dialectical, teleological), bounded rationality, and the Garbage Can Model.

Veo's IT maturation process was a complex three-phase process ('Child,' 'Teenager,' 'Adult'). The 'Child Stage' featured ambitious teleological goals undermined by immaturity, reactive decisions, poor governance, and consultant over-reliance. The 'Teenager Stage' saw dialectical conflicts, operational breakdowns, and governance collapse. Crisis led to the 'Adult Stage,' involving restructuring (Business-IT absorbed by Builders), resource mobilization ('Subsavers'), and the strategic 'Zuora 2.0' development, reflecting learned maturity and teleological IT alignment.

Balancing IT with startup growth is an emergent, conflict-driven process. Initial imbalances from immaturity and pressured decisions are often rectified through crisis, learning, and reactive development of detailed governance and IT strategies. This research empirically aids understanding of IT transformation in high-velocity settings, offering practical insights.

**Keywords:** *IT Alignment, Startup Growth, Organizational Change Theory, IT Governance, Case Study, Technology Startups, IT Transformation, Scalability, Decision-Making Dynamics*

# Table of Contents

<b>1. Introduction .....</b>	<b>6</b>
1.2 Problem Statement.....	7
1.3 Significance and Contribution of the Study.....	8
1.4 Scope and Delimitations .....	8
1.5 Structure of the Thesis.....	9
<b>2. Literature Review.....</b>	<b>9</b>
2.1 IT System Change Challenges in High-Growth Environments.....	10
2.1.1 The Necessity of Scalability in Rapidly Expanding Startups.....	10
2.1.2 Navigating System Integration Complexities.....	11
2.1.3 Legacy Systems in Agile Environments.....	12
2.1.4 Root Causes and Strategic Mitigation of IT Change Challenges .....	12
2.2 Decision-Making Dynamic in Scaling Tech Firms .....	14
2.2.1 The Dominance of Data-Driven and Evidence-Based IT Decisions .....	14
2.2.2 IT Governance and Strategic Frameworks in Scaling Contexts.....	15
2.2.3 Key Actors and Influencers in IT Decision-Making .....	16
2.2.4 Agile Methodologies and Adaptive Decision-Making .....	17
2.2.5 How IT Challenges and Decisions Influence Each Other .....	17
2.3 Gaps in Existing Research and Future Directions .....	18
2.3.1 Unexplored Facets of IT Challenges Across Startup Growth Trajectories .....	18
2.3.2 Methodological Limitations and Opportunities for Future Inquiry .....	18
2.3.3 Integrating IT, Decision Dynamics, and Growth .....	18
<b>Chapter 3: Theoretical Framework.....</b>	<b>19</b>
3.1 Introduction: Framing IT System Replacement in High-Growth Environments .....	19
3.2 Primary Lens: Organizational Change Theories.....	20
3.2.1 Life-Cycle Models of Organizational Change.....	20
3.2.2 Evolutionary Theories of Change .....	20
3.2.3 Teleological Theories of Change.....	21
3.2.4 Dialectical Theory of Organizational Change .....	21
3.2.5 From Conflict to Direction: How the Fosbury Project Shifted between Dialectical to Teleological Change .....	22
3.3 Supporting Lenses: Understanding Decision-Making Under Pressure .....	23
3.3.1 Bounded Rationality in High-Growth Firms .....	24
3.3.2 Ambiguous Decision Processes: The Garbage Can Model .....	24
3.4 Integrating Perspectives: IT Governance in a Dynamic Context .....	25
3.5 Subconclusion.....	25
<b>4. Methodology .....</b>	<b>26</b>
4.1 Introduction to Case Company: Veo Technologies.....	27
4.1.1 Growth Trajectory and Strategic Imperative for IT Transformation.....	28
4.2 Research Philosophy and Approach .....	29

4.3 Research Strategy .....	29
4.4 Methodological Choice.....	30
4.5 Data Collection and Analysis .....	30
4.5.1 Sampling Strategy and Participant Selection.....	31
4.5.2 Primary Data Collection: Semi-Structured Interviews.....	31
4.5.3 Secondary Data and Triangulation .....	32
4.5.4 Thematic Data Analysis.....	33
4.5.5 Interpretation and Narrative Construction.....	35
4.5.6 Use of Generative Artificial Intelligence (GenAI).....	35
4.6 Ethical Considerations .....	36
4.7 Methodological Limitations.....	37
<b>5. Findings.....</b>	<b>37</b>
<b>5.1 Phase 1 - Strategic Framing (Child Stage) .....</b>	<b>38</b>
5.1.1 Organizational Structure at the Time of Implementation.....	39
5.1.2 The Pre-Fosbury Setup .....	41
5.1.3 Choosing The New Systems.....	43
5.1.4 Utilizing External Consultants.....	44
5.1.5 Fosbury System Design vs. Zuora Best Practice .....	46
5.1.6 The Importance of a Steering Committee.....	48
5.1.7 A Black Box: Lack of Knowledge Transfer and Training .....	49
5.1.8 Sub-Conclusion: Phase 1 .....	50
<b>5.2 Phase 2 - Emergence of Contradictions (Teenager Stage).....</b>	<b>51</b>
5.2.1 Breakdown of Lacking Governance Structures.....	52
5.2.2 Instability and Forced Change .....	54
5.2.3 Internal Conflicts and Leadership Fragmentation .....	55
5.2.4 Capability Gaps and Training Failures .....	57
5.2.5 Operational Misalignment and System Fragility .....	58
5.2.6 Emergence of Ownership and Role Conflicts .....	59
5.2.7 Sub-Conclusion: Phase 2 .....	61
<b>5.3 Phase 3 – Adulthood: Stabilization and Zuora 2.0 .....</b>	<b>62</b>
5.3.1 Crisis Management and Reactive Actions.....	62
5.3.2. Subsavers .....	64
5.3.3 Formation of New Teams .....	64
5.3.4 Design Principles of Zuora 2.0 .....	65
5.3.5 Sub-Conclusion: Phase 3 .....	67
<b>6. Discussion.....</b>	<b>67</b>
6.1 Interpretation of Results .....	68
6.1.2 Interpretation Themes.....	70
6.3 Implications .....	74
6.3.1 Theoretical Implications .....	74
6.3.2 Practical Implications .....	75

6.4 Limitations .....	77
6.5 Future Research Directions.....	78
<b>7. Conclusion.....</b>	<b>80</b>
<b>Bibliography .....</b>	<b>82</b>
<b>Appendices .....</b>	<b>88</b>
Appendix 1.....	88
Appendix 2.....	90
Appendix 3.....	92
Appendix 4.....	104
Interview 1 .....	104
Interview 2 .....	114
Interview 3 .....	128
Interview 4.....	137
Interview 5.....	145
Interview 6.....	154
Interview 7 .....	165

# 1. Introduction

What determines whether a company succeeds? Is it strong branding, an innovative product, access to funding, or, as some argue, the often overlooked backbone of the organization: functional and scalable IT systems? There is little doubt that tech startups face numerous challenges, with the most immediate being the need to secure revenue to ensure short-term survival.

The path for tech startups is notoriously difficult, with most failing within their first few years. Research shows that over 90% of tech startups fail (Santisteban et al., 2023), with about 70% shutting down between their second and fifth year (Tamzid, 2025). Other studies suggest that only one in three IT startups survive past three years (Santisteban & Mauricio, 2017). While factors like founder characteristics, product-market fit, and financial management play major roles (Santisteban et al., 2023), the importance of having robust and adaptable IT systems is often overlooked. Indeed, even beyond the startup world, large-scale IT projects are filled with risk, often running significantly over budget and failing to deliver their promised value due to issues like unclear objectives, inadequate governance, or poor execution (Bloch et al., 2012). Poor IT decisions can cause serious issues, including bottlenecks, wasted resources, and organizational chaos (Tamzid, 2025). This is especially true for tech startups, where IT is not just a support function but core to the company's operations and value.

This report analyzes how Veo Technologies, a tech startup, handled the challenge of aligning its internal IT systems with organizational growth, with its Fosbury Project serving as the central case study. The analysis employs a three-phase dialectical model metaphorically based on human growth: the Child, Teenager, and Adult phases, to trace the evolution of Veo's commercial IT systems, governance, and organizational learning throughout this project. The central issue explored is the common struggle startups face in scaling their IT infrastructure, processes, and governance alongside rapid business growth, which can lead to inefficiencies, vulnerabilities, stalled progress, or even failure, all challenges that are strongly illustrated by Veo's journey. The complexities inherent in such large-scale IT transformations often mirror those seen in larger enterprises, which also struggle with managing scope, ensuring strategic alignment, and building capable teams, all critical elements for success (Bloch et al., 2012). This report reviews academic

and industry research to underscore how critical this balance is, utilizing the phased developmental approach to explain the complexities involved as demonstrated in the Veo case.

## 1.2 Problem Statement

High-growth technology startups operate under intense pressure to scale, yet face notoriously high failure rates (Santisteban et al., 2023). Successfully navigating this phase often relies on critical success factors, among which the effective alignment of internal IT systems, such as infrastructure, processes, and governance with rapid organizational growth is key (Santisteban & Mauricio, 2017). However, achieving this alignment presents significant challenges (Ashurbaev & Sadullaev, 2024). Initial IT decisions, often made under resource constraints or prioritizing speed, can quickly lead to technical debt and systems less suited for scale, while IT governance frequently remains underdeveloped or struggles to mature alongside the organization (Tamzid, 2025). This misalignment between IT capabilities and business needs can create operational bottlenecks, hinder innovation, and ultimately contribute to failure (Szathmári et al., 2024).

While literature outlines IT best practices, governance frameworks, and the consequences of failure during large-scale IT projects (Bloch et al., 2012; Serrano et al., 2021), the *actual process* by which scaling startups navigate the complexities of major IT transformations remains less understood. This journey is rarely linear; it is often characterized by emergent crises decision-making under uncertainty, and shaped by dialectical tensions between competing priorities, existing structures, and the need for change (De Keyser et al., 2019). For a scaling startup, how these planned change efforts handle or are challenged by internal conflicts becomes a critical test, shaping not only the project's outcome, but also the organization's ability to learn, adapt, and survive. There is a gap in empirically grounded understanding of *how* tech startups dynamically manage this balancing act in practice, adapting their systems and governance through phases of stability, conflict, and learning.

Given the critical importance of IT alignment for startup survival and success, and the complex, dynamic nature of achieving it during rapid growth, a deeper investigation into this process is needed. Therefore, this thesis seeks to address the research question:

❖ ***How does a tech startup balance its internal IT systems with its growth?***

The research investigates this question through an in-depth longitudinal case study of Veo Technologies' Fosbury Project, analyzing the challenges, decisions, governance evolution, and organizational learning involved in its attempt to overhaul core commercial IT systems through significant expansion.

### **1.3 Significance and Contribution of the Study**

By addressing the research question: *How does a tech startup balance its internal IT systems with its growth?*, through the detailed case study of Veo Technologies' Fosbury Project, this thesis aims to make several contributions. Theoretically, it seeks to provide empirical insights into the dynamic, often dialectical, processes of IT system evolution and governance adaptation within rapidly scaling startups, potentially extending existing models of organizational change and IT management in high-velocity environments. Practically, the findings may offer valuable lessons and a nuanced understanding for founders, managers, and IT leaders in other tech startups facing similar challenges of aligning technology with growth. The detailed tracing of Veo's journey through its 'Child, Teenager, and Adult' phases can illuminate common pitfalls, critical decision points, and the role of organizational learning in navigating such transformations.

### **1.4 Scope and Delimitations**

The scope of this research is focused on a single, in-depth, longitudinal case study of Veo Technologies, a Danish tech startup, and its specific internal IT transformation initiative known as the Fosbury Project. The analysis primarily examines the alignment of its commercial IT systems with organizational growth, utilizing a qualitative approach and drawing on data collected through methods such as semi-structured interviews with key stakeholders (including technical staff, management, and external consultants involved in the project), and the analysis of internal project documentation and timelines, spanning the period from December 2024 to May 2025. Key delimitations include the inherent limitations of a single-case study design, meaning the findings aim for analytical rather than statistical generalization. The study is situated within the context of a specific company in the tech industry, and its unique organizational culture, leadership, and

market conditions will have influenced the observed phenomena. Furthermore, the retrospective nature of some data collection may be subject to recall bias, and the researcher's interpretation, though grounded in established theoretical frameworks, plays a role in the analysis.

## 1.5 Structure of the Thesis

This thesis is structured as follows: **Chapter 1 (Introduction)**. **Chapter 2 (Literature Review)** will delve into existing academic and industry research on IT systems in startups, challenges of scalability and integration, IT governance, decision-making dynamics in scaling firms, and relevant theories of organizational change. **Chapter 3 (Theoretical Framework)** will outline the primary theoretical lenses used for the analysis, focusing on dialectical and teleological theories of change, supplemented by concepts from bounded rationality and the Garbage Can Model. This chapter will also introduce and explain the conceptual process model (Dialectical Process Model) developed for this thesis, which synthesizes these perspectives to guide the analysis of Veo's IT transformation journey. **Chapter 4 (Methodology)** will detail the research design, including the case study approach, data collection methods, and the procedures for data analysis employed in this study. **Chapter 5 (Findings)** will present the empirical findings from the Veo Technologies case study, tracing the evolution of the Fosbury Project through the 'Child,' 'Teenager,' and 'Adult' phases. **Chapter 6 (Discussion)** will interpret these findings in relation to the research question and theoretical framework, discussing the key themes emerging from Veo's journey and their implications. **Chapter 7 (Conclusion)** aims to summarize the main findings, reiterate the contributions of the study, acknowledge its limitations, and suggest directions for future research.

## 2. Literature Review

Information Technology (IT) systems are fundamental to the operational capabilities and strategic trajectory of modern startups. As these new businesses begin on paths of growth and scaling, their IT infrastructure, applications, and overall technology strategy must co-evolve. This evolution, however, is rarely seamless. It presents a complex array of opportunities and significant challenges that influence managerial decision-making and the sustainability and success of the startup. This

review examines IT system change challenges in high-growth contexts, the dynamics of IT-related decision-making within scaling firms, and identifies possible gaps in existing research.

## **2.1 IT System Change Challenges in High-Growth Environments**

Rapid expansion uniquely pressures startup IT systems, demanding constant adaptation that can expose or amplify vulnerabilities across technical, strategic, operational, financial, and human resource dimensions.

### **2.1.1 The Necessity of Scalability in Rapidly Expanding Startups**

Scalability, an IT system's capacity to handle an increasing workload or its potential to be expanded to accommodate such growth (Paula & Zilber, 2023), is critical for startups experiencing rapid expansion. As the customer base broadens and transaction volumes and business rules develop, the underlying IT infrastructure must maintain performance and reliability without breakdowns (Orugunta, 2025). This necessity introduces significant architectural, performance, and cost implications. Startups need to design systems that can handle increased workloads without slowing down performance or causing major increases in operating costs (Hanifzadeh et al., 2024). This often involves forecasting future technological needs based on overarching business objectives (*Ibid.*).

Certain technologies, like public blockchains, bring specific scalability challenges, such as slow transaction speeds, delays in processing, and large storage needs (Alkhateeb et al., 2021). While these examples are specific, the general principles of throughput, latency, and storage are universal concerns in IT scalability. To mitigate some of these pressures, cloud computing solutions are frequently adopted (Paula & Zilber, 2023). These offer reduced initial infrastructure costs and enhanced system availability, providing the flexibility to scale resources up or down in response to demand (Orugunta, 2025) . However, the management of cloud-related spending can itself become a significant challenge if not actively optimized. Infrastructures that are poorly designed from a scalability perspective can lead to diminished performance during peak traffic periods, system failures, an incapacity to manage increased data volumes, and difficulties in adding new

users or applications, all of which can significantly restrain a startup's growth trajectory (Iyer, et al., 2005).

The pursuit of scalability, while essential, is not without its challenges. A critical consideration is the timing and planning of scaling efforts. Investing heavily in highly scalable IT architecture before establishing product-market fit or validating the core business model can lead to a premature drain on resources (Lee & Kim, 2024). Research indicates that early scaling, in general business terms which includes aspects like premature hiring, is associated with a higher likelihood of startup failure (Kumar & Kalse 2022). This phenomenon can be partly attributed to cognitive biases where entrepreneurs, driven by ambition or external pressures, rush growth before foundational elements are secure (Lee & Kim, 2024). If a startup commits significant capital and effort towards building a sophisticated, scalable IT infrastructure without first confirming that there is a sustainable demand for its product or service, it incurs substantial financial and opportunity costs. These costs can accelerate the company's death if anticipated revenue streams do not materialize. The concept of defining a "unit of scale", demonstrating that the business model is viable and repeatable on a smaller scale before committing to large-scale IT investments, is therefore of critical importance (Forbes Technology Council, 2024).

### **2.1.2 Navigating System Integration Complexities**

As startups mature, they incorporate a growing number of specialized software tools and IT systems, and growth often involves collaborations with external partners. This increasing complexity requires strong system integration for seamless data flow, operational efficiency, and a cohesive user experience (Bajgoric & Moon, 2009). Integration challenges often arise when startups attempt to connect their systems with those of external partners. Tight timelines and incompatible technologies can strain strategic alliances (Saunders et al., 2002). Addressing these issues frequently requires comprehensive business process re-engineering. To facilitate smoother integration, adopting tools with open Application Programming Interfaces (APIs) and setting clear integration goals is essential (Bajgoric & Moon, 2009).

Underestimation of integration challenges can lead to "islands of automation" where unconnected systems operate in isolation, resulting in data silos that restricts a unified view of business

operations (Saunders et al., 2002; Juneja & Klinger, 2024). These silos impede agile, data-informed decisions, undermining the intended efficiencies of IT investments and impacting data accessibility, decision quality, and organizational agility.

### **2.1.3 Legacy Systems in Agile Environments**

Even in startups, early technology choices made under resource constraints or for rapid deployment can quickly become outdated, exhibiting characteristics of "early-onset" legacy systems that hinder agility and innovation (Wadsworth, 2025; PremierNX, 2025). Outdated systems can consume a large share of IT budgets (e.g., up to 75% for maintenance), limiting resources for innovation (PremierNX, 2025). These systems often lack compatibility with modern applications, leading to integration complexities, increased operational costs, and delayed project timelines. Common traits include limited functionality, inefficiencies, data silos, heightened security vulnerabilities, and poor scalability (Wadsworth, 2025). Despite these drawbacks, startups may resist addressing these components due to the perceived cost and complexity of digital transformation (Nadkarni & Prügl, 2020; Wadsworth, 2025).

This issue is often a result of the "move fast and break things" startup culture. When not balanced with strategic IT planning, initial IT solutions are often chosen based on rapid deployment or low initial cost without considering long-term scalability, maintainability, or interoperability (PremierNX, 2025). As rapid growth occurs, these hastily implemented systems, though suitable initially, quickly become bottlenecks and are rendered "legacy" relative to new operational demands, creating tension between the immediate need for speed and the long-term requirement for a sustainable and adaptable IT architecture.

### **2.1.4 Root Causes and Strategic Mitigation of IT Change Challenges**

Common root causes for IT challenges in startups include a lack of strategic IT planning, resource misallocation, unmanaged growth, poor communication, change resistance, and inadequate technical expertise (Bridgen, 2025; PremierNX, 2025; Saunders et al., 2002). These strategies, which address various organizational dimensions, can be conceptualized as key interconnected components for effectively navigating the complexities of IT evolution as illustrated in Figure 1.



**Figure 1:** Key Strategies for Navigating IT Challenges in High-Growth Startups

Figure 1 outlines six crucial strategic areas that contribute to proactively managing and mitigating IT system change challenges in rapidly scaling startup environments. Effective IT management in startups involves aligning IT strategy with business goals, ensuring active involvement from senior management (Molete et al., 2025). It also includes forming strategic partnerships with external vendors, supported by thorough due diligence (Saunders et al., 2002). Clear process management, early risk planning, and phased implementations help mitigate project risks (Stepanov, 2024; PremierNX, 2025). Resource management strategies, such as leasing IT equipment, can ease financial pressures (Saunders et al., 2002), while strong security and compliance measures should be established from the outset (Dekkers, 2024). Finally, investing in employee training and nurturing a culture open to innovation are essential for sustainable IT transformation (*Ibid.*).

Collectively, these approaches underscore a shift from reactive problem-solving towards a more integrated and forward-looking stance. Successfully navigating the IT landscape during high growth requires a judicious balance of strategic foresight with agile execution.

## **2.2 Decision-Making Dynamic in Scaling Tech Firms**

As technology firms navigate scaling, their decision-making concerning IT systems and strategy undergoes significant transformation, often shifting towards more structured, data-informed approaches, influenced by diverse actors and guided by various frameworks

### **2.2.1 The Dominance of Data-Driven and Evidence-Based IT Decisions**

There is a pronounced emphasis on leveraging data analytics, Management Information Systems (MIS), and Business Intelligence (BI) to inform IT and strategic choices, moving from intuition to data-driven approaches (Mokogwu et al., 2024; Paula & Zilber, 2023). Data-driven decision-making (DDDM) involves systematically using insights from data analytics to guide strategic actions, with research suggesting firms effectively harnessing data achieve superior performance (Mokogwu et al., 2024). MIS transforms raw data into actionable insights, supporting strategic decisions through real-time analytics and predictive modeling (Hamdat et al., 2024). Advanced AI, like Intelligent Choice Architectures (ICAs), aims to create and refine choices for decision-makers (Ransbotham et al., 2025).

Despite aspirations, a significant "data-action gap" often exists due to challenges in merging contrasting data sources, real-time processing, and a lack of tools or analytical skills to translate data into actionable insights (Juneja & Klinger, 2024). Many organizations struggle to use data continuously at scale, leading to continued reliance on intuition. Arguably, the "dominance" of data-driven decision-making is perhaps better characterized as an ongoing journey with practical hurdles

### **2.2.2 IT Governance and Strategic Frameworks in Scaling Contexts**

To structure IT decision-making, governance, and strategy, scaling companies often turn to established and emerging frameworks (Drechsler & Weißschädel, 2017). These frameworks serve

various purposes, addressing different layers and aspects of IT management, from overarching enterprise governance to specific project methodologies and operational practices. Figure 2 provides a categorized overview of some prominent types of frameworks relevant in such contexts:

Enterprise Gov & Architecture	Service Mgmt & Operations	Development & Delivery Agility	Decision/Improve. Tools	Revenue/Commercial
<ul style="list-style-type: none"> <li>• COBIT</li> <li>• TOGAF</li> </ul>	<ul style="list-style-type: none"> <li>• ITIL</li> </ul>	<ul style="list-style-type: none"> <li>• Agile FWs</li> <li>• DevOps-Based FWs</li> </ul>	<ul style="list-style-type: none"> <li>• OODA, PDCA</li> <li>• DECIDE, K-T</li> <li>• SWOT</li> </ul>	<ul style="list-style-type: none"> <li>• RevOps</li> </ul>

Figure 2: Categories of IT Governance and Strategic Frameworks

As shown in Figure 2, one key category is **Enterprise Governance & Architecture**, with frameworks like COBIT focusing on IT governance, alignment with business goals, risk management, and compliance, and TOGAF providing modular enterprise architecture models suitable for scaling (Andiso Pingilili et al., 2025; Drechsler & Weißschädel, 2017). **Service Management & Operations** is addressed by ITIL, offering best practices for aligning IT services with business needs and driving continuous improvement (Andiso Pingilili et al., 2025; Drechsler & Weißschädel, 2017). **Development & Delivery Agility** includes Agile and DevOps-based frameworks, enabling iterative development, collaboration, and rapid delivery (Drechsler & Weißschädel, 2017; Uludağ et al., 2022). **Decision/Improvement Tools** like the OODA Loop, PDCA, and SWOT Analysis support decision-making, agility, and continuous improvement (Mottola, 2021; Antonio et al., 2024). For **Revenue/Commercial Alignment**, RevOps frameworks guide technology choices that support sales and customer success (Mottola, 2021; Antonio et al., 2024).

Scaling firms often combine elements from these frameworks, though integration remains a challenge (Highspot, 2025). Beyond specific operational frameworks, effective IT decision-making fundamentally relies on robust **IT governance**. Defined by Weill & Ross (2004) as "the framework for decision rights and accountabilities to encourage desirable behavior in the use of IT," its effectiveness hinges on intentional design encompassing key decision areas like IT

principles, architecture, infrastructure, application needs, and investment prioritization (Weill & Ross, 2004; Maur et al., 2009). Modern perspectives on IT governance emphasize adaptability and contextual alignment over rigid, one-size-fits-all structures (Brown & Grant, 2005; Sambamurthy & Zmud, 1999). Peterson (2003) advocated for "crafting" adaptable governance through structural, processual, and relational mechanisms, recognizing that approaches may need to be hybrid (Brown & Magill, 1994; Olson & Chervany, 1980). For high-growth firms, such as the focus of this thesis, the evolution of governance from informal to more formal, yet still adaptable, structures is crucial for managing the "growing pains" associated with scaling (Sloof et al., 2018).

### **2.2.3 Key Actors and Influencers in IT Decision-Making**

IT decisions in scaling firms are shaped by diverse internal and external stakeholders (Forbes Technology Council, 2024). The strategic role of the Chief Technology Officer (CTO) and executive leaders is important, as these decisions often carry high opportunity costs and significantly impact strategic trajectory (Saunders et al., 2002; Herstatt et al., 2006). The CTO typically develops and executes the technology roadmap, oversees IT systems, manages tech risks, and leads the technical team, ensuring IT infrastructure supports rapid growth (Herstatt et al., 2006). Early recruitment of experienced IT leadership is crucial (Forbes Technology Council, 2024).

Investor expectations and involvement also exert considerable influence, often providing critical funding for IT investments (Santisteban & Mauricio, 2017). Founders are advised to cultivate an "investor mindset" (*Ibid.*). However, investment from tech giants can sometimes negatively impact future funding prospects or lead to patent domination (Huang, 2025). Also, investors balance data-driven analysis with intuition (Borlován, 2025), therefore, potential tensions can arise between founders' long-term vision, operational IT needs, and investors' focus on rapid growth and ROI, with the CTO often navigating these conflicting pressures.

### **2.2.4 Agile Methodologies and Adaptive Decision-Making**

Agile principles are increasingly applied broadly to business strategy and IT decision-making in scaling firms to enable rapid adaptation (Drechsler & Weißschädel, 2017). Scaling Agile effectively involves extending principles beyond development teams, requiring significant cultural

shifts, work management adjustments, and supporting technologies (Uludağ et al., 2022). Common challenges include cultural resistance, lack of experience with agile at scale, and managing dependencies (Ibid.). Success requires clear roles, customer-centricity, consistency, and executive buy-in (Christiaan Verwijs & Russo, 2024). True agility involves including adaptive capabilities at a strategic level, allowing rapid pivots in IT architecture, investment, and technology choices in response to market feedback and evolving business models (PDF, Source 4).

### **2.2.5 How IT Challenges and Decisions Influence Each Other**

A dynamic, symbiotic relationship exists between IT challenges and decision-making processes. IT challenges (scalability, integration, security) directly influence the decisions leadership must confront, while decision-making approaches can either exacerbate or mitigate these challenges (Ciacci et al., 2025). Data-driven decision-making is crucial for reducing complexity and deriving actionable insights (Ciacci et al., 2025; Mokogwu et al., 2024). MIS capabilities empower managers to make informed decisions more quickly (Hamdat et al., 2024). Specific challenges like organizational inertia or cybersecurity risks require focused decision-making (Ciacci et al., 2025). Meanwhile foundational IT decisions are critical for later workflows (Koseoglu, 2024). This relationship is cyclical: poor initial IT decisions can create later challenges forcing reactive, suboptimal decisions; proactive, well-informed decisions can lead to resilient systems and more stable growth.

## **2.3 Gaps in Existing Research and Future Directions**

While literature on IT systems and startup growth provides insights, several areas warrant further investigation to achieve a more comprehensive understanding.

### **2.3.1 Unexplored Facets of IT Challenges Across Startup Growth Trajectories**

Existing research often discusses IT challenges generally. There is a need for more nuanced investigation into how specific IT challenges (technical debt, integration complexity, and internal knowledge) manifest and evolve differently across distinct startup lifecycle stages (seed, early, growth, expansion) (Santisteban & Mauricio, 2017). The nature of a scalability challenge, for instance, differs significantly between an early-stage startup and one pursuing international

expansion. Current literature does not always sufficiently differentiate these stage-specific nuances (PremierNX; Santisteban & Mauricio, 2017)

### **2.3.2 Methodological Limitations and Opportunities for Future Inquiry**

Much existing research relies on cross-sectional studies or case methodologies (Ciacci et al., 2025; Santisteban & Mauricio, 2017). There is a noted scarcity of longitudinal studies tracking the co-evolution of IT systems, decision-making processes, and growth outcomes over extended periods within the same cohort of startups (Ciacci et al., 2025). Scaling is dynamic; cross-sectional snapshots may not fully illuminate evolving interdependencies. Longitudinal research is essential for richer insights into causal relationships and evolutionary patterns.

### **2.3.3 Integrating IT, Decision Dynamics, and Growth**

While literature addresses IT challenges and decision-making, these domains are often treated in relative isolation. There is a compelling need for research that explicitly integrates these areas to develop a holistic understanding of their interplay and collective impact on startup growth, resilience, and survival. Current frameworks often focus on specific aspects without comprehensively linking them to the full spectrum of startup IT challenges, decision nuances, and business outcomes (Drechsler & Weißschädel, 2017). The symbiotic relationship between IT challenges and decision-making permits more formal modeling and empirical validation. Future research could focus on developing and testing integrated theoretical frameworks to explain and predict how specific IT strategies and decision-making approaches help startups navigate challenges and achieve sustainable growth.

## **Chapter 3: Theoretical Framework**

### **3.1 Introduction: Framing IT System Replacement in High-Growth Environments**

The preceding literature review highlighted the significant IT system challenges faced by rapidly scaling startups and explored the complexities of IT decision-making dynamics within these contexts, including the role of data-driven approaches and formal IT governance structures.

However, it also noted that traditional IT governance frameworks (e.g., Weill & Ross, 2004), while providing crucial insights into decision rights and accountability, often presume a level of organizational stability frequently absent in the turbulent environments of fast-scaling firms. As studies on IT system change management indicate (e.g., Ivančić et al., 2019; Bondarouk et al., 2017), system replacement projects are rarely linear or purely rational, often encountering evolving requirements and shifting priorities not fully addressed by static governance models.

Consequently, to comprehensively analyze IT system replacement within a high-growth company like Veo Technologies, and specifically the "Fosbury Project," this thesis adopts a multi-faceted theoretical framework. This framework primarily draws upon *Organizational Change Theories*, with a specific emphasis on dialectical and teleological perspectives, to understand the macro-level dynamics of transformation. These are supplemented by theories of *Bounded Rationality* and the *Garbage Can Model* to illuminate the micro-level decision-making processes under conditions of pressure, ambiguity, and uncertainty that characterize such environments. This combined theoretical lens will facilitate an understanding of how IT system change unfolds, why formal IT governance structures may struggle or evolve, and how decisions are made amidst the complexities of rapid growth.

## 3.2 Primary Lens: Organizational Change Theories

To understand the overarching processes of transformation during the IT system, or more precisely the Subscription Management and Sales System replacement at Veo, this study utilizes the framework of organizational change "motors" as conceptualized by Van de Ven and Poole (1995). They identify four ideal types of change processes: life-cycle, teleological, dialectical, and evolutionary. While all offer insights, this thesis will particularly utilize the lens of dialectical and teleological theories to analyze the contested, high-stakes, and goal-oriented nature of the transformation observed.

### 3.2.1 Life-Cycle Models of Organizational Change

Life-cycle theory views organizational change as a programmed sequence of developmental stages, akin to biological growth (Quinn & Cameron, 1983). Greiner's (1972) model of "Evolution and Revolution" is a prominent example, identifying phases of growth (creativity, direction,

delegation, coordination, collaboration) each culminating in a crisis that necessitates structural transformation for continued growth. This model highlights how solutions to one phase's challenges become the source of the next crisis (Greiner, 1972, pp. 60–62), illustrating how organizational history shapes current options.

While useful for diagnosing internal "growing pains" in scaling firms, life-cycle theories often assume a predictable progression and stable governance, conditions rarely met in the volatile context of startups like Veo. Critics note their limitations in accounting for external shocks and political dynamics (Quinn & Cameron, 1983). The Fosbury Project, with its reactive and fragmented decision-making, suggests that a purely life-cycle perspective is insufficient, though Greiner's concept of crisis-driven growth remains relevant for contextualizing episodic transformations.

### **3.2.2 Evolutionary Theories of Change**

Evolutionary theories explain change through environmental selection, where organizational forms compete for resources, and successful variants are retained (Hannan & Freeman, 1977; Nelson & Winter, 1982). These models emphasize adaptation through variation, selection, and retention. However, they tend to underplay internal conflict and strategic agency within a single firm, making them less applicable for analyzing the specific internal dynamics of an IT project like Fosbury, where power dynamics and strategic choices are central.

### **3.2.3 Teleological Theories of Change**

Teleological theories conceptualize organizations as purposeful, goal-oriented systems that change through iterative cycles of objective setting, action, outcome evaluation, and behavioral adjustment (Van de Ven & Poole, 1995). This perspective, rooted in planned change (e.g., Lewin, 1951; Kotter, 1995), assumes change arises from intentional planning by a relatively unified actor. Teleological models are particularly useful for analyzing how startups adapt during instability. Leaders often engage in purposive change, formulating new objectives and implementing iterative learning cycles to stabilize direction (Leigh, 2016). Unlike life-cycle models, teleological theories accommodate the fluid, emergent nature of change in high-growth environments where

governance structures may evolve reactively. This makes them relevant for understanding dynamic decision-making in rapidly scaling organizations.

### **3.2.4 Dialectical Theory of Organizational Change**

Dialectical theories frame change as the outcome of conflict and struggle between opposing entities, values, or forces (Van de Ven & Poole, 1995). Rooted in Hegelian and Marxist thought, this perspective emphasizes contradiction and synthesis. Benson (1977) highlights principles of totality, contradiction, social construction, and praxis. Seo and Creed (2002) argue that institutional contradictions can spur collective action when actors perceive misalignments. Eaton et al., (2018) used this lens to model the emergence of e-ID infrastructures as a dialectical process of converging opposing interests. Dialectical theory is particularly suited for analyzing transformations involving contested ownership, shifting power, and internal opposition, conditions characteristic of the Fosbury Project and high-growth firms like Veo. It aligns with Greiner's crisis phases as points of tension. Unlike life-cycle or teleological models alone, it accounts for discontinuity and synthesis emerging from struggle, making it a core analytical lens for this thesis.

### **3.2.5 From Conflict to Direction: How the Fosbury Project Shifted between Dialectical to Teleological Change**

While this thesis primarily employs dialectical theory, the empirical case of Veo Technologies' Fosbury Project reveals a dynamic interplay between dialectical and teleological change processes, consistent with Van de Ven and Poole's (1995) assertion that organizational change rarely conforms to a single motor. Initial phases of the project appear dominated by dialectical tensions, as competing interests, unclear governance structures, and stakeholder misalignments generate conflict and organizational disequilibrium. However, as these contradictions escalate into crisis points, they catalyze purposive, goal-directed learning processes typical of teleological change. New objectives are formulated, ownership structures are realigned, and iterative adjustments are made to stabilize and scale internal IT systems. Thus, the Fosbury Project is hypothesized to exemplify a dialectic-to-teleological transition, where conflict serves as a generative force prompting adaptive goal-setting and organizational realignment over time. To visually represent

and guide the analysis of this phased, dialectical evolution within Veo Technologies, this thesis will employ the conceptual process model depicted in Figure 1

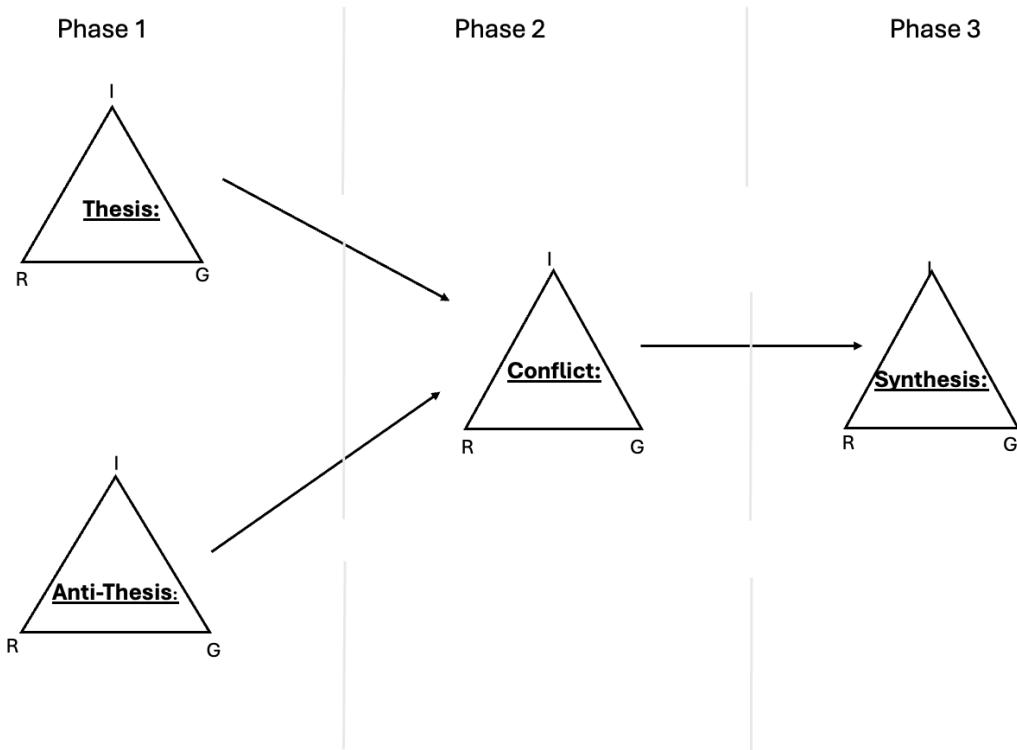


Figure 3: Dialectical process model

The phased progression nature directly corresponds to the metaphorical 'Child,' 'Teenager,' and 'Adult' stages used to frame the Fosbury Project's evolution, where the 'Child Stage' encompasses the initial Thesis and emerging Anti-Thesis, the 'Teenager Stage' represents the peak of Conflict, and the 'Adult Stage' embodies the movement towards Synthesis.

The Dialectical Process model (Figure 3) illustrates the hypothesized progression through three overarching phases. *Phase 1* is characterized by an initial **Thesis** (representing the dominant initial approach or conditions) and the emergence of an opposing or challenging **Anti-Thesis**. The key dimensions of Interests (I), Resources (R), and Governance (G) will be examined for both. The interplay and unresolved tensions between the Thesis and Anti-Thesis are expected to culminate in *Phase 2*, a period of heightened Conflict or crisis, where the I, R, and G dimensions are under significant strain. Finally, *Phase 3* represents the movement towards a **Synthesis**, a new,

reconfigured state or approach emerging from the conflict, which attempts to resolve prior contradictions and integrate learning. Again, the I, R, and G of this synthesized state will be analyzed. This model, inspired by dialectical frameworks such as that proposed by Eaton et al. (2018) for analyzing multi-stakeholder projects, will serve as an analytical template to structure the presentation and interpretation of the empirical findings from the Fosbury Project, mapping Veo's journey onto this dialectical progression.

This model, by operationalizing key dialectical concepts within a phased structure and allowing for the examination of teleological responses (particularly in the Synthesis phase), will serve as the central analytical template. It enables a structured interpretation of the empirical findings from the Fosbury Project, mapping Veo's complex journey of IT alignment and organizational learning onto this dynamic dialectical progression.

### **3.3 Supporting Lenses: Understanding Decision-Making Under Pressure**

To complement the organizational change perspective, this framework incorporates theories that illuminate the complexities of decision-making in the ambiguous and resource-constrained environments typical of scaling startups. These help explain *how* and *why* decisions regarding IT systems and governance are made in ways that may deviate from purely rational models.

#### **3.3.1 Bounded Rationality in High-Growth Firms**

Herbert Simon's (1991) theory of bounded rationality posits that decision-making is constrained by cognitive limitations, available information, and time. Within organizations, individual cognition is further shaped by role-specific expectations, embedded rules about valid information, and legitimate actions, limiting responsiveness under pressure (Simon, 1991). Organizational learning, while occurring in individual minds, is a social process; knowledge becomes organizational only when transmitted, retained, and accessible at relevant decision points. High personnel turnover, common in startups, disrupts tacit knowledge and erodes organizational memory, particularly the "chunks" of expertise distributed across individuals and their roles (Simon, 1991, p. 129). This makes sustaining coherence and innovation costly. Ingrained

organizational routines and decision structures can persist even after original actors depart. This theory will be used to analyze how cognitive and structural limitations influenced IT project decisions, knowledge management, and the ability to adhere to or adapt IT governance principles during the Fosbury project.

### **3.3.2 Ambiguous Decision Processes: The Garbage Can Model**

The Garbage Can Model (Cohen, March & Olsen, 1972) conceptualizes decision-making in "organized anarchies", settings with problematic preferences, unclear technology, and fluid participation, common in turbulent startups. Decisions emerge from the somewhat random intersection of four independent streams: choices, problems, solutions, and participants. As Cohen et al. (1972, p. 2) state, "[...] an organization is a collection of choices looking for problems [...] solutions looking for issues [...] and decision-makers looking for work." Outcomes are often determined by timing and energy rather than deliberate strategy, with some choices made by resolution, others by oversight or flight. Many problems may remain unresolved. Organizational design (access rules, decision rights) affects how problems are processed, but more hierarchical structures don't guarantee better resolution; important choices may still be made without full problem engagement (Cohen et al., 1972). This model is highly relevant for analyzing the often unpredictable and seemingly non-linear decision-making observed during the Fosbury Project, where ambiguity and limited managerial attention likely played significant roles. It helps explain how IT decisions might emerge contingently rather than through deliberate, rational strategies.

## **3.4 Integrating Perspectives: IT Governance in a Dynamic Context**

The discussion of IT governance and strategic frameworks in the literature review (Section 2.2.2), encompassing foundational theories (e.g., Weill & Ross, 2004; Peterson, 2003) and operational frameworks (e.g., COBIT, ITIL), provides benchmarks for understanding ideal structures for decision rights and accountability. This theoretical framework will use the lenses of organizational change, bounded rationality, and the garbage can model to analyze how formal IT governance principles were applied, adapted, or failed to stabilize during critical phases of the Fosbury Project. Specifically, these theories will help explain:

- How unclear or contested IT governance (a dialectical tension) created disequilibrium.

- How decision-makers, operating under bounded rationality, navigated these governance gaps.
- How the "garbage can" nature of decision-making in a fast-scaling environment might have led to IT choices that bypassed or reshaped formal governance structures.
- How Peterson's (2003) structural, process, and relational mechanisms manifested as formal adjustments or informal workarounds.
- How Greiner's (1972) model of crisis-driven growth contextualizes the episodic nature of governance evolution in response to tensions and realignments.

By combining these perspectives, this thesis aims to provide a rich, multi-layered understanding of the complex interplay between planned IT system change, emergent organizational dynamics, and decision-making realities in a high-growth technology firm.

### **3.5 Subconclusion**

This chapter has laid out the theoretical framework guiding this thesis. To understand how Veo Technologies navigated its IT system transformation during rapid growth, particularly through the Fosbury Project, the analysis will center on a Dialectical Process Model (Figure 1). This model synthesizes dialectical and teleological change theories to trace Veo's journey through its 'Child,' 'Teenager,' and 'Adult' phases, examining shifts in Interests (I), Resources (R), and Governance (G). The influence of Bounded Rationality and Garbage Can decision-making on this process will also be considered. Ultimately, this framework aims to provide a structured yet nuanced understanding of the complex interplay between planned change, emergent conflicts, and decision realities in Veo's pursuit of IT and growth alignment

## **4. Methodology**

The purpose of this methodology chapter is to outline and justify the specific research design and methods adopted to investigate the research question: *How does a tech startup balance its internal IT systems with its growth?* This chapter details the philosophical stance underpinning the study, the research strategy employed, and the concrete steps taken for data collection and analysis,

ensuring a transparent account of how the empirical insights into Veo Technologies' Fosbury Project were generated and interpreted.

The study is situated within the context of Veo Technologies, a Copenhagen-based sports tech startup that experienced rapid growth and organizational scaling. The empirical focus is the Fosbury Project, a strategic internal IT initiative aimed at replacing legacy systems and improving operational scalability. By examining this project as a single case study, the research seeks to uncover how internal system decisions unfold in high-growth startup environments and how tensions between growth imperatives and system governance are navigated over time.

## **4.1 Introduction to Case Company: Veo Technologies**

Veo Technologies (Veo), founded in Copenhagen in 2015, is a Software-as-a-Service (SaaS) company that emerged from the challenge of capturing and analyzing sports events. Its core offering combines exclusive AI-driven camera technology for autonomous recording with a cloud-based platform for users to analyze, distribute, and engage with sports footage (Veo Technologies, 2025). Veo utilizes a subscription model covering hardware and software, serving a diverse global range of customers from amateur clubs to professional organizations (Ruby, 2024; Murray et al., 2024).

### **4.1.1 Growth Trajectory and Strategic Imperative for IT Transformation**

By 2022, Veo had achieved significant international scale, having recorded over one million matches for more than 15,000 clubs across 80 countries. This rapid expansion was fueled by substantial venture capital funding, including a significant Series C round in June 2022 designed to accelerate growth and increase staffing to over 300 employees globally by 2024. This trajectory established Veo as a leading Danish tech enterprise, continually innovating its AI capabilities and expanding into new sports markets (Ruby, 2024; Murray et al., 2024).

This period of intense growth and scaling created a critical strategic imperative for Veo to overhaul and mature its internal IT systems. The existing infrastructure struggled to support the expanding operational complexity, diverse product offerings, and increasing transaction volumes. From a

methodological perspective, Veo therefore offers a compelling single-case study setting. As a high-growth, venture-backed SaaS company actively undergoing a strategic internal IT system transformation (the 'Fosbury Project'), Veo exemplifies the complexities, stakeholder tensions, and evolving governance challenges typical of scaling startups. These conditions make it a revelatory case (Yin, 2018) to explore how a tech startup attempts to balance its internal IT systems with its rapid growth under real-world pressures, with the Fosbury Project serving as the focal point for this analysis.



Figure 4: Fosbury project timeline

## 4.2 Research Philosophy and Approach

As displayed in Table 1 this study is grounded in an interpretivist research philosophy, which posits that social reality is constructed and understood through the meanings and interpretations of individuals (Bell et al., 2019). Such a philosophy is particularly suited for this research, as the core aim is to gain an in-depth understanding of the complex, lived experiences of Veo's stakeholders during the Fosbury Project, a process rich with subjective meanings, decision-making under ambiguity, and evolving interpretations. Ontologically, this aligns with constructivism, viewing the "reality" of the Fosbury Project as an emergent outcome of these social interactions. Epistemologically, knowledge is therefore pursued through an interpretive understanding of these participant perspectives (Saunders et al., 2016).

Consistent with this philosophical foundation, an inductive research approach was employed. Rather than testing pre-defined hypotheses, this approach allowed for patterns, themes, and a conceptual understanding of Veo's IT alignment journey to emerge directly from the rich qualitative data gathered. This inductive strategy provided the necessary flexibility to explore the nuances and complexities of the case, informing the development of the analytical framework, including the phased dialectical model presented in the Theoretical Framework (Chapter 3, Figure 1).

Method	Thesis Entailment	Methodology	Thesis Entailment
Sampling approach	Non-random sampling	Philosophy	Interpretivism
Type of sampling	Purposive sampling	Ontology	Constructivism
Criteria for interview Subjects	- Internal stakeholders at Veo Technologies involved in the Fosbury Project (e.g., CTO, engineers, product owners, former IT Director) - External consultants involved in the project	Epistemology	Interpretive Understanding
Interview questions	Semi-structured, open-ended questions	Primary Data	Mono-Qualitative
Interview	Semi-structured	Research approach	Inductive
Data Sources	- Primary: Semi-structured interviews - Secondary: Project documentation, post-mortems, timelines, system screenshots	Type of research	Exploratory Case Study
Analysis method (software)	Nvivo	Theoretical Lens(es)	Dialectical Theory, Teleological Theory
Analysis (Technique)	Thematic coding	Analysis method	Thematic Analysis; Narrative Construction
		Time period	Longitudinal
		Analysis (detail)	Inductive coding; Theme development; Phase-based analysis

**Table 1:** Methodology and Method

## 4.3 Research Strategy

The chosen research strategy for this study is a qualitative, single-case study focusing on Veo Technologies and its Fosbury Project. This strategy is highly appropriate for addressing "how" questions and for investigating complex, contemporary phenomena within their real-life context, especially when the boundaries between phenomenon and context are blurred (Yin, 2018; Bell et al., 2019). The Fosbury Project serves as a revelatory and longitudinal case, providing a unique opportunity to examine the intricate process of balancing internal IT systems with rapid organizational growth over time. The unit of analysis is this balancing process, explored through the lens of the Fosbury Project, encompassing strategic decisions, governance evolution, stakeholder interactions, and organizational learning. The temporal boundary for the primary case events analyzed spans from September 2022 to early 2025.

## 4.4 Methodological Choice

This study employs a qualitative mono-method approach, relying solely on qualitative data collection and analysis techniques. This aligns with the interpretivist philosophy and inductive approach, prioritizing depth, meaning, and context to understand the multifaceted nature of Veo's IT transformation journey.

The time horizon for this research is longitudinal, as the study examines the Fosbury Project and related organizational developments at Veo Technologies over an extended period, beginning in September 2022 and continuing into early 2025.

This approach is appropriate because the study seeks to understand a process of change and evolution, rather than a static moment (Saunders et al., 2016). The development of internal IT systems, stakeholder dynamics, and governance practices unfolded over time, captured analytically through the '*Child–Teenager–Adult*' phases, a conceptual lens developed as part of this thesis to interpret the case's progression. This approach enables the research to trace how

tensions emerged, shifted, and were addressed, which would not be possible through a cross-sectional snapshot. By observing the case across multiple stages, the study captures the temporal complexity of IT alignment in a high-growth startup context.

## 4.5 Data Collection and Analysis

This section details the practical steps taken to gather and analyze data, encompassing the sampling strategy, data collection methods, and the analytical process employed to answer the research question.

### 4.5.1 Sampling Strategy and Participant Selection

Consistent with the interpretivist aim of achieving deep, contextualized understanding, this study utilized a *purposive sampling strategy* (Bell et al., 2019). Participants were selected based on their direct involvement, knowledge, and varied perspectives concerning the Fosbury Project within Veo Technologies. My position as an employee at Veo, working under the IT Director during the project's timeframe, provided crucial insider knowledge. This facilitated the identification of key informants whose experiences would illuminate diverse facets of the case, ensuring a multi-angled understanding of the organizational dynamics and IT system changes under investigation.

The seven interviewees were chosen to represent a spectrum of roles, responsibilities, and stages of involvement, the seven semi-structured interviews were conducted with key stakeholders to gather primary data. The details of these interviews are summarized in Table 2 below:

Interview No.	Date	Participant Name/Pseudonym	Role / Affiliation	Mode	Duration (minutes)
1	March 19, 2025	Sam	System Integration Engineer	In-person	43
2	March 19, 2025	Ulver	Former IT Director	In-person	53
3	March 20, 2025	Øivind	System Engineer	In-person	33
4	March 27, 2025	Jesper	Zuora Product Owner	In-person	36
5	April 10, 2025	Alexander	CTO (Chief	Online	30

			Technology Officer		
6	April 30, 2025	Timo	Consultant (Subscription Factory)	Online	30
7	April 30, 2025	Øivind (2nd Interview)	Payment and Subscription Engineer	In-person	28

Table 2: Overview of Interviews

#### 4.5.2 Primary Data Collection: Semi-Structured Interviews

As reflected in Table 2 the primary data were collected through seven semi-structured interviews. This format was chosen for its flexibility, enabling the exploration of participants' lived experiences and allowing for probing and follow-up (Bell et al., 2019, p. 439–441). Five interviews were conducted in-person, and two were conducted online to accommodate participant availability. Durations ranged from approximately 30 to 60 minutes. Semi-structured interviews were selected for their capacity to elicit rich, detailed narratives while maintaining the flexibility to explore emergent themes, a vital characteristic for an inductive inquiry. This format enabled in-depth probing of underlying assumptions, reflection on contradictions, and clarification of context-specific meanings.

Several interview guides were created informed by the literature and project context, but it was used as a flexible roadmap rather than a rigid script (Bell et al., 2019, p. 439). This allowed participants to steer the conversation toward what they deemed significant. One participant (Timo, an external consultant) requested to receive the questions in advance, which was accommodated.

All interviews were conducted in private settings to ensure confidentiality and participant comfort. The two online interviews were conducted via Google Meet, while the remaining five took place in person. Informed consent was obtained from all participants prior to the interviews. Six out of seven interviews were audio-recorded and automatically transcribed using Google Meet's transcription function. The seventh interview, conducted in a setting where digital tools were not feasible, was recorded with a traditional device and transcribed manually. As Bell et al. (2019, p. 445) emphasize, transcription is essential not only for preserving accuracy, but also for enabling rigorous analysis, transparency, and potential data re-use. While transcription can be time-

consuming, it proved vital for engaging deeply and reflexively with participants' narratives during the analytical process.

#### **4.5.3 Secondary Data and Triangulation**

To complement the interviews, a range of secondary sources were incorporated, including: Internal project documentation, Post-mortem summaries from internal reflections, Screenshots of system architecture and tool transitions, A timeline of Fosbury phases developed from shared internal sources.

These materials were accessed through Veo's internal knowledge base and supplemented with artifacts shared by participants. As Bell et al. (2019) note, triangulating qualitative data enhances credibility by enabling cross-validation of events and perceptions. In this case, documents helped corroborate or contrast interview narratives and offered concrete evidence of shifts in system governance, team roles, and infrastructure configurations.

#### **4.5.4 Thematic Data Analysis**

The data analysis followed a *thematic analysis process*, as outlined by Braun and Clarke (2006), aligning with the study's interpretivist philosophy and inductive orientation. The objective was to systematically identify, analyze, and report patterns of meaning (themes) across the dataset, focusing on participants' lived experiences and interpretations of the Fosbury Project.

**Transcription and Familiarization:** Audio-recorded interviews were transcribed verbatim. All transcripts and secondary data were repeatedly reviewed for familiarization. These materials were imported into *NVivo qualitative data analysis software* to manage and facilitate systematic coding and theme development.

The coding of interview transcripts and secondary documents was an iterative process employing both deductive and inductive strategies. *Deductive codes* were guided by the study's theoretical framework, initially focusing on concepts such as "dialectical tension," "teleological adaptation," "bounded rationality," and "IT governance mechanisms" e.g., Van de Ven & Poole, 1995; Simon, 1991; Weill & Ross, 2004. Simultaneously, *inductive codes* emerged directly from the data, capturing specific participant terminology and nuanced experiences like "governance vacuum,"

"handover chaos," the "Fosbury Flop metaphor," and "platform pain points" [cite key examples from your analysis or appendix if relevant]. All data segments were systematically coded, involving constant comparison across the dataset to identify patterns, similarities, and differences. This was followed by an iterative refinement process where codes were grouped and synthesized into higher-order categories, forming the basis for theme development.

**Theme Development:** This iterative process of coding and categorization, managed within NVivo's node hierarchy and analytical tools, culminated in the identification of *seven main themes* that captured the significant patterns of meaning regarding Veo Technologies' navigation of IT system changes during rapid growth. These overarching themes, which broadly address aspects such as dialectical tensions, governance structures, implementation challenges, organizational learning, and system-specific issues, are presented in detail with their constituent codes/sub-themes in Table 3.

Main Theme	Associated Codes / Sub-Themes
1. Dialectic & Tensions	Bad Communication, Dialectic Tensions, Political Dynamics, Underestimations
2. External Factors	External Consultants
3. Governance & Decision Making	Change Management, Decision Making, Governance Errors, High-Speed Tradeoffs, Lack of Cross-Functional Governance, Lack of Prioritization, Ownership Ambiguity, Stakeholder Involvement, Strategic Disagreement
4. Implementation Assumptions	Plug and Play, Unrealistic Business Requirements
5. Learning & Capability	Builders Absorption, Knowledge Transfer, Lack of Internal Skill, Learnings, Trial and Error Learning
6. Organizational Maturity & Culture	Maturity, Teenager Analogy
7. Systems & Infrastructure	(Lack of) Organizational Overview, Current System Setup, Frustrations, Legacy System Setup, Scalability, Subsavers, Systems

Table 3: Overview of Main Themes and Associated Codes/Sub-Themes from NVivo Analysis

Following the iterative thematic coding process, the analysis employed a hierarchical approach to structure the emergent data, drawing principles from methodologies for developing grounded data structures (e.g., Gioia, Corley, & Hamilton, 2013). This involved moving from granular codes ('First Level Categories') to broader themes ('Second Level Themes'), which were then

consolidated into four overarching 'Aggregated Dimensions'. These dimensions represent the core analytical concepts derived from the data: Strategic & Governance Frameworks, Operational Processes & Infrastructure, Organizational Capacity & Development, and External & Relational Dynamics. The complete hierarchical structure, detailing the progression from First Level Categories through Second Level Themes to these Aggregated Dimensions, is visually presented in Appendix 2, Figures 6-9.

The coding process, the analytical concepts, and the thematic development will be informed by the core constructs of the theoretical framework, particularly the stages and components outlined in the *Dialectical Process model 1* (Figure 1 in Chapter 3).

These thematically grouped insights form the structural foundation of the thesis' analytical narrative. They guide both the empirical exposition and theoretical interpretation, particularly within the dialectical-teleological process model and the metaphorical 'Child–Teenager–Adult' framing developed to explain Veo's organizational transformation

#### **4.5.5 Interpretation and Narrative Construction**

The final analysis phase involved interpreting these developed themes through the lens of the established theoretical framework, primarily drawing upon Van de Ven & Poole's (1995) dialectical and teleological motors of change, supplemented by concepts from bounded rationality and IT governance literature. The goal was not merely to list themes but to understand their interplay and how they collectively explained the process of Veo balancing its IT systems with growth.

To structure and communicate the findings effectively, two key narrative devices emerged from the data analysis: The 'Child–Teenager–Adult' metaphor was developed to represent the distinct maturity phases observed in the Fosbury Project and Veo's broader internal IT evolution. A visual process model was created to synthesize the dialectical and teleological dynamics influencing IT system evolution within the high-growth startup context, illustrating how opposing forces shaped outcomes. This interpretive process allowed for the construction of a rich, theoretically informed narrative explaining the complexities of the case study.

#### **4.5.6 Use of Generative Artificial Intelligence (GenAI)**

In the course of developing this Master's thesis, Generative Artificial Intelligence (GenAI), specifically Google's Gemini model and OpenAI's ChatGPT model, was employed as a supplementary tool to assist the researcher. This usage was in accordance with the guidelines provided by Copenhagen Business School (CBS) for Master's theses, which permit GenAI for specific, declared purposes (Copenhagen Business School, libguides.cbs.dk, 2025).

GenAI was used as a language assistant to refine sentence structure, improve clarity, and suggest alternative phrasing. It also served as a brainstorming partner to help structure arguments, outline sections, and articulate complex ideas based on the researcher's drafts. Additionally, GenAI supported the development of descriptive text for conceptual models, refinement of interpretation themes, and in some cases, assisted in drafting or rephrasing segments of text using detailed prompts and researcher-provided input.

### **4.6 Ethical Considerations**

This study adhered to established ethical guidelines as outlined by Bell et al. (2019) and Saunders et al. (2016), ensuring that all research activities upheld the principles of integrity, respect, and non-maleficence. Prior to each interview, participants received a clear explanation of the study's purpose, the voluntary nature of participation, and how their data would be used. Informed consent was obtained from all participants, either in writing or verbally (recorded), and participants were reminded of their right to withdraw at any stage without consequence.

Confidentiality and anonymity were treated as core ethical obligations. Interview recordings and transcripts were stored securely on password-protected devices, and any identifying information was removed or anonymized in the final reporting. Participants were given the option to be anonymous and only be referred to only by role, which was chosen by some of the participants.

Transparency was emphasized throughout: participants were made aware that the study was conducted as part of a master's thesis and that findings would be shared in an academic context. To minimize potential harm, no evaluative judgments were made, and questions were designed to allow reflection without pressuring participants to disclose uncomfortable information.

A degree of *researcher reflexivity* was also embedded in the process. As a participant-observer involved in the Fosbury project since January 2024, the researcher recognized his dual role as both insider and researcher. This position afforded valuable contextual insight but also required conscious effort to critically reflect on assumptions and potential biases during analysis. Regular peer debriefing and systematic coding in NVivo helped mitigate this influence and foster analytical transparency.

## 4.7 Methodological Limitations

Despite its rigor, this study is subject to several methodological limitations that must be acknowledged.

As described by Bell et al. (2019) *Generalizability* is inherently constrained due to the single-case study design. The findings are not statistically generalizable to other startups or IT transformations but are instead intended to support *analytical generalization*, offering theoretically informed insights (e.g., dialectical tensions in IT change) that can inform broader discussions in organizational research.

*Retrospective bias* is another consideration. Several interviews involved recollection of past events (e.g., early 2023 planning decisions), which may be affected by hindsight, fading memory, or self-justification. While the inclusion of real-time data collection during my ongoing tenure helped address this, full objectivity cannot be guaranteed.

*Researcher bias* may also have shaped data interpretation, particularly due to my insider role. Although this positionality allowed access to nuanced dynamics and informal knowledge, it also necessitated careful reflexive practice to avoid over-identification with participants or the organization.

Lastly, the study lacks *quantitative system performance data*, such as metrics on project efficiency or IT system uptime, which could have provided triangulation from a technical performance standpoint. Instead, insights into success, scalability, and breakdowns were derived from qualitative accounts and documentary analysis.

## 5. Findings

This chapter presents the empirical findings from the case study of Veo Technologies, addressing the research question: *How does a tech startup balance its internal IT systems with its growth?* The analysis reveals that this balancing act is not a linear, predictable process but rather a dynamic and often tumultuous journey of organizational change and learning. To illuminate this evolution, the Fosbury Project, Veo's ambitious attempt to overhaul its commercial IT infrastructure, will be traced through three distinct phases. These phases can be metaphorically understood through the lens of human development: a 'Child Stage,' characterized by initial ambitions and inherent immaturities; a 'Teenager Stage,' marked by growing pains, internal conflicts, and the eruption of systemic contradictions; and an emerging 'Adult Stage,' representing a move towards greater stability, learned maturity, and more solid systems.

The 'Child Stage' at Veo reflects its early operational simplicity, where the primary business goal was direct sales, and financial survival often dictated 'fast and simple' IT solutions. As the Fosbury Project was conceived within this context, it inherited both the optimism and the underestimation of complexity characteristic of this phase.

Subsequently, the 'Teenager Stage' captures the period where the initial flaws and unaddressed complexities of the Fosbury Project began to surface dramatically. Much like a teenager undergoing puberty, Veo experienced significant 'growing pains': internal systems became problematic, unexpected issues arose, and internal conflicts intensified as the organization grappled with the demands of scaling its IT.

Finally, the 'Adult Stage' explores Veo's efforts to navigate out of this crisis, reflecting a move towards greater maturity. This involved significant restructuring, a re-evaluation of technical strategies (such as the Zuora 2.0 initiative), and the development of systems and processes more fitting for a larger, more complex organization.

Throughout these phases, the analysis will apply a dialectical process lens, interpreting the evolution as a series of tensions, breakdowns, and reconfigurations, while also acknowledging the teleological (goal-directed) efforts that shaped Veo's journey

## **5.1 Phase 1 - Strategic Framing (Child Stage)**

The Fosbury Project was initiated with the clear *teleological* purpose of replacing Veo's limited legacy IT commercial systems to achieve critical business goals. This included making the commercial team and salespeople's lives easier, gaining usable data on customers, achieving a clearer data structure, and obtaining better subscription knowledge, as this was not achieved in the company at the time. More broadly, this translated into the strategic objectives of establishing a "customer 360" view, enabling scalable subscription management, and ensuring overall easier commercial operations to support Veo's rapid growth. This intended, rational path forward was embodied by the Business IT team taking ownership of the project and the decision to engage external consultants to implement chosen solutions, namely Zuora and Salesforce, which were seen as the tools to achieve these desired ends. The origins of what would become known in Veo as the *Fosbury Project* trace back to September 2022 (Figure 4). The initiative drew its name from the Olympic high jumper Dick Fosbury, who revolutionized his sport by inventing the now-standard "Fosbury Flop." The metaphor resonated strongly with the Business IT team at Veo Technologies, who faced an unprecedented implementation challenge. The "Fosbury Flop" metaphor itself symbolized the teleological ambition for a revolutionary, goal-achieving leap forward.

### **5.1.1 Organizational Structure at the Time of Implementation**

During the main implementation phases of the Fosbury Project (late 2023), Veo Technologies operated under a fragmented and evolving organizational structure (see Figure 5). The company's executive leadership consisted of a CEO supported by a Chief Financial Officer (CFO), Chief Technology Officer (CTO), Chief Revenue Officer (CRO), and a newly hired Chief Operating Officer (COO).

Commercial activities such as Sales, Brand, and Marketing were overseen by the CRO, while Finance, Accounting, Legal, and Business Intelligence reported to the CFO. The CTO, Alexander, oversaw the Builders and Capture departments, both primarily focusing on core product development, engineering and creating the software platform where videos can be accessed by the

customer. Builders, located under the CTO, in particular, concentrated exclusively on internal software products and AI-driven features, with no direct involvement in business-facing systems such as subscriptions, CRM, or e-commerce. The responsibility for commercial systems, including subscription management (Zuora), customer relations (Salesforce), logistics integration, and distribution, was placed under the Operations department consisting primarily of IT, Business-IT, Logistics and Production teams, reporting to the COO Ditte.

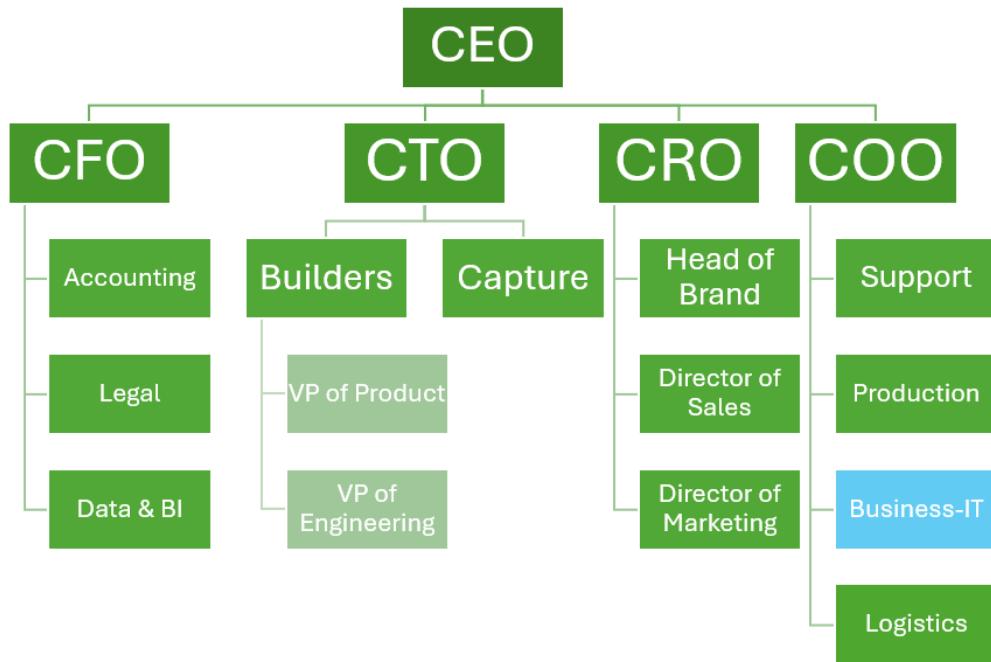
The separation of product and commercial IT was a deliberate organizational design choice, as prior experience from Ulver and Jacob Frost, the Former VP of Engineering from their previous set-up at Phase One, where a clear interface existed between product innovation and commercial operations (Interview 2, Appendix 4, Ulver). Something that was later criticized by the CTO: "[...] I've never seen something like this Business IT department before [...] scaling up and down is basically impossible" (Interview 5, Appendix 4, Alexander). However, while the functional split between Builders and Business-IT was conceptually clear, the lack of formal cross-functional governance structures meant that coordination across these domains was ad hoc, fragile, and often politically contested.

This organizational configuration created multiple points of structural tension: Business-IT lacked direct escalation paths to strategic decision-makers, Builders focused on internal product evolution with limited attention to commercial systems, and decision making and operational ownership over core subscription flows remained ambiguous (Appendix 4, Alexander). As the Fosbury implementation accelerated, these structural fractures increasingly manifested in misaligned decision-making, delayed conflict resolution, and growing organizational frustration.

### **5.1.2 The Pre-Fosbury Setup**

Veo Technologies has been experiencing rapid growth for many years, and the former sales and subscription systems had been designed for a relatively simple business model. That model was typically selling a single camera per customer, with limited instances of repeat purchases (Appendix 4, Ulver). However, this model turned lack luster as customers returned for additional cameras and overall usage across clubs expanded. This created significant operational blind spots. As the former IT Director Ulver explained, "The major issue when I started was that everyone was

asking for a proper customer 360.” (Interview 2, Appendix 4, Ulver). There was no consolidated view of the customer, no visibility into what had been purchased, and no ability to connect subscriptions to actual club accounts.



The legacy systems environment at Veo was a fragmented patchwork of systems designed to address immediate business needs, but lacking an overarching architectural or strategic coherence. The former systems engineer working in Business IT described the early technology stack as including ”Shopify, PipeDrive[...] [for] placing sales orders,” alongside CRM tools used for lead tracking and customer outreach (Interview 1, Appendix 4, Sam). Behind this frontend, the Bold platform managed subscription services, yet it was ”not scalable enough” and increasingly inadequate as Veo’s customer base diversified and expanded.

The underlying system design contributed significantly to this fragmentation. Veo operated on a hybrid setup combining Shopify with a homegrown subscription system called Zola. These systems were neither integrated nor designed for complex customer management. As Ulver recalled: ”There were just a bunch of Zola logins, how many of them had a subscription, how many belonged to the same club, etc., no one could answer that” (Interview 2, Appendix 4, Ulver). Data

was siloed, inconsistently structured, and difficult to interpret at scale, effectively blocking any attempt at developing a coherent customer 360 view.

From the outset, the ambition was to reconfigure Veo's architecture to better support commercial sales, organizational scalability and data clarity. Data was siloed, inconsistently structured, and nearly impossible to interpret in a scalable way. A sudden need to soothe the growing pains The Fosbury project was thus initiated as a strategic response. Ulver described the initial vision as, "The idea [...] was [...] to create a setup where Product would handle the product, and IT would manage the commercial part of the solution." (Interview 2, Appendix 4, Ulver). As mentioned, this division of labor echoed the prior experience of the former Vice President of engineering and IT Director Ulver, who had worked together at Phase One, where a clear separation between product development and customer-facing systems had supported scalability.

The decision to move away from the inherited system patchwork and initiate the Fosbury project appears to have been triggered by a sudden, pressing realization of data inadequacy. As stated by Ulver: "[...]it quickly became clear that all this data needed to be sorted out, so we could get an overview of what we have sold to our customers." (Interview 2, Appendix 4, Ulver). This reactive push, born out of an urgent need to address systemic failings rather than from proactive, long-term strategic IT planning, aligns with elements of the Garbage Can Model (Cohen, March & Olsen, 1972). Where a significant problem (unusable customer data) found a choice opportunity (the Fosbury project), to which solutions (new systems) and participants (the project team, consultants) became attached. Thus, the Fosbury initiative itself emerged not from a linear, planned evolution, but as an urgent organizational response to the realization that its existing IT infrastructure was fundamentally unviable for its current scale and future ambitions.

### **5.1.3 Choosing The New Systems**

"When selecting the new IT systems to support Veo's scaling ambitions, one might expect a rigorous, forward-looking strategy grounded in long-term scalability and architectural coherence. However, the empirical evidence suggests a more complex, multi-faceted decision process. The selection of the subscription management platform involved considerable discussion and evaluation. As Ulver recalled, there were 'many talks about, what should the solution be?' leading

the team to explore 'different subscription solutions,' including Zuora, Chargebee, and at least one other, with presentations comparing these products (Interview 2, Appendix 4, Ulver). While 'many of us had opinions and views,' the consensus ultimately formed around Zuora as 'probably the safest choice' for Veo's long-term scalability, given its status as a major enterprise-grade platform (Interview 2, Appendix 4, Ulver).

In contrast, the choice of CRM system appears to have been more straightforwardly influenced by key personnel experience. The project manager, Christian Houen, 'comes from a Salesforce world. So Salesforce was sort of automatically solved because of that' (Interview 2, Appendix 4, Ulver). This suggests that while the subscription platform underwent a comparative review, the CRM selection leaned heavily on familiarity.

Thus, while not necessarily a fully formalized, textbook evaluation across all components, the overall approach was shaped by a blend of deliberate comparison (for Zuora) and path dependency or familiarity (for Salesforce). This was further reinforced by the involvement of Subscription Factory, an external consultancy whose prior experience implementing Zuora and Salesforce integrations, and their alignment with the consultancy's existing playbooks, likely cemented the decision to proceed with this particular pairing and influenced the subsequent implementation strategy (Interview 2, Appendix 4, Sam). This blended decision-making, balancing evaluation with expediency and stakeholder experience, reflects characteristics of bounded rationality, where choices are made under constraints of time, available information, and existing expertise (Simon, 1991).

Despite the comparative review undertaken for the subscription platform, the prevailing sentiment among some team members was that the overarching *strategic intent* and the planned *method of integrating* these new systems still leaned heavily on existing paradigms rather than a fundamental operational rethink. Øivind, the main systems engineer in the Business IT team, noted that "[...] management thought they could copy-paste our current setup with new systems so that they could scale. They kind of wanted Shopify - just better" (Interview 3, Appendix 4, Øivind). This suggests that instead of fundamentally rethinking the system architecture, the decision-makers relied on familiar paradigms, applying existing models to new challenges with limited critical evaluation. Similarly, Sam observed that "there was some research done [...] but most of it was decided by

people up in the chain,” adding that selection was often informed by informal conversations rather than a comprehensive governance process (Interview 1, Appendix 4, Sam).

Another clear example of bounded rationality is how system decisions were made, not just which systems to adopt, but more importantly, how they were implemented, configured, and integrated. These decisions were made under pressure: limited time, incomplete information, organizational urgency, and a narrow mindset shaped by previous experience (such as the “copy-paste” logic). Instead of thorough evaluation, the team relied heavily on familiar patterns and assumptions, which later contributed to the instability that unfolded in the next phases of the project (Simon, 1991).

#### **5.1.4 Utilizing External Consultants**

With the decisions made on Zuora and Salesforce, Veo faced the challenge of implementation. Lacking deep in-house expertise in these enterprise-grade systems and pressed for time, the company chose to outsource this critical phase. They engaged Subscription Factory, a consultancy with recognized expertise in Salesforce and Zuora integrations and an official Zuora partner (Appendix 4). The IT Director, Ulver, explained the pragmatic reasoning: “We lacked technical knowledge of Zuora in the phase where the solution was being built... So we placed our trust in Subscription Factory and didn’t have the resources to validate their work” (Interview 2, Appendix 4, Sam).

Subscription Factory was thus tasked with providing an implementation model designed for speed and scalability. While this addressed Veo’s immediate resource and knowledge gaps, the consultancy’s extensive involvement profoundly shaped the project’s technical trajectory, dictating platform configurations, business logic, and integration assumptions. This approach, however, came with significant trade-offs, particularly concerning internal ownership and the system’s ultimate flexibility.

Something which the consultancy firm also criticized was the scope of the project; specifically, that it attempted to solve too many things at once without a phased or prioritized implementation strategy. As Timo from Subscription Factory recalled, “It was a big project. It was not only about recurring billing [...] it was everything and everything and everything together” (Interview 6, Appendix 4, Timo).

Rather than focusing on setting up core billing infrastructure first and layering complexity over time, Veo initiated a full-scale overhaul across subscriptions, pricing logic, customer touchpoints, Salesforce CPQ, and fulfillment operations simultaneously. This "everything-at-once" logic not only introduced significant integration complexity but also magnified the organization's internal coordination challenges. From a change management perspective, the lack of scoping discipline made it difficult to sequence decisions, assign clear responsibilities, and manage stakeholder expectations across teams, especially under conditions of limited capacity and unstable governance. The attempt to implement such a broad solution space in a single wave contradicts teleological principles of iterative learning and instead led to overloaded implementation efforts that outpaced the organization's absorptive capacity.

In hindsight, several stakeholders reflected that the reliance on Subscription Factory - and the lack of internal counterparts to challenge or translate between business and technical needs, contributed to later implementation issues. As Ulver put it, "they were never going to say no, (Interview 2, Appendix 4, Sam)" highlighting the risk of vendor lock-in and misalignment when external actors are left to define both the problem and the solution.

Even Subscription Factory acknowledged that the consultancy team may have enabled complexity instead of pushing back more strongly. Timo Zuideest reflected that:

*We were more or less following Veo too much in their way of thinking. I think we had to make more noise about: guys, keep it simple. I think we all made it a bit too overcomplicated in some areas. That's my lesson learned as well for this (Interview 6, Appendix 4, Timo).*

The reliance on external consultants was not just a response to short-term capacity constraints, but also reflected a broader belief within the organization that technical specialization could be temporarily outsourced without long-term consequences. As Ulver reflected, "Thomas Hagen was... an advocate for using consultants for these kinds of things. And it's quite normal to bring in consultants... because you typically don't want in-house specialists for that kind of work" (Interview 2, Appendix 4, Ulver). The choice to externalize such a foundational component of Veo's digital infrastructure reflected a classic bounded rationality trade-off (Simon, 1991): a

rational response under resource and capability constraints, but one that limited Veo's ability to adapt and evolve the system architecture organically. This also set the stage for later dialectical tensions between external consultants, internal stakeholders, and shifting governance structures.

### **5.1.5 Fosbury System Design vs. Zuora Best Practice**

The expectation from, particularly senior leadership, was that implementing Zuora and Salesforce would be a relatively straightforward, 'plug and play' exercise. Rather than adjusting business processes to fit the new systems, the dominant logic was to preserve existing business logic from the legacy stack and layer it over the new tools. This led to a forced translation of old workflows into new platforms that were not designed to support them natively as the expectation was Shopify - just better (Appendix 4).

This 'copy-paste' mentality meant that instead of using the built-in strengths of Zuora and Salesforce, the project focused on recreating complicated old processes within the new platforms. The aim was to replicate past behaviors rather than adapt operations to the new systems' scalable design. This approach fundamentally mismatched the new platforms with their intended uses, making the systems unstable and creating long-term maintenance headaches. What could have been a key moment for strategic improvement became an exercise in simply copying the old architecture. As a result, Veo inherited the old system's inefficiencies without achieving the desired scalability from the new investment.

Zuora is built to support scalable, out-of-the-box subscription billing as Jesper, Veo's Zuora Product Owner, explained:

*From my experience, how Zuora is usually worked is to use as much out-of-the-box feature as possible. You place an order, you invoice for that order, you collect payment, you send all the transactional data, and you do revenue recognition, in Zuora or in an ERP system.*  
(Interview 4, Appendix 4, Jesper)

Veо's implementation, however, deviated from this standard architecture. The original intent behind adopting Zuora was to streamline billing, invoicing, and subscription logic in a standardized manner. Veо's implementation strategy failed to follow this simplified logic. At the time of Jesper's arrival, the platform was already live, and the system architecture had been developed primarily by external consultants, with minimal in-house ownership or oversight. As he recalled: "I saw the solution and I thought it was probably not the worst, but the most complex solution I've ever seen."(Interview 4, Appendix 4, Jesper).

Normally, complex business rules are kept separate from Zuora and managed by other connected tools. At Veо, however, many of these important rules were built directly *inside* the Zuora platform. This included detailed workflows and custom decision-making steps, often using Zuora's own specialized programming language (Liquid), which most of Veо's developers weren't familiar with. Jesper, a key team member, confirmed this unusual setup: 'Usually, businesses handle logic outside Zuora. But when I got here, all of that logic lived inside. It was clear that the consultants were just following requirements, and built it all in' (Interview 4, Appendix 4, Jesper)

To illustrate the divergence between Zuora's intended use and Veо's implementation logic,Table 4 presents a simplified step-by-step comparison:

Step	Zuora Best Practice	Fosbury Implementation
1. Order Placement	Standard order-to-subscription process	Custom CPQ logic with manual setup
2. Invoice Generation	Auto-generated upon order confirmation	Often delayed; tied to fulfillment logic
3. Payment Collection	After Invoice	Before invoice (upfront logic)
4. Data Sync	Transactional data pushed to ERP	Transactional data pushed to ERP
5. Revenue Recognition	Revenue recognized in ERP	Revenue recognized in ERP

Table 4: Zuora's Best Practice, vs. Veо's use

### 5.1.6 The Importance of a Steering Committee

According to Peterson (2003), effective IT governance requires the integration of structural, processual, and relational mechanisms. In the Fosbury project's initial stages, however, the development of these mechanisms appears to have been a significant challenge, leading to role confusion, inconsistent information flows, and delayed coordination between key units. Despite the project's strategic ambitions, the early governance arrangements were perceived by some as notably underdeveloped.

For instance, the external consultant Timo from Subscription Factory recalled that during the critical early phases of design and initial implementation, there was a perceived lack of robust oversight: "... there was not really a steering committee... [and] there was a lack of internal collaboration on the project" (Interview 6, Appendix 4, Timo). This sentiment regarding the *effectiveness* or *visibility* of a guiding coalition was echoed by observations from internal team members about the exclusion of technical staff from crucial early conversations. Ulver, the IT Director, noted, "we lacked technical knowledge of Zuora in the phase where the solution was being built" (Interview 2, Appendix 4, Sam), and Sam, a systems engineer, observed that "we did not have the right people in the right room to actually make decisions" (Interview 1, Appendix 4, Sam). Øivind further confirmed that "no technical people were in the room when they talked to the consultants," (Interview 3, Appendix 4, Øivind) meaning important integration assumptions were made without validation from those who would later implement or maintain the system.

However, a formal "Fosbury steering committee" did exist, at least by the time Alexander Grosse joined as CTO. Grosse recounted being invited to it "a few weeks after I joined," indicating its presence (Interview 5, Appendix 4, Alexander). Yet, his initial assessment of this committee painted a picture of dysfunction rather than effective governance: "I felt from the beginning that this was not on a good path, there was this steering committee [trying to] high five [and] we will solve everything[...] and Fosbury will solve everything" (Interview 5, Appendix 4, Grosse). He added, "I had a bad feeling to be frank but I was not deep enough to undermine it with a lot of facts. Right? This is just based on former experience."

Thus, while a steering committee was formally instituted at some point, the collective evidence suggests it may have been established later than ideal, or more critically, that it failed to function as an effective, integrated governance body in the project's formative stages. The absence of

technical insight at strategic discussions, the feeling of decisions being "handed down" without context as described by Sam ("Most of it was decided by people up in the chain [...] we were just handed decisions without proper context"), and Grosse's perception of a disconnected and overly optimistic steering group all point to significant deficiencies in Peterson's (2003) processual and relational governance mechanisms. This environment, characterized by a disconnect between high-level pronouncements and on-the-ground realities, and a lack of inclusive, technically informed decision-making, fostered conditions akin to the "organizational anarchy" described by the Garbage Can Model (Cohen, March & Olsen, 1972), where problems, solutions, and actors struggled to align effectively.

### **5.1.7 A Black Box: Lack of Knowledge Transfer and Training**

While the choice to rely on Subscription Factory was driven by lack of internal knowledge, the nature of the collaboration left the internal technical stakeholders, Øivind and Sam largely in the dark during critical stages of implementation. The consultants worked with minimal involvement from Veo's own technical staff, which resulted in a knowledge vacuum once the systems were handed over. As Øivind noted, "I wasn't introduced to the consultants until way into the project" (Interview 3, Appendix 4, Øivind). Key design and configuration decisions were made without input from the very engineers who would later be responsible for operating and evolving the platforms.

This dynamic turned the implementation into a *black box* - as the process was shaped externally, shielded from internal scrutiny, and lacking mechanisms for structured knowledge transfer. As Sam explained, "There were no trainings [...] we would continue to learn about these systems while having it all in production" (Interview 1, Appendix 4, Sam).

Øivind echoed this, describing his introduction to the system as entirely reactive:

No, I learned as I went [...] that was the biggest learning I had, often we went live in production so that I could sit in Zuora and see how everything worked. Every request I got I just had to do trial and error to figure things out (Interview 3, Appendix 4, Øivind).

Rather than entering the operational phase with confidence and clarity, the team was forced into post-hoc learning under pressure, without guidance or scaffolding.

The absence of systematic training or documentation reflects a broader organizational underestimation of the expertise required to operate enterprise-grade platforms like Zuora and Salesforce. Ulver candidly admitted that "we underestimated the complexity of the setup, and just assumed that Sam and a few others would be able to figure it out"(Interview 2, Appendix 4, Sam).

Phase 1 of the Fosbury Project, its "Child Stage," launched with high hopes—a "Fosbury Flop" intended to resolve pressing data issues and support Veo's rapid growth. However, this initial phase vividly revealed an organizational immaturity ill-suited for such a complex transformation.

### **5.1.8 Sub-Conclusion: Phase 1**

Phase 1 – The Child Stage of the Fosbury Project began with high hopes. Inspired by the metaphor of the "Fosbury Flop," the project aimed to fix urgent data issues and support Veo's rapid growth. But from the start, it became clear that the organization wasn't ready for such a complex transformation. The project was launched reactively, triggered by a data crisis, and took place within a fragmented structure that separated technical and business domains.

Early decisions reflected this immaturity. While the choice of systems involved some evaluation, it was mostly based on familiarity. More critically, the implementation approach was flawed. A "copy-paste" mindset led the team to rebuild old processes inside new tools like Zuora, ignoring standard features and creating unnecessary complexity. Governance was weak: there was no proper steering group, and technical experts were often left out of strategic decisions. At the same time, the company relied heavily on external consultants for a large-scale rollout, without proper knowledge transfer. This left the internal team in the dark, unable to fully understand or manage what had been built.

Phase 1 was defined by a reactive launch, unrealistic assumptions, poor governance, and a failure to build internal capability. These early missteps laid the groundwork for the tensions and breakdowns that followed in the next phase.

## **5.2 Phase 2 - Emergence of Contradictions (Teenager Stage)**

Following the 'Child Stage' of strategic framing, the Fosbury Project entered its complicated 'Teenager Stage.' This phase represents a critical moment where the **initial Thesis**; the project as conceived and launched in Phase 1, with its embedded flaws, optimistic assumptions (like a 'plug and play' solution), and underdeveloped structures, collided forcefully with the complex realities of implementation. The 'growing pains' became serious as a powerful **Antithesis** emerged, characterized by the eruption of previously lurking problems and the introduction of new, challenging forces and perspectives.

Much like a teenager experiencing an identity crisis and rebelling against established norms, Veo during this period witnessed a dramatic escalation of internal contradictions. The idealistic assumptions of the project's early days were shattered by operational breakdowns, governance failures, and profound capability gaps. This phase was less about proactive, goal-directed (teleological) advancement and more about navigating a storm of dialectical conflict. Reactive crisis management became the norm as the Business-IT team and the wider organization grappled with escalating internal conflicts, resource limitations, and the awkward, often painful, manifestations of a system and an organization struggling to mature. The following subsections will empirically detail these multifaceted tensions, illustrating how the initial optimism of the 'Child Stage' gave way to organizational turbulence and the height of the project's crisis, driven by the clash between the inherited thesis and the emergent antithesis.

### **5.2.1 Breakdown of Lacking Governance Structures**

As the Fosbury Project transitioned into its active implementation phase, the underdeveloped governance structures established in the "Child Stage", which represents the flawed **thesis** for project oversight, collided with escalating complexities and new leadership dynamics. This collision formed a powerful **antithesis**, leading to a near-complete breakdown of effective governance, a hallmark of the project's turbulent "Teenager Stage.".

This chaotic state was explained by what the CTO perceived as a fundamental misunderstanding of the project's criticality and complexity at the highest levels. A lack of deep engagement from

top management, perhaps emergent from the project's early framing, led to significant underestimations. The CTO elaborated:

*To be very frank, ... I don't think people really understood what the project was about. And I think if those two [CEO and former COO] had understood [...] Henrik [the CEO] [...] always thought [IT systems were] 'easy'[...], that building a camera is hard, selling it is hard. If it goes wrong the whole company is in danger and I don't think [they] realized what kind of huge and important project this is." (Interview 5, Appendix 4, Alexander)*

This profound underestimation, the view that complex IT system overhauls were "easy" compared to product development, directly contributed to the governance vacuum.

This lack of initial top-level prioritization and anchoring was also stated by the former IT Director, Ulver, who reflected on the foundational principles of change management:

*I think it's a shame that Henrik [CEO] didn't prioritize it more from the start... if you read anything about IT change management, it's on page 1 of all theories and books, that if it's not anchored in top management, you can almost forget about it" (Interview 2, Appendix 4, Sam).*

While acknowledging the immense pressures Henrik faced: "Henrik was struggling with things that made a project like this look like the very lowest priority" (Ibid.), Ulver pointed to the direct consequences of this initial oversight. The failure to establish a clear, top-down strategic direction and defined responsibilities from the outset became another critical *antithetical* element that undermined the project's stability. As he stated, "*If it had been more anchored from the start with Henrik, and a strategy for how we run the business, and who runs what, then we wouldn't be reorganizing entire departments that work with it, while we're developing the project*" (Interview 2, Appendix 4, Sam).

In August 2023, a new Chief Operating Officer (COO) was appointed, however, the introduction of this additional leadership layer further exposed the absence of shared strategic alignment. The

COO and IT Director quickly realized they had diverging priorities and visions for the project. As Ulver reflected:

*When Ditte joined, it also became obvious that there were things we didn't agree on. And you can argue about what is right and wrong. But it definitely doesn't help the project when there isn't a shared understanding within the group about it.* (Interview 2, Appendix 4, Sam).

The leadership misalignments at the executive level further complicated the governance weaknesses already present in the Fosbury Project's structure. The distribution of decision rights were that the COO was in charge of the project and migration but decision making on the project was decentralized to Ulver and, misunderstood and under prioritized by top management (Appendix 4, Ulver). This lack of clear leadership and the ensuing power struggles mirrored a teenage identity crisis within the project's management.

From the consultancy's perspective, this fragmented leadership dynamic was also visible. One of the consultants observed that the new COO appeared to be managing the project primarily through individual performance metrics rather than shared governance: "She was managing on KPIs, not even really on budgets, but on personal KPIs. It was her personal thing to get this live" (Interview 6, Appendix 4, Timo).

According to the CTO, another critical mistake was made after the new COO started: "*and then they did a mistake to rush a migration because the new system was not in a state to be migrated to [...] we migrated into ruins more or less right*" (Interview 5, Appendix 4, Grosse). This epitomized the contradiction: a project intended to build a robust future was actively creating "ruins" due to a complete failure of strategic oversight.

The consultant, Timo, acknowledged the immense pressure: "at the end, we were all captured by the situation we were in[...]. facing complexity" (Interview 6, Appendix 4, Timo). The project became increasingly driven by these immediate pressures rather than a coherent, strategically grounded plan. Furthermore, the lack of direct engagement from key business and finance stakeholders in project discussions, with Business-IT often acting as a "proxy," limited the ability to surface tensions or balance priorities effectively (Interview 6, Appendix 4, Timo). Timo even

suggested a "fear culture" might have discouraged necessary escalation. This absence of a functional relational governance structure, as emphasized by Peterson (2003), left the project vulnerable to repeated disconnects, fully exposing the **contradiction** between the need for collaborative, informed governance and the chaotic reality of its execution.

### **5.2.2 Instability and Forced Change**

The Fosbury Project's implementation was marked by notable organizational turbulence, particularly in staffing and role continuity. This instability, according to CTO Alexander Grosse, was significantly worsened by the foundational structure and operational limitations of the Business-IT department itself. Grosse described it as an unusual and challenging setup: "I've never seen something like this business IT department before... people before [seemed to say]." He argued that with a siloed approach between departments and often relying on very limited engineering resources ("Øivind was more or less the only engineer the whole time"), for such a department, "scaling up and down is basically impossible[...] how can you scale up or down? it's impossible" (Interview 5, Appendix 4, Alexander). This inherent structural inability to adapt and scale affected the impact when several critical departures occurred during high-risk moments, without timely replacements. Jesper, the Zuora Product owner recalled that: "we were in Business-IT with a very small team. At the time, just two full-time employees and a student worker. Sam [Systems Engineer] left at a critical time."(Interview 4, Appendix 4, Jesper).

This was compounded by leadership disagreement over hiring and termination decisions. As Ulver explained: "I hired someone to start with; he resigned, so I hired another. Ditte didn't like him, so I was more or less forced to fire him. Then Ditte didn't want to participate in finding a replacement" (Interview 2, Appendix 4, Sam). The resulting vacuum in technical roles not only disrupted continuity but also deepened existing frustrations around ownership and prioritization within the Fosbury team.

Alexander Grosse expressed disbelief that key engineering roles were left unfilled despite mounting complexity in the Zuora and Salesforce systems. "They didn't rehire any engineers,

which is still a miracle to me. Sam had it, then moved to Builders, but even he wasn't replaced. Then it was just one person left, alone. Poor guy." (Interview 5, Appendix 4, Alexander).

These staffing decisions reflected broader fractures in the organization's approach to change management. Rather than proactively aligning talent to evolving system needs, personnel changes often appeared reactive, politically influenced, or simply neglected. The result was an understaffed, overstretched core team managing enterprise-scale platforms without the structural support typically required for such transitions.

### **5.2.3 Internal Conflicts and Leadership Fragmentation**

The Fosbury Project was significantly shaped by internal conflicts, both in terms of leadership capability and organizational alignment. Several interviewees pointed to a critical misalignment, an inherent *antithesis*, between the experience of those tasked with leading the project and the actual scale and complexity of the transformation underway. As Sam recalled, "The leadership, at least those running the project, had their own thoughts and experience on how to proceed, but it did not match the scale we were talking about" (Interview, Appendix 4, Sam). This gap suggested that the project's initial leadership thesis was ill-equipped for the challenges at hand.

This lack of high-level engagement from the C-suite was noted by interviewees as a significant deviation from typical project management best practices, an absence that itself acted as a powerful antithetical condition. As the external consultant observed, "Normally, when things get stuck, C-level steps in and says 'we're doing this, this, and this.' That didn't happen here" (Interview, Appendix 4, Timo). He emphasized the critical nature of this absence, stating, "It was open-heart surgery, and no one in the C-suite was driving it" (Interview, Appendix 4, Timo). This underscored a critical contradiction: a project of immense strategic importance was proceeding without the necessary executive guidance and intervention, leaving it vulnerable to drift and internal power struggles.

This gap between leadership involvement, their assumptions and organizational reality weakened the governance framework, leaving project direction fragmented and mismatched with the

demands of scaling enterprise IT systems. Leadership lacked not only technical depth, but also an appreciation for the broader structural and processual changes required to transition from a fragmented startup infrastructure to a coherent, scalable systems landscape.

The project also faced internal resistance from key stakeholder groups, particularly within the finance department. At the time, both the CFO, and the accounting manager, expressed strong opposition to the project's direction. As Ulver noted, the accounting manager in particular was a vocal critic and challenged the implementation effort throughout its early phases, further complicating stakeholder alignment (Interview, Appendix 4, Ulver). These frictions unfolded alongside a breakdown in program-level coordination. As CTO Alexander Grosse reflected, "Christian Houen was a nice guy, but he was simply in the wrong position. You cannot expect someone with his background to lead such a project; he was not set up for success" (Interview, Appendix 4, Alexander).

Simultaneously, the Builders department, led by the CTO and composed primarily of engineering staff focused on internal systems, underwent a major philosophical shift. As former VP of Engineering Jacob Frost was phased out and Alexander Grosse assumed broader influence, the organization experienced what one interviewee described as a "*religious shift*", a fundamental change in the underlying logic of governance and ownership across technical domains (Interview 2, Appendix 4, Ulver). While these overlapping leadership changes initially exacerbated governance fragmentation and strategic confusion, they also laid the groundwork for necessary realignment. The eventual absorption of Business-IT under the CTO was not merely reactive but increasingly inevitable, as it became clear that only through structural consolidation could Veo begin to restore strategic coherence and operational stability.

#### **5.2.4 Capability Gaps and Training Failures**

The governance breakdowns detailed above were not the only manifestation of the Fosbury Project's troubled 'Teenager Stage.' A critical *antithetical* element, directly stemming from the Phase 1 decision to outsource development of the systems without parallel internal capacity building, was the emergence of significant capability gaps and training failures. The initial *thesis* that Veo could seamlessly operate the complex new systems post-consultant handover collided with the Sam reality of a knowledge gap. As the systems moved into production, it became clear

that the Business-IT team lacked the training, documentation, and deep system knowledge required for effective day-to-day operations.

Without structured onboarding, documentation, or transition processes, engineers were forced into a reactive, trial-and-error approach to system operations, often under live, high-risk production conditions. As Øivind reflected, "I learned as I went [...] often we went live in production so that I could sit in Zuora and see how everything worked" (Interview 3, Appendix 4, Øivind). This lack of systematic knowledge transfer amplified operational risks, led to cascading technical debt, and created significant frustration among the engineers tasked with supporting critical customer-facing systems.

Despite Veo's increasing reliance on enterprise-grade systems, there was little internal investment made early on to train staff or build institutional knowledge. As Jesper noted, "there was very limited documentation. Typically, you would expect solution design documents and integration documents, but here there was almost none" (Interview 4, Appendix 4, Jesper) This absence of foundational technical knowledge created operational fragility and further compounded the risks of staff turnover.

Both Sam and Øivind underscored the organizational fragility of assuming that systems like Zuora and Salesforce could be maintained by junior staff without specialized onboarding. Øivind stressed that "if I had left, the project would have been set back months because there was no redundancy or documentation" (Interview 3, Appendix 4, Øivind).

### **5.2.5 Operational Misalignment and System Fragility**

Beyond the internal capability gaps, the 'Teenager Stage' was further characterized by severe operational misalignments and an inherent fragility in the new systems, another potent *antithetical* force challenging the project's viability. The Phase 1 *thesis* that the new systems would effectively support Veo's commercial operations was directly contradicted by the reality of configurations misaligned with practical business needs, largely due to fuzzy, evolving, or poorly communicated requirements. As Øivind, a systems engineer on the Fosbury Project recalled:

*We had huge problems because of requirements from Veo that money [should be] up front, but not for hardware that only needs to be authorized. For software, you should charge up front [...] we were complete novices in this field* (Interview 3, Appendix 4, Øivind).

Requirements were inconsistently communicated from senior management through multiple organizational layers, leading to repeated misinterpretations and implementation inefficiencies:

*It all points back to the consultants doing bad work, which is the standard thing. So it's a lot of wasted time going in between, lost in translation because a lot of middle management and people got some requirements from top management* (Interview 3, Appendix 4, Øivind).

Several internal stakeholders later reflected that simplifying the commercial checkout flow could have mitigated significant downstream complexity. As one senior team member stated,

*If I could have snapped my fingers, I would have removed the requirement of upfront payment. It would have simplified the checkout: we could have used the out-of-the-box invoice and payment flows. It would have solved so many problems. But then you get into sales commissions, finance needing money upfront, and it gets difficult. Cash flow is still very important for the company* (Interview, Appendix 4, Øivind).

These reflections underscore that the capability challenges faced during the Fosbury Project were not solely technical in nature. Rather, they reflected deeper organizational misalignments between business expectations, operational requirements, and technical system capabilities, misalignments that remained unresolved during the teenager phase and seeded further tensions throughout the project's later stages.

### **5.2.6 Emergence of Ownership and Role Conflicts**

The operational and capability crises of the 'Teenager Stage' clashed in the most explicit *dialectical conflict* of the Fosbury Project: the start of conflicting ownership and role conflicts. The *thesis* of Veo's initial organizational design, a clear separation between 'Product' (Builders) and 'Commercial IT' (Business-IT), was directly challenged by a powerful *antithesis* embodied

by the CTO, Alexander Grosse, and his fundamentally different philosophy regarding system ownership and technical capability:

*[...] you need a certain size to be able to scale up and down. [...] A few people from Business-IT moved to Builders initially, Isac, Christopher, and David, but they basically didn't rehire any engineers, which is still a miracle to me. Sam then had it alone, and even after he moved to Builders, they didn't rehire for him either* (Interview 5, Appendix 4, Alexander).

The initial mismatch between the project's ambition and the operational capacity of Business-IT contributed directly to emerging tensions over system ownership, strategic accountability, and long-term governance, tensions that would intensify as the Fosbury implementation moved forward.

The ownership and role conflicts that emerged during the Fosbury Project were not solely the result of immediate operational challenges; they also reflected deeper philosophical tensions regarding the division of labor between Business-IT and Builders. Under the leadership of former Vice President of Product, Jacob Frost, Builders had maintained a strict separation between product development and business system operations: "Frost kind of had an allergy against everything which had to do with finances or business" (Interview 5, Appendix 4, Alexander).

This organizational logic was echoed by Ulver, who noted, "Product only made product. That is, the codebase itself, the product itself, and its distribution. There was a clear interface between product and business systems, and that was the model we aimed to replicate" (Interview 2, Appendix 4, Sam).

However, the arrival of a new CTO, Alexander Grosse, initiated a profound philosophical shift. Grosse challenged the earlier separation, advocating for stronger integration between product and business systems to ensure architectural stability and scalability. As Ulver observed, "With Grosse's arrival and Frost's departure, the entire setup turned, it was a different religion" (Interview 2, Appendix 4, Sam). This fundamental clash of governance philosophies, between separation and integration, destabilized the original project structure and intensified tensions over

system ownership, technical accountability, and decision-making authority during the teenager phase of the Fosbury Project.

Beyond philosophical tensions, the structural limitations of the Business-IT organization became increasingly visible as implementation challenges escalated. Grosse criticized the operational design of Business-IT, arguing that it lacked the capacity to scale system support or respond flexibly to changing priorities. As he observed, "Scaling up and down was basically impossible [...] there was no inner structure where you could shift work based on priorities" (Interview 5, Appendix 4, Alexander). The transition of Business-IT personnel into Builders later revealed the latent capacity that had been previously untapped: "Suddenly, once they moved into Builders, roughly 20 people could help. You saw a lot of people working on tech, doing L3 tickets. It was clear that it could have been done earlier if governance had been aligned" (Interview 5, Appendix 4, Alexander).

Grosse further underscored the strategic misjudgment of the initial project setup: "I cannot believe how we can have the biggest project in company history and put it on a department which cannot scale. [Makes] Zero sense to me" (Interview 5, Appendix 4, Alexander).

This mismatch between project ambition and organizational capacity was a decisive driver of the ownership conflicts that ultimately led to Builders' absorption of Business-IT, initiating a partial stabilization of the Fosbury Project during its later stages. These ownership and role conflicts, rooted in both philosophical and structural misalignments, ultimately produced a forced integration under the CTO's leadership, setting the conditions for the gradual stabilization of Veo's internal systems in the subsequent phase.

### **5.2.7 Sub-Conclusion: Phase 2**

The 'Teenager Stage' of the Fosbury Project was a period of intense turmoil, where the flawed thesis inherited from its 'Child Stage', "a project that looked promising but was built on a shaky foundation, collided strongly with a powerful antithesis. This antithesis marked the breaking point. Governance collapsed, major skill gaps became visible, and the systems proved fragile and

misaligned with real needs. Conflicts over leadership, ownership, and strategy intensified, especially as the CTO began to challenge how things had been done. Like a turbulent adolescence, this period was marked by chaos, reactive management, and a loss of direction. The original goals were overshadowed by constant contradictions and they were bound up by a complicated web of systems and past choices. Instead of progress, the project began to fall apart, creating more problems than it solved. This crisis made it clear that the original approach wasn't working, and it forced the organization to rethink and eventually take the more structured path seen in the 'Adult' phase.

### **5.3 Phase 3 – Adulthood: Stabilization and Zuora 2.0**

The intense conflicts and chaos of the "Teenager Stage" led to a full-blown crisis, which became the thesis for Phase 3 of the Fosbury Project. It was clear the existing approach wasn't working, and the project was at risk of failure. In response, Veo entered what can be seen as the "Adulthood" stage, a period defined by teleological, goal-driven actions that served as the antithesis to the earlier breakdowns. These were not just quick fixes, but deliberate efforts to stabilize the situation, reorganize responsibilities, and apply lessons learned. The aim was to reach a new synthesis: a more mature, scalable, and sustainable approach to managing Veo's commercial IT systems.

At the height of the crisis, operational instability, poor governance, and a lack of resources led to a major, though initially unplanned, structural shift: the absorption of Business-IT into the Builders department. Jesper summed up the severity of the problem: Business-IT had "three people handling a system that did not work" for more than 20,000 subscribers. "*So you would obviously see that it's not going to work. [...] We got absorbed by Builders*" (Interview 4, Appendix 4, Jesper). This turning point marked the beginning of Veo's long and difficult journey toward stabilization and a more mature, "adult" approach to its IT transformation.

#### **5.3.1 Crisis Management and Reactive Actions**

The immediate response to the Fosbury Project's crisis, forming a crucial part of the *antithesis* to the prior chaos, was a series of decisive interventions. These actions, while born out of the

necessity to take apart the failing structures of Phase 2, were also inherently *teleological*, aimed at regaining control and initiating stabilization. In September 2024, top management made impactful decisions: the IT-Director and COO were fired, and the entire operations department was dissolved. This meant members of the Business-IT team were primarily relocated under the Builders department, run by CTO Alexander Grosse, an intervention he described as overdue and necessary.

*I think the main thing I did right was pushing for it. I asked several times, 'Can I give you people? Should we join forces?' I only succeeded when things got really bad, and then we just threw people at the problem. We needed to - the situation was dangerous* (Interview, Appendix 4, Alexander).

The absorption of Business-IT meant that Grosse could immediately assign additional resources to the project, a welcome change for the former Business-IT staff, who had long been overwhelmed. As Jesper described, this was "the first time [we] could hold [our] head above water." (Interview 4, Appendix 4, Jesper). What followed was the rapid mobilization of a 'force' of engineers, named 'Subsavers', tasked with clearing the significant backlog of internal support tickets that had accumulated during the chaotic rollout. This effort became the foundation of the Subsavers initiative. As Øivind summarized: "Phase one of Subsavers was to basically get ahead of the subscription ticket backlog, the support backlog, and to ensure that we get to a stable state where we can actually work on the system that we have." (Interview 7, Appendix 4, Øivind 2)

Sam similarly admitted that the team's absorption into Builders followed an internal recognition that existing capacity was insufficient to support even the earliest phases of the rollout:

*So 90% were still in [the] old system with 10% in [the] new, and this is just a rough number. We were still struggling to support these 10% in the new system with unstable products. [...] That's when we realized that we need more manpower, that's why the team was absorbed* (Interview, Appendix 4, Sam).

From the perspective of those absorbed, the transition brought immediate operational benefits and relief. Jesper described it as a turning point:

*It was actually only when we moved to Builders where I saw the first light in town. And I think that's when it went surprisingly well — we got a lot of resources given to us. We could allocate, and we started to get on top of all of the tickets that we had" (Interview 4, Appendix 4, Jesper).*

Reflecting on the long-term impact, he also noted a shift in organizational dynamics and governance clarity: "I think that part is introduced [...] we have a lot more resources, we have more mandate to say, because it's also not as much firefighting. The governance in general is definitely better." (Interview, Appendix 4, Jesper)

### **5.3.2. Subsavers**

Following the structural reorganization, the 'Subsavers' initiative emerged as a key *teleological* strategy within the broader *antithetical* response to the project's earlier failures. It was a focused, goal-directed effort to achieve the immediate objectives of reducing the critical support ticket backlog and beginning the arduous process of internal knowledge dissemination after Business-IT and the broader Operations department had been absorbed into other parts of the company. A period of intense internal firefighting began. This shift however, came with its own uncertainty: there was no immediate plan for how the absorbed employees would be integrated. As Øivind noted, "we were basically just told that we would move into Builders, but there wasn't really a plan for how we'd fit in yet." (Interview, Appendix 4, Øivind)

For several months, team structures remained informal, and responsibilities were fluid. The initial focus was simply on throwing manpower at critical issues. As Øivind added, "it just started slowly forming in the background." (Ibid.)

To support the influx of engineers during the Subsavers period, Jesper and Øivind took on the responsibility of internal training. Initially, the sessions followed a basic plan, but they quickly shifted to live debugging of actual L3 support tickets. The focus was on teaching how Veo's business systems worked, particularly the Salesforce-to-Zuora acquisition flow. As Øivind recalled, "some people were already doing tickets on their own, while others really weren't following along,"(Ibid.) highlighting the wide variation in experience and learning speed.

### 5.3.3 Formation of New Teams

As the initial firefighting efforts of the 'Subsavers' began to stabilize the most pressing operational issues, Veo moved towards a more structural *synthesis* to the previous organizational chaos: the deliberate formation of new, specialized teams. This was a clear *teleological* move, designed to instill clearer mandates, create defined ownership boundaries, and make long-term accountability, directly addressing the dysfunctions of the 'Teenager Stage.' After months of uncertainty about how former Business-IT members would be integrated, the absorption into Builders eventually led to the team being split.

One of the clearest outcomes of this restructuring was the initiation of what is still in development in the Revenue Platform team, named **Zuora 2.0**, a redesigned, more maintainable version of the subscription management system. While the original Fosbury implementation had been characterized by overcustomization, limited documentation, and hardcoded business logic, the new setup reflected a more deliberate and scalable architectural vision (Appendix 4). Led by Øivind and Jesper, the Revenue Platform team has begun systematically refactoring workflows, shifting away from fragile in-system logic and instead leveraging out-of-the-box Zuora functionality where possible. This approach is aiming not only to improve platform stability, but also marks a philosophical shift: instead of bending systems around legacy practices, the organization has begun adapting its operations to fit scalable system design principles.

### 5.3.4 Design Principles of Zuora 2.0

The development of 'Zuora 2.0' epitomizes the *synthesis* emerging from the Fosbury Project's earlier dialectical struggles, showcasing a profoundly *learned teleological* approach. The design principles adopted by the newly formed Revenue Platform team were a direct *antithetical* response to the overcustomization, complexity, and lack of internal control that plagued the initial implementation. A core learning from the Fosbury Project, now being actively applied, is this fundamental shift in how Zuora 2.0 is being implemented.

As Øivind put it, "We realized that the current setup was so complicated and clustered that it made more sense to start over completely." (Interview 7, Appendix 4, Øivind)

Moreover, as Jesper noted, Zuora is typically intended to be used with as many out-of-the-box features as possible, minimizing custom development and reducing operational complexity (Interview, Appendix 4, Jesper). This strategic reorientation stems from the realization that the original configuration had become overly convoluted and cost-intensive, both technically and financially. As Øivind explained, "it doesn't make sense to build on the solution we have today. We should just build everything from scratch with new assumptions" (Interview 7, Appendix 4). One of the key drivers behind this decision is the cost model of Zuora itself, "Workflows cost a lot of money. It's actually a huge number, we're paying per workflow task" (*Ibid.*). By designing a clean, controlled setup in a new tenant, the team aims to ensure that future workflows are both technically transparent and financially sustainable.

A major risk that is still a theme is what Øivind emphasized. Only a handful of people, still today, can confidently manage Zuora and Salesforce workflows, creating major organizational risk:

*We're still in a stage where there's so much legacy and tech already that we have such a high bus factor... If any of these three suddenly [leave], then we lose a lot of internal knowledge that we really need to be able to run this.* (Interview 7, Appendix 4, Øivind 2)

Another key rationale behind the clean-slate rebuild of Zuora 2.0 lies in the effort to reduce technical fragility and organizational dependency on a narrow set of experts. The legacy setup, shaped by external consultants and ad hoc internal input, resulted in a highly complex system architecture with a dangerously low "bus factor", only one or two individuals could confidently manage critical workflows. Rather than continuing to build atop this fragile foundation, the new implementation aims to structurally simplify the solution, improve maintainability, and ensure long-term resilience. Informed by the lessons of Fosbury, the team has also moved away from the risky "big bang" release strategy. As Øivind explained, "We'll introduce all the critical components in V1 [...] Then when we move to V2, no processes will change. It'll only be the backend that changes" (Interview 7, Appendix 4, Øivind 2). This phased approach reflects a broader shift in governance maturity: change is now sequenced, risk-assessed, and scoped

incrementally. Øivind further emphasized that technical execution would not begin until strategic alignment was secured: "We won't start doing any work until the scope and everything is clear. We have the project timeline. We understand the risks that we want to derisk and why we're doing it" (*Ibid.*). Together, these principles signal a departure from the urgency-driven, fragmented execution that characterized the Fosbury Project, and mark a more deliberate, learning-informed model of change.

### 5.3.5 Sub-Conclusion: Phase 3

The "Adulthood" phase of the Fosbury Project emerged directly from the *thesis* of acute crisis inherited from its tumultuous 'Teenager Stage'. In response, Veo enacted a series of decisive, *goal-directed & teleological actions* that served as the *antithesis* to the preceding chaos. This involved a significant organizational restructuring with the absorption of Business-IT into Builders, the immediate mobilization of resources like the 'Subsavers' to tackle operational fires, and the strategic formation of new, specialized teams.

This phase was fundamentally characterized by organizational learning and a move towards a new synthesis. The most potent symbol of this was the 'Zuora 2.0' initiative, where the painful 'legacy pain' of the initial implementation directly informed a more mature design philosophy emphasizing simplicity, maintainability, and a phased, risk-aware approach. While challenges like the 'bus factor' indicate that this adulthood is still an evolving state, Veo demonstrated a significant increase in maturity. The shift from reactive crisis management to proactive, planned development and improved governance marked a crucial step towards a more stable and sustainable alignment between its IT systems and growth ambitions.

## 6. Discussion

This chapter critically examines the findings from the case study of Veo Technologies and its Fosbury project. Moving beyond the presentation of results in Chapter 5, the aim here is to interpret their meaning, explain their significance in relation to the research question: *How does a tech startup balance its internal IT systems with its growth?*, and connect it to the theoretical framework and broader academic literature. This discussion will demonstrate how, specifically, the evolution

of Veo's IT systems, particularly through the Fosbury project, was a complex, non-linear process shaped by dialectical tensions, teleological efforts, and the inherent challenges of a rapidly scaling tech startup.

## 6.1 Interpretation of Results

The primary findings of this study on Veo Technologies and the Fosbury project indicated that the attempt to balance, and ultimately replace, internal IT systems in alignment with rapid growth was not a smooth, pre-planned transition but an emergent, often chaotic, and intensely learning-intensive journey. This evolution unfolded across three distinct metaphorical phases, each with its own logic and set of challenges:

### 6.1.1 Summary of Key Findings

**Phase 1 (Child Stage - Strategic Framing):** This initial phase was characterized by ambitious *teleological goals* to revamp commercial IT systems in response to clear operational pain points (e.g., lack of a customer 360 view, desire for scalability to support growth). However, this initial *thesis* for IT transformation was quickly spoiled by significant *antithetical elements* rooted in Veo's startup nature and early-stage immaturities. These included a siloed organizational structure unfit for a large-scale IT project, a reactive project initiation driven by an immediate data crisis (as seen in Garbage Can Model dynamics), and system selection and initial design processes heavily influenced by bounded rationality (e.g., reliance on key individuals' familiarity with certain systems like Salesforce, a "copy-paste" mentality for implementation). Furthermore, an over-reliance on external consultants without adequate internal knowledge transfer mechanisms, coupled with critically underdeveloped and often ineffective governance structures, laid a flawed and unstable foundation. The *emerging contradiction* was a significant and growing gap between the project's ambitious vision and the problematic reality of its execution.

**Phase 2 (Teenager Stage - Emergence of Contradictions):** This phase witnessed the dramatic eruption of *dialectical conflict*. The flawed *thesis* inherited from Phase 1, the project as initially conceived and set in motion, collided forcefully with a compelling *antithesis*. This antithesis comprised the Sam realities of operational breakdowns ("migrated into ruins"), a near-complete

governance collapse, escalating leadership misalignments and internal power struggles (e.g., between the COO and IT Director, and the emerging influence of the new CTO, Alexander Grosse, with his challenging "new religion" for Business-IT), critical capability gaps due to earlier oversights in training and knowledge transfer, and the inherent fragility of the poorly designed systems. This resulted in a state of acute *crisis and contradiction*: the Fosbury project was failing, original teleological goals were unachievable, and the organization was sunk in instability.

**Phase 3 (Adult Stage - Stabilization and Zuora 2.0):** Veo's response to the crisis of Phase 2 marked the beginning of this phase. The undeniable crisis itself became the *thesis* that necessitated radical change. The *antithesis* was a series of decisive, *goal-directed (teleological) actions*, largely driven by new leadership under CTO Grosse. These included significant organizational restructuring (absorption of Business-IT into Builders, departure of key former leaders), mobilization of resources to fight operational fires (the "Subsavers" initiative), the formation of new, specialized teams with clearer mandates, and, crucially, the development of a fundamentally new, *learned teleological approach* to system development embodied by the "Zuora 2.0" initiative (emphasizing simplicity, out-of-the-box functionality, and a phased, risk-aware implementation). The emerging *synthesis* was a more mature, stable (though still evolving) operational reality for Veo's commercial IT, characterized by embedded organizational learning, improved (though still developing) governance, and a more sustainable alignment between IT capabilities and growth ambitions.

Regarding the research question, these findings demonstrate that Veo's attempt to balance and replace its IT systems with growth was initially characterized by significant *imbalance*. This imbalance stemmed from internal organizational immaturities, decision-making processes constrained by bounded rationality and exhibiting Garbage Can characteristics, and underdeveloped governance mechanisms not suited for the scale of the transformation attempted. The "balance" was only achieved, or at least moved significantly towards, after a period of intense crisis and dialectical struggle. This conflict forced critical organizational learning and a fundamental restructuring of not only its IT systems and strategy but also its internal organization and governance approach to IT.

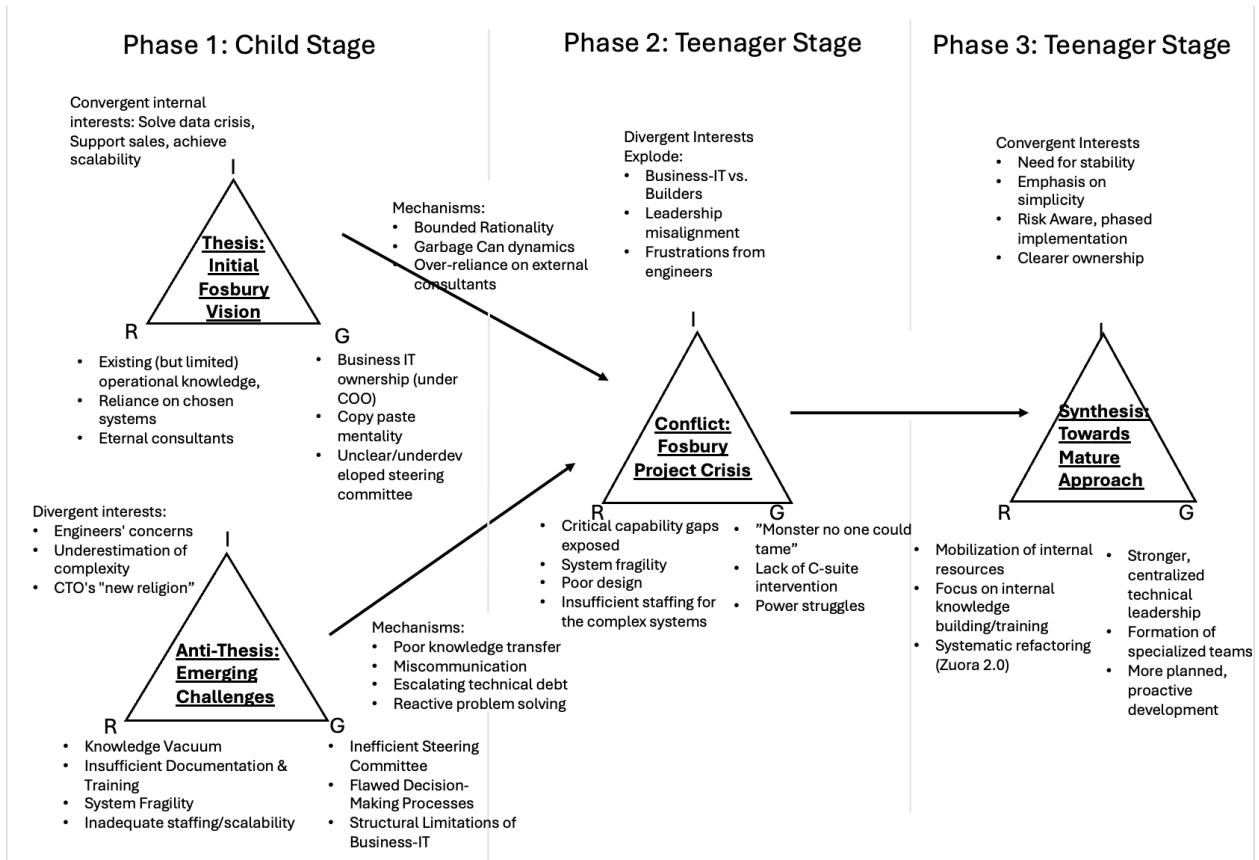


Figure 5: Dialectic process model describing the emergence of the Fosbury Project evolution

### 6.1.2 Interpretation Themes

The findings, interpreted through the study's theoretical framework (Dialectical Process Theory, Teleological Theory, Bounded Rationality, Garbage Can Model, and IT Governance concepts), reveal several key themes that explain the dynamics of the Fosbury project and Veo's IT evolution:

**Theme 1: IT System Change as a Conflict-Driven Process in Startups:** As depicted in Phases 1 and 2 of the evolutionary model (Figure 5), the evolution of the Fosbury project at Veo did not follow a linear, planned lifecycle model often presumed in traditional IT project management. Instead, its path is mostly understood as a *dialectical process* (Van de Ven & Poole, 1995). Change was driven by the interplay of *thesis* (initial plans, existing organizational structures, and dominant philosophies like Frost's separation of Product and Business IT), *antithesis* (conflicting operational realities, flawed decisions based on bounded rationality, the emergence of new leadership perspectives like Grosse's "new religion," and systemic breakdowns), and the resulting

*contradictions and crises*. For instance, the initial, optimistically framed thesis of Phase 1, laden with antithetical flaws (like the "copy-paste" design and weak governance), directly led to the crisis (contradiction) of Phase 2. Grosse's vision for integrated IT then acted as a new antithesis to the failing Business-IT setup, leading to a power struggle and eventual restructuring (synthesis in Phase 3). This dialectical interpretation helps answer how Veo *attempted* to balance IT and growth by showing that the path was fraught with conflict, and that progress often emerged from the resolution of these conflicts rather than smooth execution of initial plans. The initial attempts to balance often failed or exacerbated imbalances precisely because these underlying dialectical tensions were not adequately recognized or managed.

**Theme 2: Balancing Goals and Chaos in IT Change:** While dialectical conflict, as illustrated throughout the phases in Figure 5, was a primary driver of change, teleological (goal-directed) efforts were constantly present, though their nature, focus, and effectiveness evolved significantly across the phases. In **Phase 1** (Figure 5), the initial teleological goals (customer 360, scalability, operational efficiency, captured under "Convergent internal interests" of the Thesis) were clear and ambitious, forming the project's basis. However, the *methods* chosen to achieve these goals (the antithetical elements like flawed design and over-reliance on consultants) fundamentally undermined the likelihood of success. In **Phase 2**, teleological efforts became fragmented and often reactive. The COO, Ditte, attempting to push the project forward based on personal KPIs, or the Business-IT team struggling to implement the systems amidst mounting problems, represented teleological actions. However, these were largely subverted or distorted by the overwhelming dialectical conflicts, governance breakdowns, and resource constraints. The original goals became obscured by the immediate need to manage crises. **Phase 3** saw the emergence of a *renewed and more informed teleology* from the crucible of the dialectical crisis. The "Subsavers" initiative was a focused teleological effort to stabilize. More significantly, the "Zuora 2.0" initiative, with its clear design principles, emphasis on out-of-the-box functionality, phased rollout, and prerequisite of clear scoping, exemplifies *learned teleology*. The goals were still present (a stable, scalable system), but the *approach* was now deeply informed by the failures of the past. This interplay demonstrates that in a startup context, initial teleological pursuits might be naive or flawed, leading to dialectical conflict, which, if navigated (often painfully), can lead to more mature and effective goal-setting and execution strategies. The balance between IT and growth is thus not just about

having goals, but about developing the capacity to pursue them effectively, a capacity often forged in response to conflict.

**Theme 3: Bounded Rationality and Messy Decision-Making in Early IT Choices:** While modern IT strategy often emphasizes Data-Driven Decision-Making (DDDM) (see Section 2.2.1), the Fosbury Project at Veo frequently deviated from this ideal. In Phase 1 (the 'Child Stage') and extending into Phase 2, key decisions were shaped more by Bounded Rationality (Simon, 1991) than by comprehensive data analysis. For example, Salesforce was selected largely due to personal familiarity, while Zuora's "copy-paste" implementation reflected cognitive shortcuts rather than validated design. The project's scope and complexity were severely underestimated, and reliance on external consultants without robust internal validation further reflected decision-making under pressure, limited information, and urgency, factors creating the well-known "data-action gap" (Juneja & Klinger, 2024).

These sacrificing behaviors had lasting consequences, contributing to the systemic tensions of Phase 1. The Garbage Can Model (Cohen, March & Olsen, 1972) also helps explain early project dynamics, where problems, solutions, and stakeholders drifted in and out of decision arenas, reflecting a form of "organized anarchy." This ad hoc and reactive approach to IT decision-making limited Veo's ability to align IT with growth, embedding technical debt and friction that later amplified as the company scaled.

**Theme 4: IT Governance as Emergent, Reactive, and Politically Contested in Scaling Startups:** Veo's IT governance for the Fosbury project, as illustrated by the evolution of the 'G' components across the three phases in Figure 5, was not proactively designed with the foresight and comprehensiveness suggested by normative frameworks (e.g., Weill & Ross, 2004).. Instead, it emerged *reactively* and was often highly contested, particularly as the company scaled and the project's complexity became apparent. **Phase 1** was characterized by a significant lack of effective structural, processual, and relational governance mechanisms (Peterson, 2003). Top management underestimated the project's complexity, leading to insufficient attention and oversight. Even when a steering committee was formed, its effectiveness was questionable, with technical expertise often excluded from key discussions. The governance breakdown in **Phase 2** was a direct consequence of this initial underdevelopment, amplified by leadership misalignments and the

sheer pressure of implementation failures. Decision-making became "a monster no one could tame." The changes in **Phase 3**, CTO Alexander Grosse consolidating ownership of commercial IT within Builders, the establishment of new specialized teams with clearer mandates, and the more planned, risk-aware approach to "Zuora 2.0"—represent the *emergence* of more mature (though still developing) governance. This shift was not a planned evolution but a necessary adaptation born out of crisis and the dialectical struggle for control and competence. This emergent governance story is central to understanding the challenge of balancing IT and growth: effective governance is not a given but must be actively constructed and often reconstructed, especially in the turbulent environment of a scaling startup.

**Theme 5: Organizational Learning Shaped by Conflict and Crisis:** The Fosbury project, despite its major challenges and near-failure graphically represented in the 'Conflict' stage of Figure 5, ultimately served as a significant, but painful, organizational learning experience for Veo. The failures, conflicts, and operational breakdowns of Phases 1 and 2 (the dialectical struggle) directly informed the more mature and considered approaches observed in Phase 3. The design principles of "Zuora 2.0", emphasizing simplicity, out-of-the-box functionality, avoiding over-customization, phased rollouts, and clear scoping before execution—are explicit manifestations of this learning. They represent a conscious effort to avoid repeating the mistakes of the initial implementation. The very decision to rebuild Zuora from scratch, rather than continuing to patch a deeply flawed system, indicates a significant learning threshold being crossed. This theme powerfully connects the dialectical process (conflict and crisis leading to change) with teleological outcomes (new, more effective goals and methods for achieving them). It suggests that for startups, the path to developing robust IT capabilities and achieving a sustainable balance with growth may necessarily involve navigating such difficult learning cycles, where crises act as catalysts for fundamental re-evaluation and adaptation.

### 6.3 Implications

The findings from this in-depth case study of Veo Technologies and the Fosbury project offer several important implications for both theory and practice, particularly concerning how rapidly scaling tech startups navigate complex IT system transformations.

### 6.3.1 Theoretical Implications

Firstly, this study contributes to the academic understanding of IT implementation and organizational change in several ways. In one way the Veo case demonstrates that commercial IT system replacement and implementation in high-growth startup environments rarely follow purely clear, linear, or predictable lifecycle models. The Fosbury Project's journey, from a reactive start and flawed early decisions to crisis and eventual learning, shows the limits of traditional models that don't fully account for how startups actually work. In fast-moving, complex environments like Veo, factors like organizational immaturity, limited information, internal politics, and unexpected change play a big role. This study suggests that theories of IT implementation should better reflect these real-world complexities and the non-linear nature of change in startups.

Secondly, the findings lend strong empirical support from *dialectical process theories* (Van de Ven & Poole, 1995) as a valuable lens for understanding complex, conflict-ridden IT-driven organizational change. The Fosbury Project's different phases, shaped by thesis, antithesis, and the resolution of contradictions, resulting in a *synthesis*, show that conflict and power struggles can actually drive change, not just block it. This study highlights how a *dialectical perspective* helps explain the messy, often unpredictable nature of IT transformation in startups.

Thirdly, while dialectical tensions were central to the Fosbury Project, the case also shows that *teleological* (goal-driven) efforts were always present, even if they changed over time. In Phase 1, the project started with clear goals, though poor execution made them hard to achieve. During the chaos of Phase 2, different people still tried, often unsuccessfully, to push the project toward certain outcomes. Most importantly, in Phase 3, we see a kind of "*learned teleology*," where new and more realistic goals (like those behind Zuora 2.0) were shaped by lessons from earlier failures. This suggests that a *hybrid model*, combining dialectical conflict and teleological learning, helps explain how IT systems evolve in startups, where early plans often fail but can lead to stronger, more informed direction through the resolution of tensions.

Lastly, this research aims to offer a counterpoint to more prescriptive and static IT governance literature (e.g., Weill & Ross, 2004; Peterson, 2003) by illustrating how IT governance often emerges reactively and is forged through struggle and crisis in startups, rather than being

comprehensively designed and implemented upfront. Veo's journey from underdeveloped and dysfunctional governance in the early phases of the Fosbury project to a more structured (though still developing) approach in Phase 3, driven by necessity and new leadership, highlights that for scaling companies, effective IT governance may be less about adopting a framework and more about an iterative process of learning, adaptation, and power realignment in response to critical challenges.

### **6.3.2 Practical Implications**

Veo's experience with the Fosbury Project can offer useful lessons for other tech startups, especially for leaders and teams managing IT change in fast-growing companies with limited resources.

Startup and scaleup leaders should anticipate that significant IT changes will inevitably involve conflict, especially between the urgency of business needs and the requirements for engineering stability, between established practices and new technological paradigms, or between differing leadership philosophies. Recognizing these tensions not only as obstacles but as potential starters for necessary debate, learning, and ultimately, more robust solutions is crucial. Creating forums for constructive disagreement can be more productive than attempting to suppress conflict. Also, IT governance in a startup cannot be a static, rigid framework. It must be dynamic, adaptable, and mature alongside the organization. While comprehensive frameworks might seem too burdensome for early-stage companies, foundational elements like clear decision rights for critical IT projects, consistent C-level engagement and guidance, and inclusive stakeholder involvement (especially technical expertise in strategic discussions) are vital from the outset, even if they need to be iterated upon as the company scales.

Moreover, the Veo case vividly illustrates the long-term dangers of underestimating IT complexity, over-relying on individual familiarity or simplistic analogies (like the "copy-paste" mentality for system design), and failing to invest in internal knowledge building during initial system choices and implementations. These early missteps, often driven by bounded rationality under pressure, can create substantial technical and organizational debt that becomes increasingly

costly to address as the company grows. Meanwhile, external expertise is often indispensable for startups lacking specific skills, reliance on consultants must be managed strategically. This includes ensuring strong internal project ownership and oversight, defining clear and realistic project scopes (avoiding the "everything-at-once" trap), insisting on robust mechanisms for knowledge transfer to internal teams, and actively working to avoid creating "black boxes" that hinder internal understanding and long-term system maintainability.

Furthermore, IT project challenges, and even significant failures like those experienced by Veo during the Fosbury project's "Teenager Stage," can be important (albeit painful) learning opportunities. Creating a culture where it is safe to acknowledge mistakes (as Veo's CEO reportedly did), conduct post-mortems, and consciously adapt practices, processes, and even structures based on these lessons is crucial for building organizational resilience and improving future IT endeavors.

Ultimately, this case shows how important a strong technical leader can be. A CTO or COO isn't just responsible for making things work , they also need to challenge bad assumptions, push for needed changes, promote smarter system design, and make sure IT supports the business instead of slowing it down.

## 6.4 Limitations

It's important to recognize certain limitations which the study of Veo Technologies and the Fosbury Project brings, which may affect how the findings are interpreted or applied to other cases in the future.

There are limitations to the using a single-case study design which this research employed. Focusing in-depth on only one organization and on only one specific, complex commercial IT transformation project. While this approach allows for a rich, contextualized exploration of the dynamics of IT system change and growth in a tech startup, the findings may not be directly statistically generalizable to all tech startups or all large-scale IT projects. The unique

organizational culture, leadership dynamics, specific market pressures, and contingent events at Veo undoubtedly shaped its journey. However, the study aims for *analytical generalizability*, where the insights and theoretical interpretations derived from this case can inform the understanding of similar phenomena in other comparable contexts.

Moreover, the primary data for this study was gathered through retrospective interviews, where participants recalled events, decisions, and experiences that often occurred months or even years prior. While efforts were made to triangulate information where possible (e.g., through project timelines or different stakeholder accounts), reliance on memory can introduce potential biases, such as selective recall, post-hoc rationalization of events, or the collapsing of timeframes (Bell, et al., 2019). The emotional intensity of certain project phases might also have influenced how events were remembered and recounted.

Given that most of the interviews were conducted in English, which is not the mother tongue for many interviewees or the researcher, the potential for misinterpretation by either party must be acknowledged. For interviews held in Danish, nuances could have been lost or altered during the translation process into English for analysis and citations. While care was taken, the subtleties of language can impact the depth of understanding and interpretation.

The analysis of the qualitative data collected is inherently interpretive. As the researcher, the application of the chosen theoretical lenses (dialectical process theory, teleology, bounded rationality, Garbage Can Model, IT governance) to the empirical data involves a degree of subjective judgment in identifying themes, patterns, and causal links. While this interpretive approach allows for a deep and nuanced understanding of complex social phenomena, it is acknowledged that other researchers with different theoretical perspectives or backgrounds might draw different or additional interpretations from the same dataset. Furthermore, the fact that this study relied solely on qualitative data from interviews and observations of project dynamics is a factor. While this provided rich insights into the "how" and "why" of events, the lack of quantitative metrics, such as more detailed system performance data before and after specific changes in the Fosbury project, specific financial costs associated with project delays or rework, or quantifiable measures of employee/user satisfaction over time, limits the ability to objectively

measure the precise impact of certain interventions or the overall efficiency of different project phases.

Lastly, the narrative and findings are inevitably shaped by the specific access granted to the researcher within Veo Technologies and the particular individuals who participated in the study. While a range of stakeholders were interviewed both present and former employees involved in the project, the perspectives of more of those who may have left the company during the more turbulent phases of the Fosbury project, or those in different departments less directly involved but still affected, might offer additional nuances that were not fully captured. The relationships established and the trust built with participants also influence the nature and depth of the information shared.

## 6.5 Future Research Directions

The insights gained from Veo's experience with the Fosbury project open several promising avenues for future research. To build upon this study's findings and address its limitations, further investigation could explore the dynamics of IT transformation in rapidly scaling technology companies through comparative case studies. Such research would help identify common patterns, critical success factors, and variations in how different organizations navigate dialectical tensions and evolve their IT governance when undergoing similar large-scale system replacements or implementations.

Furthermore, longitudinal research on IT governance maturation in startups is needed. Tracking the evolution of IT governance capabilities and system changes over an extended period, beyond a single major project, could provide deeper insights into the ongoing interplay of dialectical and teleological processes and how these governance capabilities develop, or fail to develop, sustainably over time.

The influence of specific decision-making styles also merits deeper exploration. Future studies could examine how bounded rationality and Garbage Can dynamics specifically influence different types of IT decisions (e.g., infrastructure choices versus application development, or build-versus-

buy decisions) within the unique resource-constrained and uncertain environments characteristic of startups, offering valuable insights for practitioners.

Delving more deeply into the mechanisms of change, research could focus on dialectical conflict as a specific driver for the emergence and evolution of IT governance structures, roles, and practices in startups. Understanding how these conflicts are navigated and resolved to produce more mature governance would contribute significantly to both change management and IT governance theories.

The crucial role of leadership in navigating IT transformation crises also presents a rich area for inquiry. Investigating how different leadership styles and interventions at various levels (e.g., CEO, COO, CTO) impact the management of dialectical tensions, the resolution of crises, and the success of subsequent teleological efforts during major IT changes in startups could offer practical guidance for leaders in similar situations.

Finally, the impact and management of external consultants in startup IT projects requires more systematic research. Studies focusing on optimal engagement models that maximize knowledge transfer, ensure alignment with long-term strategy, and mitigate risks related to dependency and the creation of internal "black boxes" would be highly beneficial for startups seeking external expertise for critical IT initiatives.

## 7. Conclusion

This thesis set out to answer the central research question: ***How does a tech startup balance its internal IT systems with its growth?*** The in-depth longitudinal case study of Veo Technologies and its Fosbury Project reveals that achieving such balance is far from straightforward. Rather than a linear, predictable path, the journey is a complex, emergent process deeply influenced by compounding a series of decisions, often made under conditions of uncertainty, pressure, and KPIs, and the inherent difficulties of aligning evolving IT capabilities with the tough demands of rapid scaling. What may seem like an obvious misstep in hindsight was often an unrecognised obstacle or a pragmatic choice at the moment of decision.

The investigation into Veo's experience with the Fosbury Project illustrates that the "balancing act" of aligning IT with growth is an iterative process, profoundly driven by dialectical tensions, emergent crises, and valuable organizational learning. This journey unfolded across three metaphorical phases: 'Child,' 'Teenager,' and 'Adult', each revealing distinct aspects of how this balance is sought and sometimes lost.

This study shows that tech startups like Veo balance IT systems with growth not through foresight alone, but by navigating the conflicts and crises that emerge when growth outpaces IT capabilities. Early decisions often prioritize speed, creating imbalances. Specifically, unclear or contested IT governance, a key dialectical tension discussed in the theoretical framework, often generated significant organizational disequilibrium. This disequilibrium, compounded by clashes between existing systems and escalating scaling demands, typically culminated in crisis-driven restructuring and organizational learning, eventually resulting in more sustainable, goal-directed IT strategies. Balance, therefore, is not static, but a hard-earned and evolving state.

The findings from Veo's journey highlight key implications. **Theoretically**, they show that IT transformation in startups is a non-linear, conflict-driven process, challenging traditional lifecycle models. Initial teleological efforts often clash with internal tensions, with IT governance emerging reactively through struggle. **Practically**, startups should anticipate and manage IT-related conflicts, adopt flexible governance, control complexity and consultant use, and foster a culture of learning from failure. Strong, adaptable technical leadership is critical in steering this turbulent path.

While this single-case study offers rich, contextual insights, its findings are specific and aimed at analytical, not statistical, generalization. Future research could expand on these insights through comparative case studies to explore broader patterns in IT alignment, governance maturation, and leadership during IT crises.

In closing, the alignment of internal IT systems with rapid growth is a large, yet fundamental, challenge for tech startups. Veo's experience with the Fosbury Project illustrates that while this path is full of dialectical tensions and emergent crises, it is also ground for organizational learning

and maturation. Successfully navigating this balancing act, transforming conflict into learning and crisis into renewed, informed purpose. This fact is a critical factor of a startup's resilience and ultimate success in the dynamic technological landscape. Functional, scalable, and strategically aligned IT truly forms the backbone upon which sustained growth is built.

## Bibliography

Aasi, P., Rusu, L., & Vieru, D. (2017). The Role of Culture in IT Governance Five Focus Areas.

*International Journal of IT/Business Alignment and Governance*, 8(2), 42–61.

<https://doi.org/10.4018/ijitbag.2017070103>

Aldoubaee, A., Hassan, N. H., & Rahim, F. A. (2023). A Systematic Review on Blockchain Scalability. *International Journal of Advanced Computer Science and Applications*, 14(9). <https://doi.org/10.14569/ijacsa.2023.0140981>

Aminuddin Hamdat, Ceskakusumadewi B, Samalam, A. G., Muhammad Rizal, & Izaac L.D Lawalata. (2024). The Impact of Management Information Systems on Decision-Making Efficiency. 1(2), 56–74. <https://doi.org/10.70184/qpw1rq78>

Andrade, P. R. M. de, Albuquerque, A. B., Teófilo, W. D., & da Silva, F. A. (2016). Change management: Implementation and benefits of the change control in the information technology environment. *arXiv*. <https://arxiv.org/abs/1603.03110>

Andiso Pingilili, Ntebele Letsie, Gift Nzimande, Bonginkosi Thango, & Lerato Matshaka. (2025). Guiding IT Growth and Sustaining Performance in SMEs Through Enterprise Architecture and Information Management: A Systematic Review. *Businesses*, 5(2), 17. <https://doi.org/10.3390/businesses5020017>

Antonio, M., Orestis Terzidis, Lütz, P., & Heblich, B. (2024). Critical decisions at the early stage of start-ups: a systematic literature review. *Journal of Innovation and Entrepreneurship*, 13(1). <https://doi.org/10.1186/s13731-024-00438-9>

Ashurbaev, O., & Sadullaev, I. (2024). Top Challenges Startups Face: A Literature Review. *Zenodo*, 4(12). <https://doi.org/10.5281/zenodo.14561370>

Bajgoric, N., & Moon, Y. (2009). Enhancing systems integration by incorporating business continuity drivers. *Industrial Management & Data Systems*, 109(1), 107–125. <https://doi.org/10.1108/02635570910926609>

Bell, E., Bryman, A., & Harley, B. (2019). *Business Research Methods* (5th ed.). Oxford University Press.

Benson, J. K. (1977). Organizations: A dialectical view. *Administrative Science Quarterly*, 22(1), 1–21. <https://doi.org/10.2307/2391741>

Bondarouk, T., Parry, E., & Furtmueller, E. (2017). Electronic HRM: four decades of research on adoption and consequences. *The International Journal of Human Resource Management*, 28(1), 98–131. <https://doi.org/10.1080/09585192.2016.1245672>

Borlovan, P. (2025). *The Investor's Dilemma: Balancing Intuition and Data in Startup Funding*.

Seedblink.com. Accessed May 04, 2025

<https://seedblink.com/2024-03-28-the-investors-dilemma-balancing-intuition-and-data-in-startup-funding>

Bridgen, B. (2025, January 27). 8 new IT challenges businesses face in 2025. *Teamwork.com*. Accessed 30 April, 2025: <https://www.teamwork.com/blog/it-challenges/>

Brown, A. E., & Grant, G. G. (2005). Framing the Frameworks: A Review of IT Governance

Research. *Communications of the Association for Information Systems*, 15(8).  
<https://doi.org/10.17705/1cais.01538>

Ciacci, A., Ivaldi, E., Penco, L., & Testa, G. (2024). Measuring Digital Sustainability Paying Attention to the Economic, Social, and Environmental Dimensions: A European Perspective. *Social Indicators Research*, 177(1). <https://doi.org/10.1007/s11205-024-03459-9>

Christiaan Verwijs, & Russo, D. (2024). Do Agile scaling approaches make a difference? An empirical comparison of team effectiveness across popular scaling approaches. *Empirical Software Engineering*, 29(4). <https://doi.org/10.1007/s10664-024-10481-10>

Cohen, M. D., March, J. G., & Olsen, J. P. (1972). A garbage can model of organizational choice. *Administrative Science Quarterly*, 17(1), 1–25. <https://doi.org/10.2307/2392088>

De Keyser, B., Guiette, A., & Vandenbempt, K. (2019). On the dynamics of failure in organizational change: A dialectical perspective. *Human Relations*, 74(4), 001872671988411. <https://doi.org/10.1177/0018726719884115>

Dekkers, A. (2024). Regulatory and industry standards compliance in hardware technology startups (Master's thesis, Delft University of Technology). TU Delft Institutional Repository. <https://repository.tudelft.nl/islandora/object/uuid%3Aa976df95-9ae4-4e82-82c8-ad2600869358>

Drechsler, A., & Weißschädel, S. (2017). An IT strategy development framework for small and medium enterprises. *Information Systems and E-Business Management*, 16(1), 93–124. <https://doi.org/10.1007/s10257-017-0342-2>

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., & Yoo, Y. (2015). Distributed tuning of boundary resources: The case of Apple's iOS service system. *MIS Quarterly*, 39(1), 217–243. <https://doi.org/10.25300/MISQ/2015/39.1.10>

Forbes Technology Council. (2024, April 8). 20 Specific Challenges Tech Startups Face And How To Overcome Them. *Forbes*. Accessed February 14, 2025: <https://www.forbes.com/councils/forbestechcouncil/2024/04/08/20-specific-challenges-tech-startups-face-and-how-to-overcome-them/>

Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking Qualitative Rigor in Inductive Research. *Organizational Research Methods*, 16(1), 15–31. <https://doi.org/10.1177/1094428112452151>

Greiner, L. E. (1972). Evolution and Revolution as Organizations Grow. *Harvard Business*

*Review*, 10(4), 397–409. <https://doi.org/10.1111/j.1741-6248.1997.00397.x>

Hanifzadeh, F., Talebi, K., & Jafari-Sadeghi, V. (2024). Scalability of startups: The impact of entrepreneurial teams. *Journal of Global Entrepreneurship Research*, 14(1), Article 15. <https://doi.org/10.1007/s40497-024-00383-7>

Hannan, M. T., & Freeman, J. (1977). The population ecology of organizations. *American Journal of Sociology*, 82(5), 929–964. <https://doi.org/10.1086/226424>

Herstatt, C., Tietze, F., Nagahira, A., & Probert, D. (2006). The Chief Technology Officer (CTO) in Theory and Practice - A Literature Review and Results from Field Research in Japan. *SSRN Electronic Journal*, 41. <https://doi.org/10.2139/ssrn.2118082>

Highspot Team. (2025, March 8). The Top RevOps Frameworks that Drive Growth and Alignment. *Highspot*. <http://highspot.com/blog/revenue-operations-framework/>

Huang, Linmei. (2025). From IT-Startups to Tech Giants: How Social Connections Influence Investors' Perspective on Valuations in the Technology Space. *CUNY Academic Works*. Accessed May 2025: academicworks.cuny.edu/gc\_etds/6137/

Ivančić, L., Vukšić, V., & Spremić, M. (2019). Mastering the Digital Transformation Process:

Business Practices and Lessons Learned. *Technology Innovation Management Review*, 9(2), 36–50. <https://doi.org/10.22215/timreview/1217>

Iyer, L. S., Gupta, B., & Johri, N. (2005). Performance, scalability and reliability issues in web applications. *Industrial Management & Data Systems*, 105(5), 561–576. <https://doi.org/10.1108/02635570510600098>

Joseph, G., Aboobaker, N., & K.A., Z. (2021). Entrepreneurial cognition and premature scaling

of startups: a qualitative analysis of determinants of start-up failures. *Journal of Entrepreneurship in Emerging Economies*, ahead-of-print(ahead-of-print).

<https://doi.org/10.1108/jeee-11-2020-0412>

Juneja, R., & Klinger, J. (2024). *Top 5 IT challenges leaders are facing (and solutions to them)*. Elastic.co. Accessed May 7, 2025: <https://www.elastic.co/blog/top-5-it-challenges-leaders-face-solutions>

Koseoglu, Rana. (2024). *SCALING STRATEGIES and BUSINESS PERFORMANCE IMPROVEMENTS for STARTUPS in the DIGITAL ECOSYSTEM CC-BY-NC*.

Kotter, J. P. (1995). Leading change: Why transformation efforts fail. *Harvard Business Review*, 73(2), 59–67. <https://hbr.org/1995/05/leading-change-why-transformation-efforts-fail-2>

Lee, S. (R.), & Kim, J. D. (2024). When do startups scale? Large-scale evidence from job postings. *Strategic Management Journal*. Advance online publication. <https://doi.org/10.1002/smj.3540>

Leigh, K. (2016). Models of organizational change. In A. Farazmand (Ed.), *Global encyclopedia of public administration, public policy, and governance* (pp. 1–5). Springer. [https://doi.org/10.1007/978-3-319-31816-5\\_47-1](https://doi.org/10.1007/978-3-319-31816-5_47-1)

Lewin, K. (1951). *Field theory in social science: Selected theoretical papers*. Harper & Row.

Maur, W. I. D., Walbeek, W. V., & Batenburg, R. (2009). A framework for integrating IT governance and business/IT alignment principles. *International Journal of Business Innovation and Research*, 3(5), 555. <https://doi.org/10.1504/ijbir.2009.027177>

Mokogwu, C., Achumie, G. O., Adeleke, A. G., Okeke, I. C., & Ewim, C. P.-M. (2024). A data-driven operations management model: Implementing MIS for strategic decision making in tech businesses. *International Journal of Frontline Research and Reviews*, 3(1), 001-019. <https://doi.org/10.56355/ijfrr.2024.3.1.0027>

Molete, O. B., Mokhele, S. E., Ntombela , S. D., & Thango, B. A. (2025). The Impact of IT Strategic Planning Process on SME Performance: A Systematic Review. *Businesses*, 5(1), 2. <https://doi.org/10.3390/businesses5010002>

Mottola, O. (2021). *The Revenue Operations (RevOps) Framework: A Qualitative Study of Industry Practitioners*. Accessed May 07, 2025:

[https://digitalcommons.harrisburgu.edu/isem\\_dandt/1](https://digitalcommons.harrisburgu.edu/isem_dandt/1)

Nelson, R. R., & Winter, S. S. (1982). *An evolutionary theory of economic change*. Harvard University Press.

Orugunta, M. (2025). Scaling startups: Lessons from cloud-native success stories. *International Journal of Information Technology and Management Information Systems*, 16(2), 1121–1138. [https://doi.org/10.34218/IJITMIS\\_16\\_02\\_071](https://doi.org/10.34218/IJITMIS_16_02_071)

Paula, A., & Zilber, S. N. (2023). Digital Startups Survivor and Scalability: Identifying Variables for Accelerated Growth. *Open Journal of Business and Management*, 11(05), 1927–1947.

<https://doi.org/10.4236/ojbm.2023.115107>

Pingilili, A., Letsie, N., Nzimande, G., Thango, B., & Matshaka, L. (2025). Guiding IT Growth

and Sustaining Performance in SMEs Through Enterprise Architecture and Information Management: A Systematic Review. *Businesses*, 5(2), 17.

<https://doi.org/10.3390/businesses5020017>

Peterson, R. R., 2003. Integration Strategies and Tactics for Information Technology Governance. In: *Strategies for Information Technology Governance*. Empresa, Spain: Panama: IGI Global, pp. 37-80.

PremierNX. (2025, January 10). 2025 Digital Transformation (DX) Blueprint: Overcoming Legacy System Challenges for Lasting Success. *Premier NX*. Accessed March 02, 2025: <https://premiernx.com/blog/digital-transformation/>

Quinn, R. E., & Cameron, K. (1983). Organizational life cycles and shifting criteria of effectiveness: Some preliminary evidence. *Management Science*, 29(1), 33–51. <https://doi.org/10.1287/mnsc.29.1.33>

Ransbotham, S., Kiron, D., & Gerbert, P. (2025, January 28). The Great Power Shift: How Intelligent Choice Architectures Rewrite Decision Rights. *MIT Sloan Management Review*. <https://sloanreview.mit.edu/article/the-great-power-shift-how-intelligent-choice-architectures-rewrite-decision-rights/>

Sambamurthy, V., & Zmud, R. W. (1999). Arrangements for Information Technology Governance: A Theory of Multiple Contingencies. *MIS Quarterly*, 23(2), 261. <https://doi.org/10.2307/249754>

Santisteban, J., & Mauricio, D. (2017). Systematic literature review of critical success factors of Information Technology startups. *Academy of Entrepreneurship Journal*, 23(2), 1–23. [https://www.researchgate.net/publication/322094432\\_Systematic\\_literature\\_review\\_of\\_critical\\_success\\_factors\\_of\\_Information\\_Technology\\_startups](https://www.researchgate.net/publication/322094432_Systematic_literature_review_of_critical_success_factors_of_Information_Technology_startups)

Santisteban, J., Morales, V., Bayona, S., & Morales, J. (2023). Failure of Tech Startups: A Systematic Literature Review. *CSEI: International Conference on Computer Science, Electronics and Industrial Engineering (CSEI)*, 111–126. [https://doi.org/10.1007/978-3-031-30592-4\\_9](https://doi.org/10.1007/978-3-031-30592-4_9)

Saunders, C., & Chan, Y. E. (2002). Rapid-growth firms: The challenge of managing information technology. *Ivey Business Journal*, 66(3), 62–67. <https://iveybusinessjournal.com/publication/rapid-growth-firms-the-challenge-of-managing-information-technology/>

Saunders, M., Lewis, P. and Thornhill, A. (2016) Research Methods for Business Students. 7th Edition, Pearson, Harlow.

Seo, M. G., & Creed, W. E. D. (2002). Institutional contradictions, praxis, and institutional change: A dialectical perspective. *Academy of Management Review*, 27(2), 222–247. <https://doi.org/10.2307/4134353>

Serrano, J., Faustino, J., Adriano, D., Pereira, R., & da Silva, M. M. (2021). An IT Service Management Literature Review: Challenges, Benefits, Opportunities and Implementation Practices. *Information*, 12(3), 111. <https://doi.org/10.3390/info12030111>

Simon, H. A. (1991). Bounded rationality and organizational learning. *Organization Science*, 2(1), 125–134. <https://doi.org/10.1287/orsc.2.1.125>

Sloof, D., Walen, T., & Lentfert, S. (2018). *IT & Risk Management The importance of the internal organization of scale-ups in overcoming their growing pains*.  
<https://www.compact.nl/pdf/C-2018-1-Lentfert.pdf>

Stepanov, O. (2024, January 19). *Strategies to Mitigate Custom Software Development Risks*.

Computools. Accessed 02 May ,2025

<https://computools.com/strategies-to-mitigate-custom-software-development-risks/>

Szathmári, E., Varga, Z., Molnár, A., Németh, G., Szabó, Z. P., & Kiss, O. E. (2024). Why do startups fail? A core competency deficit model. *Frontiers in Psychology*, 15. <https://doi.org/10.3389/fpsyg.2024.1299135>

Tamzid, A. (2025, February 19). IT Governance and Strategy for Startups: Best Practices for 2025. *Bright Defense*. Accessed May 01, 2025: <https://www.brightdefense.com/resources/it-governance-and-strategy-for-startups/>

Uludağ, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., & Matthes, F. (2022). Revealing the state of the art of large-scale agile development research: A systematic mapping study. *Journal of Systems and Software*, 194, 111473. <https://doi.org/10.1016/j.jss.2022.111473>

Van de Ven, A. H., & Poole, M. S. (1995). Explaining development and change in organizations. *Academy of Management Review*, 20(3), 510–540. <https://doi.org/10.5465/amr.1995.9508080329>

Wadsworth, A. (2025, March 11). Outdated Tech, Hidden Costs: Why Legacy Systems Undermine ROI And Innovation. *CloudTweaks*. Accessed April 16, 2025  
<https://cloudtweaks.com/2025/03/outdated-tech-hidden-costs-why-legacy-systems-undermine-roi-and-innovation/>

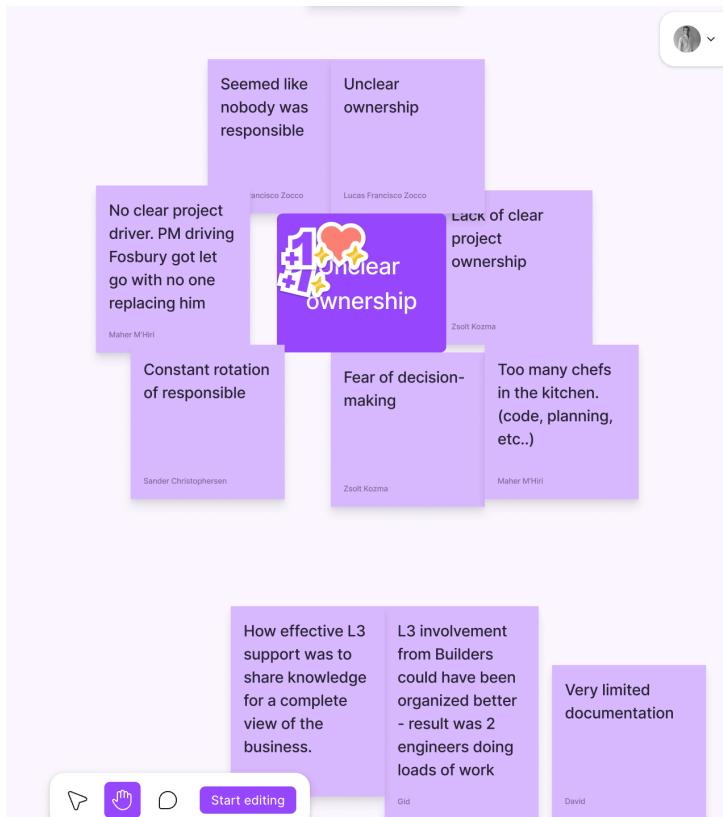
- Weill, P., & Ross, J. W. (2004). *IT Governance How Top Performers Manage IT Decision Rights for Superior Results*. Boston: Harvard Business Review Press.
- Yin, R. K. (2018). Case Study Research and Applications: Design and Methods (6th ed.). Thousand Oaks, CA: Sage.

## Appendices

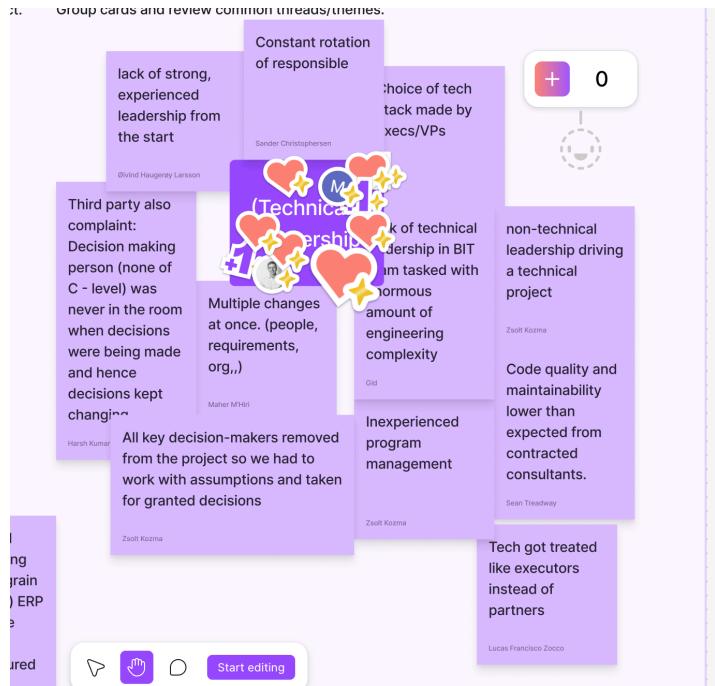
### Appendix 1

Example screenshots from Post Mortem Internal Documents only some are posted to provide clear overview:

**Screenshot 1:**



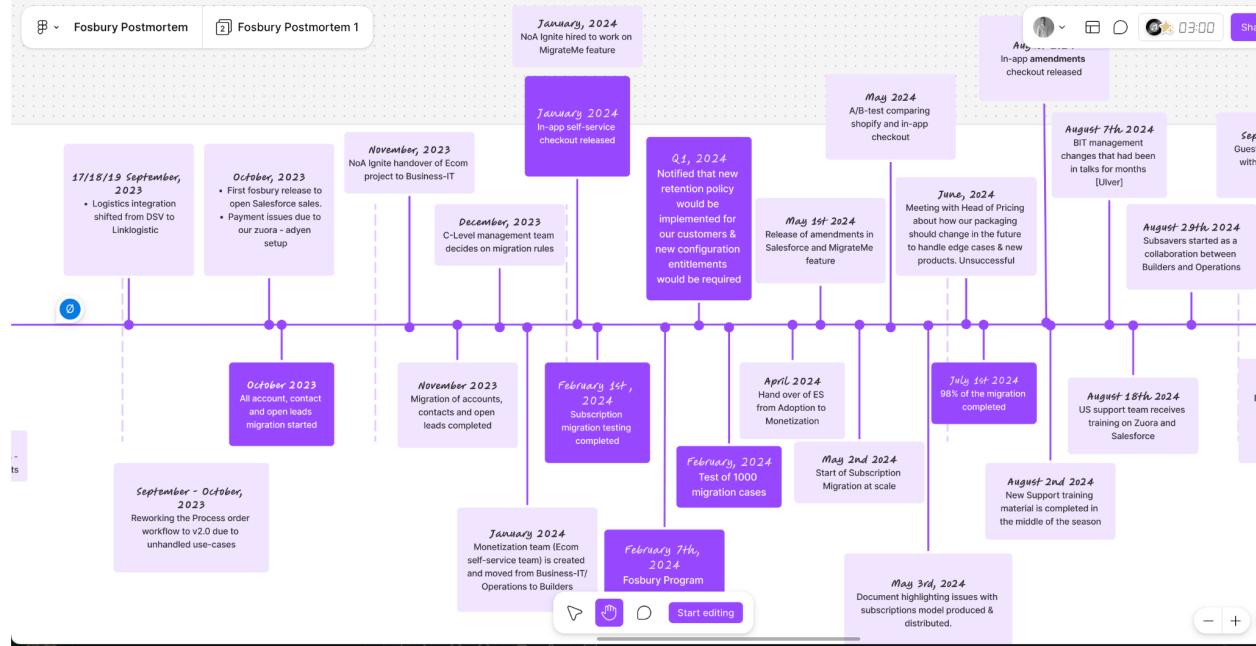
**Screenshot 2:**



## Screenshot 3 & 4:

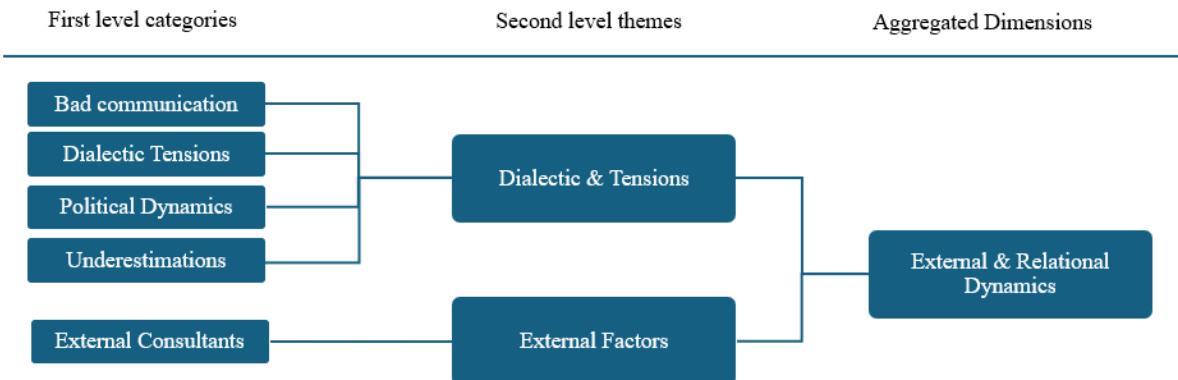
Fosbury rough timeline from my google calendar (not all fosbury events are here) 1 Nov 2022 John test engineer 11 Nov Touch base with Noa ignite Sprints with sub factory 16 Nov ecom deployment options with Daniel fly 28 Nov Zola integration design with Alex (sub F) 1st Dec zuora DSV product sync with sub f 13 Dec Noa ignite ecomm middle layer planning Veo running other projects in parallel like DATO cms to Easy translate , analytics launch etc 23 Jan 2023 meeting with sub f subscription master planning prep 25th Jan quote task refinement 27 Jan programmatically send quote to zuora CPQ 27 Jan goodbyes John test engineer 31 Jan Kamo Fosbury with Jerry 2 Mar review requirements for com 24 Feb 1st ecom frontend deployment GitHub actions 27 Feb system dev cluster 1 march smoke test user journey new sales 2 march quote refinement 6 march noa ecomm middle layer kick off 9 march Documentation 14 march ecomm middle layer discussions with noa 16 march zuora DSV Zola integration test Zapier sales force 17 march internal discussion to improve working with zuora and sub F 27 march ES for amendment and features with Maher Manfred and Timo priorities 29 march ecom with sub f Dinner with Sub f 31 march demo with sub f 4 April migration code review and beers 5 April demo with sub F 14 April demo with Sub f 18 April sub f pointer demo amendment 18 April Fosbury collaborative demo 21 April activation dates and pending orders with zuora and sub F 24 April workflows with manfred 25 April amendments <b>First release planned somewhere in May 2023, was meant for new sales only via salesforce checkout</b> This is where scope creep happened because we had discovered amendment is expected to work after new sale 2nd may ecom deployment prod ready discussion with cloud team 4th may demo with sub f end to end zuora and ecom orders	9th may tonur ecom frontend deployment 16 may veo test run workflow 11 with sub f 1st jun Christopher and Isaac onboarding 2 jun test Fosbury order in visibility 7 jun intro to linklog We were done with DSV integration for Fosbury and then it was discarded because we chose to change to link logistics 12 jun webshipper 12 jun workflow v2 design with sub f 21 jun ecom handover from noa 26 jun workflow v2 review 27 jun requirement gathering for link log integration with veo logistics 28 jun freemium rules were changing 4 July vbcombe handover 5 July linklog project handover to Marko (noa?) Db discussion about vbcombe db 7 July linklog tech design 13 July review workflow 2.0 tech design (edited) 10 August veo ecom and salesforce logic with sub f 11 August plan and billing load time 14 August David PM joins 15 August meet Ditte 15 August TDD review with sub f and zuora comments review 16 August KT on webshipper / linklog to Veo logistics 22 August family and club+ products planning 22 August linklog testing 28 August read documentation from sub F 31 August linklog prod deployment coordination Ecom documentation and solution handover from noa 1st sept plan release workflow 2.0 5 sept linklog to BC with Poul 7 sept test linklog in production 11 sept Sorin joins as release and test manager 17 sept go live linklog 20 sept linklog feedback with Veo logistics 25 sept migration Timeboxed 26 sept hypercare planning for Oct Fosbury release Migration alignment and planning with Ditte 5 Oct understanding payment issues during beta release 10 Oct go live with bridge call with sub f 12 Oct payment optimisation / hot fix Payment flow optimisation continues 23 Oct amendment flow demo/testing 26 Oct Veo invoices aging issues Overdue invoices Zuora university intro session 1 Nov Sales ops issues Veo open invoice issues continues
---	---

## Screenshot 5:

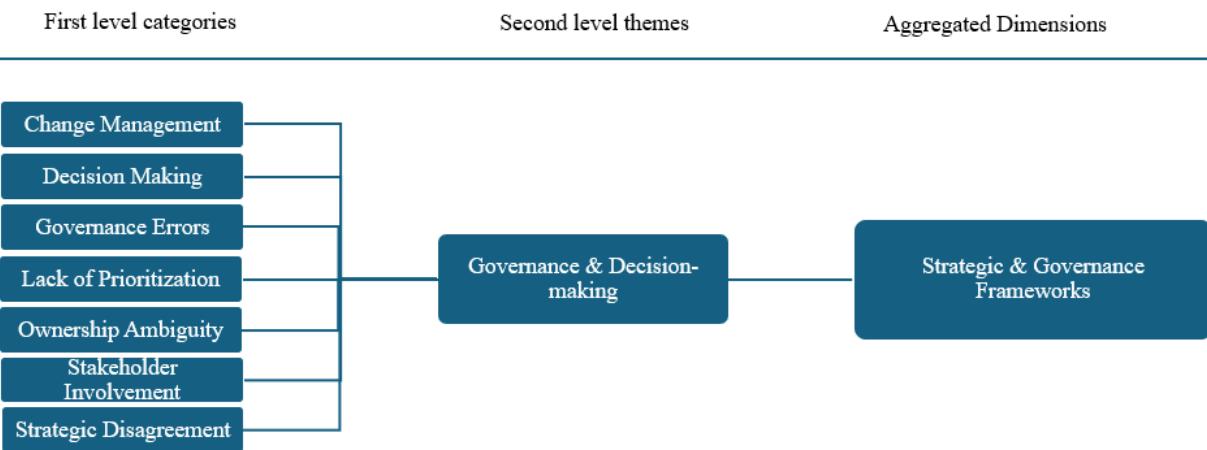


## Appendix 2

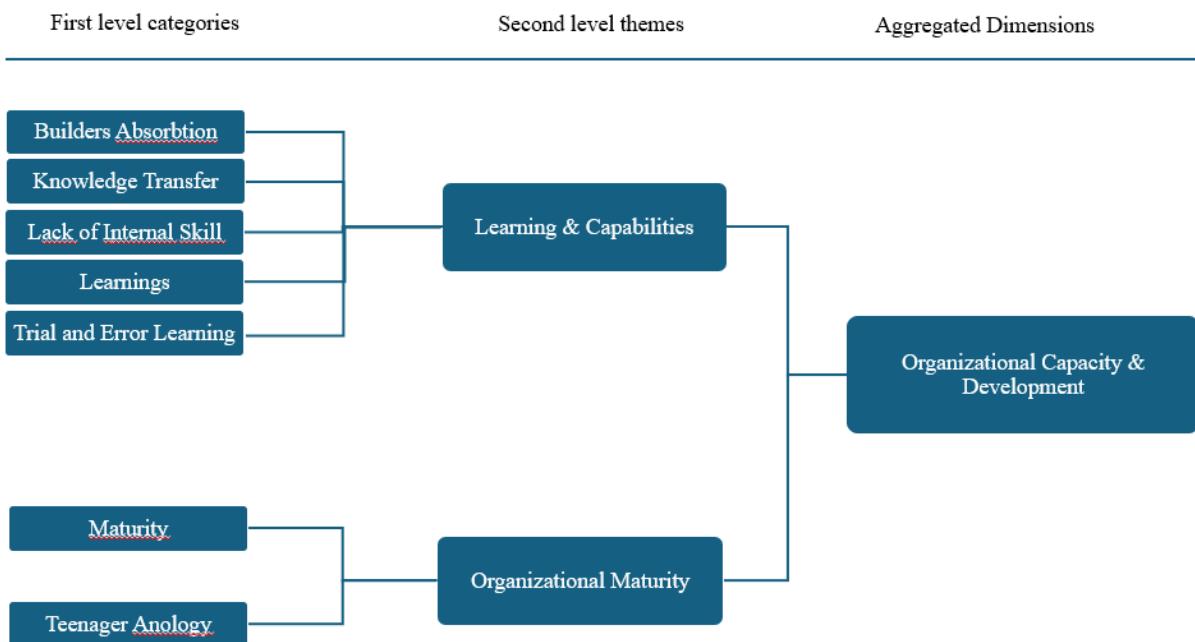
### Gioia Method



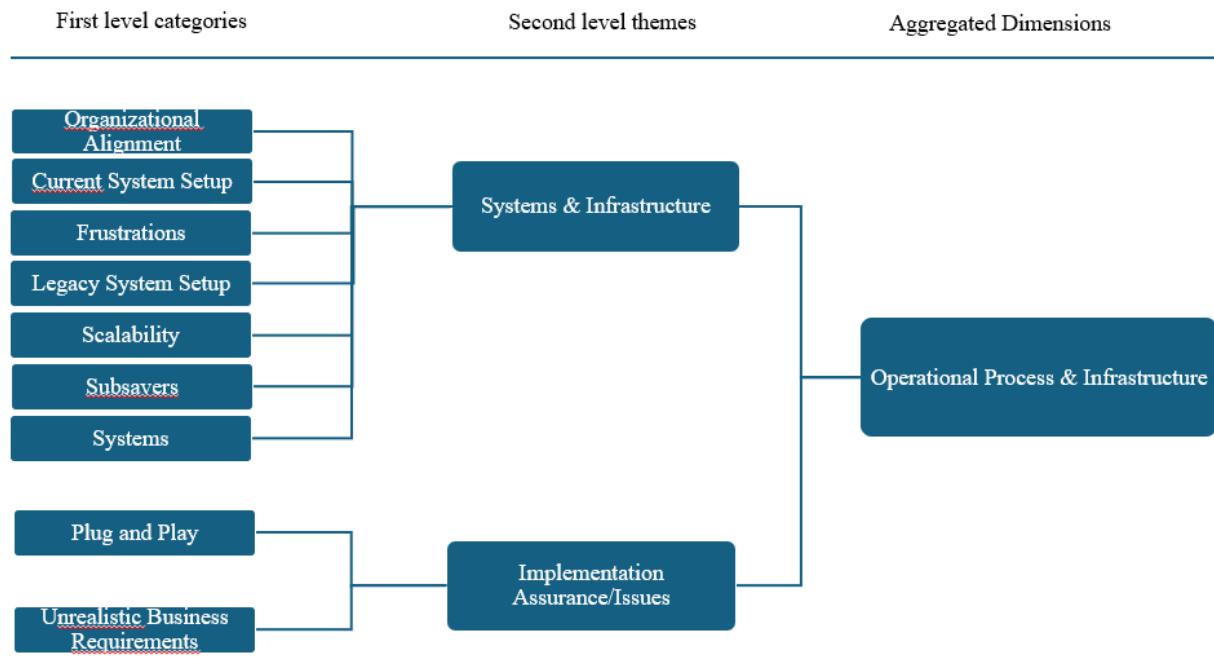
## 2.1 Figure 6



**2.2 Figure 7**



**2.3 Figure 8**



**2.4 Figure 9**

## Appendix 3

Interview Guides:

### Interview w. Sam System Integration Engineer

<b>Purpose with interview</b>	Get information on – Development, Challenges, & Decision-Making
<b>Timeframe and Recording</b>	Interviews are conducted either physically or online through Google Meets, and the timeframe is maximum 60 minutes.
<b>QUESTIONS</b>	

1. Can you briefly introduce yourself and describe your role in the Fosbury project?
2. At what stage did you join, and what were your key responsibilities?
3. From your perspective, how did the project evolve over time?
4. The post-mortem highlighted tight deadlines and scope creep. How did these factors influence early technical decisions?
5. What were the most critical trade-offs made in the technology stack and architecture?
6. The project involved migrating to a new platform. What were the biggest technical challenges in that process?
7. Were there any early warning signs that the team would struggle with scale or complexity?
8. The post-mortem mentions heavy reliance on external consultants. How did this impact development?
9. There were also concerns about architectural mistakes and tech debt. Can you elaborate on what went wrong?
10. How did the team handle unexpected issues like system failures or performance bottlenecks?
11. Were there any challenges in aligning with business requirements and shifting priorities?
12. Given the high pressure, how did the team balance technical quality with speed of delivery?

13. The post-mortem highlights a lack of strong leadership and ownership issues. How did this affect development teams?

14. Were decisions made collaboratively, or were they dictated from above?

15. How effective was the communication between engineers, management, and external consultants?

16. The team struggled with unclear requirements and fuzzy specifications. How did that impact development?

17. Did you feel the governance structures supported the team, or did they add unnecessary friction?

18. Looking back, what critical mistakes would you have avoided?

### **Interview w. Ulver Former IT Director**

<b>Purpose with interview</b>	Get information on – Technical Execution, Development Challenges & Lessons Learned
<b>Timeframe and Recording</b>	Interviews are conducted either physically or online through Google Meets, and the timeframe is maximum 60 minutes.
<b>QUESTIONS</b>	
1. Can you describe your role as IT Director and your involvement in the Fosbury project?	
2. What were the overall strategic goals behind the launch of the project?	
3. How did these goals evolve as the project progressed?	

4. The post-mortem analysis mentions a lack of clear ownership and decision-making. How were the most important decisions handled?

5. Was the governance structure effective in balancing technical and business priorities?

6. How did management navigate the pressure from rapid deadlines and scope creep (expansion of the project's scope)?

7. Were there disagreements between technical teams and management about the project's direction?

8. The project was very dependent on external consultants. How did management view their role?

9. There were several changes in management and unclear ownership. How did that affect the team?

10. How did collaboration between teams and knowledge sharing across the organization work?

11. Were there any critical moments where management had to intervene to realign the project?

12. The post-mortem analysis mentions technical debt, system instability, and lack of testing. How did management react to this?

13. What unexpected risks arose during the project?

14. How did the company handle errors and setbacks in the critical phases of the project?

15. If you could go back, what would you have done differently from a governance perspective?

16. What process improvements could have made the project more successful?
17. What are the biggest learnings about scaling a complex IT project under tight deadlines?
18. How has Veo Technologies changed its approach to IT governance and project management since the Fosbury project?
19. What advice would you give to IT managers who have to handle large transformations like this?
20. How was Top Management involved in the decision-making processes for the Fosbury project? Were they primarily strategic decision-makers, or were they also involved in technical details?
21. What role did Top Management play in navigating the strategic challenges, such as scope creep and changing priorities? Were quick decisions made, or were there bureaucratic delays?
22. Did you experience instances where technical recommendations from the development team were overridden by business decisions from Top Management? How did that affect the project's progress and technical solutions?
23. What governance mechanisms were in place to ensure that Top Management had insight into the project's status and challenges? Were there regular meetings, reports, or escalation processes, and did they function as intended?

### **Interview w. Øivind System Engineer**

<b>Purpose with interview</b>	Get information on – Technical Execution, Development Challenges & Change Management in Fosbury Project
<b>Timeframe and Recording</b>	Interviews are conducted either physically or online through Google Meets, and the timeframe is maximum 60 minutes.

## QUESTIONS

1. Can you introduce yourself and describe your role in the Fosbury project?
2. When did you join the project, and what were your key responsibilities?
3. How did your role evolve as the project progressed?
4. Did you feel you had clear expectations and direction from the beginning?
5. What were the main technical challenges you faced during the project?
6. The project involved migrating to a new platform—what were the biggest challenges from a developer's perspective?
7. Were there major technical constraints that made development difficult?
8. How was collaboration within the engineering team? Did senior developers and leads provide adequate support?
9. The post-mortem highlighted scope creep and shifting priorities—how did these changes affect development work?
10. How often did technical requirements or deadlines change, and how did the team handle it?
11. Were developers given enough time and resources to adjust to these changes, or was it more

reactive?
12. How were changes communicated? Were they clear and structured, or did they cause confusion?
13. Did you feel involved in decisions that impacted development, or were they mostly top-down? 14. How did leadership or project managers handle unexpected challenges?
15. The post-mortem noted issues with external consultants and knowledge gaps. Did you feel there was a reliance on outside help?
16. Were there enough opportunities for knowledge sharing and internal learning, or was everyone working in silos?
17. Did leadership provide clarity when the project faced difficulties, or was the team left to figure things out on their own?
18. What role did junior developers play in decision-making? Did you feel your input was valued?
19. Looking back, what would have made your role and development work smoother?
20. If you could change one key aspect of the project, what would it be?
21. How has your experience in Fosbury shaped your approach to working on large-scale projects?
22. What advice would you give to junior developers entering a fast-moving, high-pressure project?

### Interview w. Jesper Zuora Product Owner

<b>Purpose with interview</b>	Get information on – Late onboarding, fire-fighting, product ownership, coordination & lessons learned
<b>Timeframe and Recording</b>	Interviews are conducted either physically or online through Google Meets, and the timeframe is maximum 60 minutes.
<b>QUESTIONS</b>	
1. Can you briefly introduce yourself and your current role in relation to Zuora at Veo?	
2. When did you join the project, and what were your key responsibilities?	
3. What were your initial priorities when you joined? Did you receive a proper handover, or did you have to figure things out yourself?	
4. You joined during a phase described as "fires everywhere." What were some of the most critical problems you encountered?	
5. How did you go about understanding the system landscape and existing Zuora setup while things were moving so fast?	
6. Were there any quick wins or immediate changes you were able to implement that helped stabilize the situation?	
7. Did you feel that ownership of the Zuora part of the project was clear when you came in?	
8. How did you coordinate with the developers and engineering teams working on Zuora integrations?	

9. Were there any conflicts between business requirements and technical limitations in the Zuora domain?

10. The post-mortem mentions architectural issues and tech debt—how did those manifest in Zuora-related work?

11. How involved were you in decision-making about changes to workflows, pricing, or subscription logic in Zuora?

12. What governance or decision-making structures were in place around Zuora as a system?

13. Were consultants involved in the Zuora implementation? If so, how did that collaboration go from your side?

14. Did you feel the Zuora roadmap aligned with the overall Fosbury goals, or was it being handled as a side-track?

15. What constraints did you face in terms of system limitations or integration with Salesforce, Linklog, etc.?

16. Looking back, what would you say was the biggest obstacle to a smoother Zuora migration?

17. What would you have changed if you had joined earlier or had more control over the setup?

18. What do you think worked surprisingly well, even under pressure?

19. How would you approach a similar subscription system migration in the future based on your experience here?

20. What advice would you give to someone stepping into a chaotic project mid-way as a product manager?

### Interview w. Alexander CTO

<b>Purpose with interview</b>	Get information on –Business IT Absorption, fire-fighting, Governance coordination & lessons learned
<b>Timeframe and Recording</b>	Interviews are conducted either physically or online through Google Meets, and the timeframe is maximum 60 minutes.
<b>QUESTIONS</b>	
1. Can you briefly describe your role and when you joined Veo as CTO?	
2. At what point did you get involved in the Fosbury project, and what was your initial assessment of the situation?	
3. What were your initial priorities when you joined? Did you receive a proper handover, or did you have to figure things out yourself?	
4. What were the most urgent challenges you identified when you took charge of the Fosbury project?	
5. What were your first actions or decisions in trying to turn the project around?	
6. Did you bring in any new frameworks or governance practices to help stabilize or redirect the project?	
7. How did you approach stakeholder involvement differently compared to the earlier phases?	

8. Was there a clear shift in ownership or responsibility for critical decisions?
9. How would you describe your philosophy or approach to IT governance in high-growth environments like Veo?
10. In your view, what cultural or structural changes were necessary to make the Fosbury project successful?
11. How did you ensure internal knowledge retention and reduce dependency on external consultants?
12. Did you encounter resistance to change from within the organization? How did you address it?
13. Looking back, what do you think were the key success factors that helped 'rescue' the Fosbury project?
14. Are there any decisions you made during the process that you would reconsider today?
15. If you were to advise a CTO at another fast-growing tech company facing a similar transformation project, what would be your top 2-3 lessons?
16. Would you describe the transformation as a move from "startup chaos" to "mature governance"? What were the indicators of maturity you aimed for?
17. Do you think Fosbury is an inevitable phase for scale-ups? Could it have been avoided with a different early approach?

#### **Interview w. Timo External Consultant at Subscription Factory**

<b>Purpose with interview</b>	Get information on –Learn about external perspective, Initial project, Governance coordination & lessons learned
-------------------------------	--

<b>Timeframe and Recording</b>	Interviews are conducted either physically or online through Google Meets, and the timeframe is maximum 60 minutes.
<b>QUESTIONS</b>	
1. How did Subscription Factory first get involved in the Fosbury project at Veo?	
2. What was the original scope or mandate as you understood it at the outset?	
3. Did Subscription Factory use a standard implementation playbook, or was it tailored to Veo's needs?	
4. Were there any architectural decisions (e.g., business logic in Zuora) you advised on, but Veo chose otherwise?	
5. How did you approach integrating Zuora and Salesforce in this case? Was it plug-and-play, or more layered?	
6. How would you describe the collaboration with Veo — in particular, how responsibilities were distributed between internal stakeholders and your team?	
7. Were there any aspects of the project where you felt additional involvement from Veo (e.g., technical stakeholders or leadership) was needed earlier?	
8. In your view, how well prepared was Veo in terms of internal knowledge about Zuora, Salesforce, or the complexity of subscription-based architecture?	
9. Did the technical or organizational setup at Veo pose any particular challenges for the implementation?	

- |  |
|--|
| 10. Did the project change in size or direction compared to what was originally agreed upon?   |
| 11. Looking back, do you feel the project unfolded in line with initial expectations?  |
| 12. If you could go back to the start, is there anything you would have recommended differently - whether to Veo or internally within your own team? |
| 13. Compared to similar projects you've been involved in, what stood out about the Fosbury case?   |
| 14. Based on this experience, what advice would you give to a fast-growing startup planning a similar transformation?                                |

## Appendix 4

### Interview Transcripts

#### Interview 1

Sam / Thomas - Fosbury Project Thesis Interview - 2025/03/19 10:01 CET - Transcript

##### Attendees

Thomas Agerbæk Ruby, Sam

##### Transcript

**Thomas Agerbæk Ruby:** ...open for business up to. Yes. But let's just start out with an introduction. If you could briefly introduce yourself, yeah, and what your role is now and what it was in the Fosbury project.

**Sam:** My name is Sam, and I joined four years ago. And then I joined Veo as system integration engineer, then became senior system integration engineer, and right now I am a senior software engineer. And I was part of the beginning of the journey. I was not fully involved in the

beginning, but eventually became fully involved because I was also handling the old systems, so primarily working... yeah.

**Thomas Agerbæk Ruby:** Yeah, that's exactly what I wanted to know: at what stage did you join in the timeline and how much do you know from the beginning of the project?

**Sam:** So, when I said that I wasn't involved fully, I meant the decision-making and more the meetings which were about concrete details. I wasn't involved in every one of them from the beginning, but I was aware of it from the very beginning, from the kickoff itself, that we have this project where we are switching from old systems to new system for scalability.

**Thomas Agerbæk Ruby:** And in the beginning? How was the project presented in terms of timeline and resources and everything? Was it like, "We're just going to do this quickly; it's going to take six months"?

**Sam:** So, that's why, kind of, this is something that was not presented/communicated very well. We had the first deadline or timeline that from the kickoff, somewhere here, we said that we will release this on the 1st of May 2023. So at least there was a six-month period.

**Thomas Agerbæk Ruby:** Yeah, yeah, yeah. And from your perspective, how did the project evolve over time from this thing to what it became? Yeah.

**Sam:** So, we had an idea of what we were doing. So, we had a bunch of systems. I'm just going to call out the systems. So, we had Shopify; we had Pipedrive. And this is what we use for placing sales orders. This is what we use for the CRM, so getting leads into the system so that our CSS can start calling them and then ask customers to place orders, which would be an order in Shopify. And then behind Shopify, there was a system called Bold, which was the subscription management system. And we were being a company selling hardware, but the hardware needs software – a subscription – to perform, to be active. So, the requirements [or "issues"] were... one of them was Bold was not scalable enough, and we had some more different problems that we had. So, from here we wanted to get to Zuora, which is a subscription management system, and from here we wanted to Salesforce. In simple terms, there were a little bit more systems involved; this would overwhelm the whole project. But with the expected release date, I think the idea was to allow all the new sales starting 1st of May into Zuora.

**Thomas Agerbæk Ruby:** So, the unexpected release date... only new sales were the proper focus?

**Sam:** But by the time we were getting close to the deadline, new sale was also not very stable, but we also had a requirement of... right after a new sale, usually our customers, they also want to add more to their subscriptions. So, amendment, yeah, was also needed. So, this was the first scope creep. And so, there was another important system that we had: we were using Stripe as a

payment gateway, and here we wanted to go to a scalability [solution] or also... Adyen gives you more options to integrate with more payment providers.

**Thomas Agerbæk Ruby:** Yes, scale and all this entire project was because we realized, we... we started with these systems and we need something for the future, and we keep on growing and having more customers, so we can more efficiently manage...

**Sam:** Exactly.

**Thomas Agerbæk Ruby:** ...our payments, everything.

**Sam:** We did not want to be blocked by these systems in other scalabilities because there was a little bit of manual effort going into these systems as well, and as we grow our customer base, the manual effort would also grow and then other system limitations would also bother the scale. So, that's why the idea was to move on to better systems.

**Thomas Agerbæk Ruby:** It was also a thing called the technology stack, yeah. How did you know... how we came to the conclusion that we need these systems and not others, and how do we actually use them? Yeah.

**Sam:** So, there was some research done, and then again, most of it was decided by people up in the chain. And then I think some of it was by talking to other people who had experience with different systems. And I remember at... Zuora, and then they also recommended Zuora, and that's how we took Zuora. But Salesforce was one of the standard in market for CRM for managing a 360° view of a customer. And as... I have no idea why we chose Adyen, what was the theory behind it or the research behind it.

**Thomas Agerbæk Ruby:** And how did the team look back then, who made decisions, who helped in starting the project? Yeah.

**Sam:** So, there was the systems team who was responsible for managing almost all the systems in Veo, or at least a majority of it. And I was part of the team, and we were responsible for the BI [Business Intelligence?] systems, and then we were responsible for making this change. And that team, in the beginning, it was... our managers, IT director... Is it okay to call out the names?

**Thomas Agerbæk Ruby:** Yes, I will anonymize [you].

**Sam:** So, [Name Redacted] was IT director, and then [Name Redacted] is a senior software integration engineer. Then [Name Redacted] was also an IT director and a project manager to manage the project. And then there was Christian Holm who was managing the whole thing. Yeah. And there were other components. So, these systems were eventually supposed to interact with Zuora as a product, which is their product. And then we had some people in product teams

as well. So, it was not that big of a team. It was a fairly small team, and it was very optimistic to hit the expected release date. Yeah.

**Thomas Agerbæk Ruby:** And you mentioned that in the beginning, a lot of the calls were made by higher-ups, and did these higher-ups have any knowledge of what it took to do this new integration of new systems? And...

**Sam:** Yeah, I'm not so sure about that. And I think ideally it's a good idea to get an idea of how big the project should be or to take that kind of information first to decide on the timeline. But it was the other way around: our season would probably start somewhere from May, [so the project] should be [done by then]. And I think another thing was that we would probably use everything out of the box, and it would be just plug and play. That was the mentality of the IT director. And that's where the knowledge of a senior integration engineer, who had been dealing with integrations and who... I'm talking about me now. I knew about how customized a solution is, and that information was kind of not even taken into consideration.

**Thomas Agerbæk Ruby:** Your opinion? You're not being asked of your opinions if there... you need to make like tradeoffs or in the technology stack, or is this possible, or do we need to focus on other systems, or will this be compatible? Yeah.

**Sam:** So, systems wise, I think it's not such a bad choice. It's not something that was way away from where we wanted to go. But it was more about how much effort it will take in implementing these integrations. And I think it was... so we did, in fact, me and the director, [discuss] on what kind of integration or what kind of requirements we have. So, the requirements were kind of clear, but the effort to implement those was not.

**Thomas Agerbæk Ruby:** And yes, you talked about the project involved migrating to new platforms, and what would you say were the biggest technical challenges? The process of migrating a lot of customers that you have?

**Sam:** Yes, so that's also a very overall project. The first thing is to have all the new customers that are onboarding, so new sales, and making changes to the new customer subscribers who are expected to be in the new system. So, according to me, I think once we have a stable future to go to, only then we should be thinking about migrating our existing base to the new platform. I'm not saying that we cannot start preparing for the migration work, but the migration should be executed once we are very sure about where we are taking our customers. So, one of the challenges was... it kind of was that our integrations were not stable enough. So, we were still handling a lot of cases case by case when a customer raises a request, and then that's what caused it to be not that stable enough. So, there was a lot of manual effort; this boiled [over]. And then when we started migrating people, that manual effort just bombarded [us]. And one of the important things I think I would like to highlight, which was supposed to be considered

before starting the project or when we knew that we were going for these systems, was the knowledge about these... schemes [or "systems"] of some stuff. So, there were no trainings, and trainings could have helped us gain more knowledge about the system, which were kind of gained on the job. So, while having it all in production, we would continue to learn about these systems.

**Thomas Agerbæk Ruby:** And one thing that is also mentioned in the post-mortem is, at one point, the project or we began to rely on external consultants, and how did this decision impact development of the project? Yeah.

**Sam:** So, that's a very good question. So, talking about how many resources we had, I said earlier that we had very low resources, but I think the mentality of the head, the director, was to give a lot of work to consultants so that if there are bigger pieces of work, the consultant who already have knowledge about these systems [can do them], to enable those consultants to do all the work. But the key point in doing that is that instead of knowledge coming inside Veo, it was going outside Veo. So, we were training external consultants with Veo knowledge so that we can start building things for how we want it, instead of the other way around. So, instead of Veo leading the project and gaining information from consultants, and then whatever is a doable piece of work that can be done by a consultant, even though Veo is driving it, could have been done in that way. So, this is what happened. So, we had to explain these requirements to external consultants, and then that takes time and then iterations to get the project moving forward.

**Thomas Agerbæk Ruby:** But of course, there's... what I saw in the post-mortem was, somebody mentioned that it was very much a waterfall approach, where you guys are saying, "We would like this," and then a while later they come with something, and there's not a lot in between you stating the requirements and they come up with a solution where you can communicate and explain...

**Sam:** No, no, it was supposed to go the other way around. "Is it supposed to look like this?" Is that also something you feel was...?

**Thomas Agerbæk Ruby:** So, it was more like the communication was very important in this project, yeah. Then us being in the right people being in the right rooms to talk about requirements and then agree on a set of requirements and accordingly proceed.

**Sam:** But that was also kind of... So, again, I might be sounding like that it was a complete failure, the whole project. It kind of worked, but we also had a lot of learning on the way, and eventually we did make it happen.

**Thomas Agerbæk Ruby:** Yeah, that is also what I want to... when I'm looking at it in an academic way, I want to be as objective as possible. So, also when you talk with people, and the way we've been talking about the project is, "It's been so difficult and it's been a failure, and

everybody's been fired and all this stuff." But also look at what actually went well, and it actually got done in the end. And of course, I think that's also one of my focuses in my project here is that when you are a company that is in high growth, high scaling, at one point, I think every company will get to a point, will have their own project where we're moving on and we're making all the wrong decisions, but learning so much and getting a lot of value. And if you actually succeed [with] this project, then you will have made a huge leap for further success in the future because of all the learnings and because of actually getting the new systems and getting ready for even more growth. Yeah. So, when you look at it like company-wise, for me it's more like of a teenager period, where you're getting all the new features of the body. It's ugly, it's not very mature and everything, but once that period is over, you will grow into something mature.

**Sam:** Yeah, that's very well said. I think that's how Veo has been on this. As a young kid, and then also talking to consultants, and then also listening to their viewpoint. So, some of the things that they were talking about is that we did not have the right people in the right room who actually make decisions, and we had to then iterate on those subjects more often, for example. So, what was the problem was that we relied on experts who didn't have the knowledge that we have, which... they would have trouble coming out with the solutions that we actually needed because we were spending all the time telling them, "This is what we need," and not like learning yourself, "How could we do it?" We're just saying, "We need this," and then there is this black box of them... "We will come up with something." And then they give you a product that you find out later this is probably not exactly going to work like this, and then there's all this iteration going back and forth with these external consultants, and you just spend so much energy and time on trying to manage these people.

**Thomas Agerbæk Ruby:** Yeah. That you're paying to work. Yeah.

**Sam:** So, it's not only that. So, again, there's a saying in Hindi that you cannot clap with just one hand; that you need both hands to clap. So, it was not just Veo setting some expectations from external parties. It was also the other way around that external parties had some expectations from Veo as well: that you will give us clear decisions. So, they come up with options. "These are the options which are standard options that you can adopt to." And then we would have to choose one. But we wanted those external parties to choose an option for us by understanding our requirements. And this is where we had to go back and forth quite a lot because the standard options that were presented by the external parties were not a fit. We need a customized solution. So, that's why a customized solution was built especially for Veo after a lot of iterations on requirements. And then we actually had our first release somewhere in October with, let's say, let's call it version one of [the] workflow, which of course, it was working for some of the happy cases, but some of the cases were not working as expected. And that's where we had a window where we had released, but then we had stopped all the new sales again in the new systems, and then went back to the third party which was helping with Zuora workflows, and then they

understood our requirements even much better, and then they came up with the version two. I think version two is what is running right now, with some minor versions on top of it. But yes, so there was this back and forth on getting the requirements right. And so, also, yeah, one of the big things was that there was not a single person or a team of people who were supposed to lead the project in the right direction. So, we were having requirements from product side, we were having requirements from sales side, or from marketing side, that "This is how we would like it to work." And we could not build for all of them. And yet, we did build for all of them, but not making everyone happy. And that's okay, I guess. And this in the post-mortem is also highlighted.

**Thomas Agerbæk Ruby:** A lack of exactly strong leadership or somebody who takes the leadership and takes all these opinions and decides what to do. Ownership issues, for example, of the entire project. So, when all these different opinions came and somebody needed to make a decision, how was decision... was that made from above or was it individually in teams? So, could you try to work it out or...?

**Sam:** Yeah, it depends on how... Big decisions were, of course, made by people up in the chain, and the smaller decisions were still considered within the team, by the team as well.

**Thomas Agerbæk Ruby:** But would you say there was a collaboration issue with all these different systems? Everybody had opinions of what they should be able to do. Also...

**Sam:** So, one was collaboration issues, and another was the big people having the right skill set for running this project rightly. So, the leadership, I believe, did not have... at least the leadership who was running the project according to Christian. Christian had his own thoughts in terms of how to proceed and had his own experience, but it was not matching what the scale that Veo was talking about. So, we wanted to scale to a better system, but we did not want to lose quality, but we ended up losing on quality because of lack of skills by the leadership.

**Thomas Agerbæk Ruby:** Have you ever talked about IT governance or how decisions are made in a company about how we should run systems and everything like that? Have you thought about that?

**Sam:** Yeah, there were talks about that. So, at least not me directly, because I was more of how to integrate systems, but security was a topic in general: how do we ensure that there is security around these systems so that we don't have any attacks, and that does come under IT as a department, or IT was a department back then. So, there were talks, and [Name Redacted] was giving those talks as well. And yeah, so [Name Redacted] had a lot on his plate, but skills in some of the areas but not in all of them, but he tried to, I think, lead on those areas as well where he did not have expertise [in] building an architecture of how the integrations would look like. So, how... part of the design was, I think, in his mind it was a responsibility given to third parties.

Yeah, but there was no followup or a validation on how they are doing. There was no clear indication of that.

**Thomas Agerbæk Ruby:** So, if you were to start the project today, what would you have done differently in that aspect?

**Sam:** Yeah, I would start with knowledge and trainings about all these systems. And then, first, within Veo forming a team which would be called the architecture team or something, who would take decisions in terms of how the architecture should look like, even if they don't have the knowledge of these new systems. So, they will get those knowledge by trainings or by talking to consultants, and I would keep all this knowledge in Veo. So, also hire people, hire experts in new systems. So, for example, at this beginning... we got [Name Redacted] product managers here. Yeah, 2024 somewhere. Yeah. And we should have had [them] way in the beginning. And also [Name Redacted], who was the product manager of Salesforce, should have been hired before. So, I would have knowledge, more people, have a team which would take this decision, which would become a point of contact between the third parties and internal Veo, and also some of the decisions where C-level people are required, so have them in the room with third parties. That was one of the complaints that we had that was probably [related to the] project that was... the COO did it. Also, the requirements were bounded by our financial decisions, how we want to do finance. For example, Veo has a preference of... because Veo is also a growing company, so money in the bank matters for Veo. So, when a sale happens, then we would like the subscription part of the money upfront in [the] bank, which is different from the hardware part, which is supposed to happen when the shipment happens. So, that had a lot of friction on how the interventions were designed. But the CFO was never involved in any of the meetings so that the consultants could explain it to the CFO that this is how the standard thing works. Then you could get an immediate opinion of what is right for Veo instead of working with it and then get some feedback later on.

**Thomas Agerbæk Ruby:** Yeah. Okay. Yeah. Exactly. Okay.

**Sam:** So, it's like middleman convincing CFO, but then CFO saying that no, no, yeah, not this. So, then middleman getting convinced that okay, not this, then telling that to third party again, and then third party is raising something else again, back and forth. So, in my approach, also, I'm kind of pointing that out, but all of these decisions should be taken in... kind of moving forward. Yeah, we [should] not [go] back and forth.

**Thomas Agerbæk Ruby:** But I think there's a question here: What advice would you give a development team facing similar high pressure scaling challenges?

**Sam:** Yeah, I would say you stand on that we would like our requirements to be clear before making implementations, because sometimes what happens, you just end up developing

something which is discarded because requirements were not that clear. And so, these development teams, for example, in our case was also third parties' development teams who just went with a functional requirement with an ambiguous understanding of it, and then developed the whole thing. Requirements changed, then they kept enhancing on top of it instead of starting a fresh development, which causes complexity in your code as well.

**Thomas Agerbæk Ruby:** Okay. Is there anything you feel like we haven't been talking about yet that needs to be addressed?

**Sam:** Yeah, I think we're not that far, but I think... yeah, yeah. I think we're going in the direction of what all happened and then what all could have been done better. And we should have also realized this to be a big deal, a big project, because Veo was much more focused on the core product itself, which is amazing, of course, but how to get new people onboarded, new potential customers, was not given that much of priority, but it is now because of these reflexes that have happened. We don't have... so whatever is a development work is now owned by engineering. It's treated as part of the product, compared to how it was before, which was just operations. With operations, but then overall I think we had some setbacks. So, another advice is that don't start with a deadline which is very unrealistic, because you will end up spending a lot of bad time in here or you will lose on your design approaches because you will end up removing or scoping down the project.

**Thomas Agerbæk Ruby:** Yeah.

**Sam:** ...to hit the deadline. And then because of that you will end up having some constraints in your design which will kind of contradict the whole point of us scaling up. We would rather then struggle in scaling up.

**Thomas Agerbæk Ruby:** That is true.

**Sam:** And because of that, you will end up spending time which is maybe three or four times more than it would have been if we were taking a very flexible timeline. Yeah. So, this whole [project] could have been finished in one and a half year if we would go with one and a half year being the actual deadline. We started with six months, then we expanded it to another few six months, then another, and then yes, it kept expanding.

**Thomas Agerbæk Ruby:** Is it officially done, do you know that...?

**Sam:** What do you mean, the entire migration for the project? I have... I think it is 99% definitely done. I think there are still some sales happening in Shopify, but otherwise, most of it is actually over. So, now the focus is on doing the implementation in the right way compared to our stuff.

**Thomas Agerbæk Ruby:** And during the entire project, your team definitely built understaffed, right? Yes. Yeah.

**Sam:** So, that was also why you were talking about experts from the beginning. And you can also see, the team that started is completely dissolved and turned into something completely else at the end of the project. So, if you go to... what was it, September 2024? Yeah. Everybody, like IT director and COO retired. Yeah. And the business IT team was like... I had moved on to a product team. My reason of moving on, one of the reasons was that it looked like that we want to give all this development work to external companies going forward as well. So, I did not see myself fitting in the team anymore. So, that's why, for my career choice, I moved out of a different team. But then the team was dissolved, but also the team was kind of taken, absorbed by the engineering team, the builders team, which is where I wanted to... or I went to. So, I moved to builders, and then the whole team moved to builders. And then the COO and Tech [Director] were not needed anymore in the company. So, that was also one of the learnings from Veo in general was, these projects cannot [be] done in four men; you should maybe do 100 men instead. Yes. Which also was... at the end of the old setup, there were so many fires, and when we tried to actually come out with the product and now we're using Zuora, now we're using Salesforce... [fires] kept happening until Veo realized we need to put all our manpower on this project until it's solved. Yeah. Because it became such a big burn, which also created a soft crisis, and all engineers should prioritize helping out wherever they could. Yeah. So, all that fire actually began due to us migrating our existing base, which was still 90% of the total base. So, 90% were still in the old system, with 10% in new. And this is just a rough number. We were still struggling to support these 10% in the new system with unstable products. And then instead of raising more manpower inside, we raised more liquid [resources?] in here, and that's why it caused the fire, and that's why [Veo] made [us] realize that we need more manpower. That's why the team was absorbed. Other teams started pitching in.

**Thomas Agerbæk Ruby:** Okay. Do we have anything to conclude on, something you feel like is missing out?

**Sam:** No, I think we have... So, overall, [it] was supposed to be six months. That was unrealistic, and I kind of knew it. I would say one and a half year would have been a good bet to come up with a very stable initial product and then having new sales and amendments in the new system, and then slowly learning from it and then migrating more. So, it could have taken the same amount of time, but it would have felt much more stable, much more scalable, which I guess instead we had a lot of learnings the wrong way around. Yeah. We made some customers unhappy on the way, and customers should be a priority, and it's hard to win back customers compared to giving a good experience to a customer from the beginning. So, it could have been the same amount of time, or even shorter a bit, but we would have not lost customers due to bad customer experience because of the things that we went through.

**Thomas Agerbæk Ruby:** Anything I'm also going to talk a little about it, but anything about the COO's involvement? I just know that there was a little bit of friction between that part and so...

**Sam:** So, COO was not in the company from the beginning. We had a different COO, and then he left. Then, somewhere, I think we're here in October, we had the new COO. And again, these big decisions were supposed to be taken by big people in a way, but they were being taken by the IT director. So, big people as in COO, CEO did not care much about the project until they saw fire. And I always say when I was in the team, so this Business IT systems team is like a backbone of the company. It was the backbone of the... So, the thing with the backbone is, if it doesn't hurt, we don't care about it. So, when it hurts, it's the only thing you [focus on]. And that's why it happened. Yeah.

**Thomas Agerbæk Ruby:** All right. I think I won't take more of your time. I hope this helps.

**Sam:** Yeah, definitely does. It definitely does. Let's see. [noise?] probably just entered. Extra cautious. I also want to check.

*Meeting ended after 00:43:24*

## Interview 2

**Ulver / Thomas Master's Thesis - 2025/03/20 10:05 CET - Transcript** Attendees: Thomas Agerbæk Ruby

### Transcript

**Thomas Agerbæk Ruby:** And Frost was out, right?

**Ulver:** When was that? I had just started around that time here. Who is that Frost?

**Thomas Agerbæk Ruby:** Well, that was him who was VP Engineering when I started.

**Ulver:** Okay. That was your...

**Ulver:** That was what Thomas Hagen took over, or what?

**Thomas Agerbæk Ruby:** What are you saying?

**Ulver:** Was that what Thomas Hagen took over? Well, then what the hell did he take over?

**Thomas Agerbæk Ruby:** Thomas Hagen?

**Ulver:** Thomas Hagen.

**Thomas Agerbæk Ruby:** Well, he... What the hell was Thomas Hagen at that time?

**Other speaker:** He was... He was...

**Thomas Agerbæk Ruby:** That's a good question what title Thomas had at that time.

**Ulver:** He was also some kind of VP, right?

**Thomas Agerbæk Ruby:** Yes. But where Frost (or Jacob Frost, as he's called), as one of my former colleagues from Phase One, whom Thomas Hagen also knows, and Jacob Frost was hired there before me, and Frost and Hagen were somewhat in agreement about the strategy back then and agreed that they would like to copy the model we had run up at Phase One, which was that IT took care of these e-shop systems and the customer-related tasks/systems, sales system, CRM systems, all of that. And product, they only do product. That means the code base itself and the product itself and the distribution of the product. And it was such a setup where there was a clear interface between them. And they wanted to copy that. So that's why I came down there. And so that was the idea. And then, as time has passed, there has been a lot of turnover. And especially the thing with Kruse arriving, and Frost is out, and Kruse has a different religion. It has kind of turned, what should I say, the whole... Where it all started, it has been turned upside down. And who does what? So there has been a lot of people change along the way, which has made it difficult, if you look at something like Kotter's eight steps and read... I can just send you that book if you haven't come across it.

**Ulver:** Thanks.

**Thomas Agerbæk Ruby:** Because it speaks to something like creating a coalition in the company, and that it is anchored in top management. And Henrik has never been very involved in it from the start. And it's easy enough to say afterwards, etc., and many times it also goes fine without.

**Ulver:** That's also what I kind of, when I talked to Sam today, also that thing about how it took a very long time before top management started to take an interest in the Fosbury project. It was almost only when there was a fire in the whole building. That they were like, "Oh yeah, now we'll just give it some attention and fix it."

**Thomas Aserbæk Ruby:** That's one of the points with those change management theories, Kotter's and others who do that kind of thing, that if it is not anchored in top management and they are involved, then it can be very difficult. Yes, then it can be difficult.

**Ulver:** We also talked about that thing about how there were a lot of different parties involved. There was top management, then there was you or us, and then there were external people hired to work on it. That communication was from the one to the middle to the top, and it was never brought together.

**Thomas Agerbæk Ruby:** That's also a classic, how much can we use external vs. internal. And what is the distribution? Where it is clear that we lacked internal knowledge about Zuora. We lacked technical knowledge about Zuora in the phase of the project where the solution was built.

We put our trust in Subscription Factory and did not have the resources to check up on it. Both on the architecture side – I was formally the architect on it, but was also simultaneously IT manager. That shows a bit the danger of mixing project and operations. It can be difficult to find time for it, and I didn't have Zuora experience either. So it was difficult to validate things from Subscription Factory's side. The other dimension in that is also test resources to test solutions. I come from a world where we have had a lot of test resources. But that has never really been a Veo discipline. That is, having a test department. And I hired one to start with, he quit himself, so I hired another one. Ditte didn't like him, so I was somewhat forced to fire him. Then I was in there. And then Ditte didn't want to help find someone new on that front. So there were also some things, when Ditte arrived, there were also obvious things that we didn't agree on. And you can discuss what is right and wrong. But it certainly doesn't help the project when you don't have common ground in the group regarding it.

**Ulver:** I myself came in there a little over a year ago now, and there was already a lot going on regarding the Fosbury project. When did you get into it? Can you remember when you started on it?

**Thomas Agerbæk Ruby:** I was involved from, I was hired before it even started. So the project didn't exist at all when I started there. When I started at Veo, and probably one of the reasons I was hired there, was that they had a Shopify solution together with some home-built subscription management, in Zola [likely referencing legacy systems]. And it was getting out of control, and there was a need to scale up and figure out how should we handle this? What do we do going forward? And the idea with Jacob Frost and Hagen and me, was to create a setup where product does product, and IT did the commercial part of the solution. And that was also set up from the start, before, I think it was before you arrived. There was Christoffer and the web team.

**Ulver:** Yes, meaning Kristoffer Houen.

**Thomas Agerbæk Ruby:** No, the ones who do [the] product. It's called something. The ones who do the self-service solution.

**Ulver:** Oh, Monetization is what it has become now.

**Thomas Agerbæk Ruby:** David and Christoffer and Isak were some of those I hired to start with. And they were IT, because the intention was that IT should create a micro frontend service for the application, where customers could service themselves and buy. Not service the product itself, but where they could service the e-shop part. And buy and manage their subscription, etc. Because product didn't want anything to do with that at that time. But then comes the shift from Jacob Frost over to Kruse, and then it slowly starts to move towards... No, product would like to own that. That part of it too. And we had a lot back and forth. And it ends up, eventually, with us moving David and Christoffer and company over to product. So, that whole journey. But when I started, there was no project called Fosbury at all. And Christian Houen wasn't there either. There was someone named Berte, and someone named Christian Jønsson, who are also no longer there, who wanted something new and were on board with us needing to create a better

platform for customer data. The big problem when I started was that they were asking for customer 360. No one could figure out how many subscription customers there were, how many subscriptions a customer had, etc. You couldn't talk about a customer. There were just a lot of Zuora logins, but how many of them had a subscription, how many of them were the same customer, etc., no one could answer that. It was completely impossible to figure out, with the way data was organized. So, I don't think much more than six months, three quarters of a year after I had been there, did we agree that it cannot continue like that. If you talk about change management, there were also some opponents. Primarily finance. Back then there was a CFO named David, and an accounting manager named Amel. And Amel, she was a very strong opponent of the project. Very strong. And Christian Houen fought very much with that, and eventually gave up. They didn't talk at all in the end.

**Ulver:** And why was she such a strong opponent?

**Thomas Agerbæk Ruby:** Yes. She had been involved in building the Shopify setup that Veo had when I started, where a lot of integrations had been made. And that was also fine. From a financial point of view, making a lot of automation that made everything – money in and out, bank reconciliation, and towards Shopify and data, etc. – that it flowed quite well back and forth from their perspective. But they don't care about answering these customer 360 questions that marketing and commercial, etc., need. Which clubs have how many subscriptions, how do we set up and how? Because it was already starting, slowly, to be that some had more than one camera. Today there are many who have many cameras, but when it started there was just one club with one camera. And after more and more came, the problem became bigger, the more there were who had more than one camera. Yes. So it quickly became clear that all this data needed to be organized so we could get an overview of what we have sold to our customers, and we needed to approach it [spandstræk - potentially 'systematically' or 'structured' but keeping original conversational style] so that subscriptions and money, revenue, the entire revenue stream around subscriptions and who are customers, payers, and users. It had to be structured in a more organized data model. And some more structured systems, because no one could find their way around it. You could potentially also talk to BI. Magnus Enk is not there anymore, right? No. But Bjarke is still there, right?

**Ulver:** He's quitting now.

**Thomas Agerbæk Ruby:** Quitting now?

**Ulver:** Yes, he's quitting at the end of the month.

**Thomas Agerbæk Ruby:** Yes. That's too soon. Bjarke is probably one of the few left, then, where you would be able to hear how much they struggled to answer questions about customers before Fosbury. He will certainly also think that there are some things about Fosbury that are more troublesome/different, but he can clearly also tell how difficult it was. We had, what were they called, Nora Connect, they did a lot of consulting work for Veo during the time I was there. They started by doing a project where they tried to stitch all this data together. Together with Bjarke. To be able to try to create a customer database. And try to answer some

of these questions about, how many users do we have? How many paying customers do we have?

**Ulver:** Yes, the overall strategic goal behind the Fosbury project was that it had become clear to Veo that we are growing, we are getting more and more customers. When we started with our old Bolt and Zuora setup and so on, Shopify, it was good up to a certain size, when we started to organize ourselves better.

**Thomas Agerbæk Ruby:** Total classic. You start somewhere with a scale-up there, and you hack together some Shopify and do some subscription in your product, etc. And then it runs, and then no one wants to touch it until it becomes so big that you can't fit into it anymore. And then the need arises to scale to a more enterprise-level setup.

**Ulver:** How did you then arrive at what the next step should be? I mean, how did you arrive at, for example, the Zuora and Salesforce setup there?

**Thomas Agerbæk Ruby:** There was a product selection process there. And I think it was around the time when Houen started. We had a project manager in IT, Oswald Blomqvist, who was super nice, but not very performance-oriented. So therefore, Michael, who was the then, shall we say, program sponsor... What's it called, you were involved in, Michael? [Likely addressing the researcher]. He was my boss before Ditte.

**Ulver:** No, he's not. He's on the board of Veo.

**Thomas Agerbæk Ruby:** And was my boss before Ditte. And he then hired Christian Houen. To get a program manager on the program. Who had a little more, shall we say, performance culture. And [when Christian] started, we had had many talks about, what should the solution be? And it pointed towards something, some subscription management platform, something that can handle subscriptions at scale, and some form of CRM system, customer [data]base system. And we looked at some different subscription solutions. Zuora, what are they called?

**Ulver:** There was also something B, something.

**Thomas Agerbæk Ruby:** Chargebee. And a third one too. There are, if you have access to some of the old Fosbury files, there are some presentations from back then where we looked at different products. And there were many of us who had opinions and views, but ended up agreeing that Zuora was probably the safest choice, which Veo would be able to scale with for the rest of their lifespan, almost. It's not because they are behind [not number] one, the really big ones. And on the sales side, Christian Houen, he comes from a Salesforce world. So therefore, Salesforce became kind of automatically resolved. The obvious choice to go with. And Zuora and Salesforce have native integrations, etc. Subscription Factory has also made a lot of solutions with Salesforce, Zuora together, etc. So it slowly became the most obvious solution. It wasn't something we had to fight about or discuss. The more we talked about it, the more obvious it became that those were the choices.

**Ulver:** When you kind of arrived at that, where [you] then talked about that if you were to do it differently today, you would perhaps get some internal knowledge from the start. Some internal knowledge about the systems, etc. Were there any considerations about that at all back then?

**Thomas Agerbæk Ruby:** Remember that we specifically talked a lot about what we *should* have done in hindsight. There was a will, and I can certainly see that Thomas Hagen is also a proponent, or was a proponent, of using consultants for that kind of thing. And it's very normal to use consultants to implement that kind of thing because you don't want in-house resources, specialists for that kind of thing. But we probably underestimated how complex a platform it was, how complex a setup it was, and that it wasn't enough to just have a Sam and a couple of others. They can probably just get into Zuora a little, and they are good at everything else, so how difficult can it be? So I think perhaps we underestimated the need for expertise on the platforms we had.

**Ulver:** Yes, because it was, it ended up being quite a lot that had to be moved over as well.

**Thomas Agerbæk Ruby:** And then of course, you can always say that you have consulting firms, we used Subscription Factory. They do the job well enough, [why] only blame others. Where should the knowledge lie, etc. And how much can you expect from vendors. But I have talked to them many times, and I know that they have, or that Alex has said to me, that he thinks it was too complex a project. He doesn't want that kind of thing. They would rather do some simpler projects, but as consultants, you never say no, and they want to earn money, etc. So you won't get them to say no. You won't get them to say, "Oh, this is perhaps a little too much for me and for us." So there has clearly been a bit of a mismatch there too, in hindsight, it has been too complex a task for them, combined with the fact that we haven't had in-house expertise from the start, that has been a bad combination. It's also difficult/time-consuming to do.

**Ulver:** What about something like, I mean the management itself, top management, where we talked a little about them not being super involved in the whole process. How do you see that whole part of the project? If it could have gone the way you wanted, how should it have been handled?

**Thomas Agerbæk Ruby:** I think it's a shame that Henrik didn't prioritize it more from the start. But it's easy enough for me to sit and say, if you are in IT, and if you read something about IT change management, it says on page 1 in all theories and books, that if it is not anchored in top management, then you can almost forget about it, and then it ends up failing or all sorts of things. But it's easy enough to say, in a company like Veo at that time, Henrik was struggling with things that made a project like this seem like the absolute lowest priority. He was struggling with the fact that we couldn't produce enough cameras. He was struggling with the fact that it was corona time. He was struggling with many things that were much worse, immediately, for the Veo business. So it has probably...

**Ulver:** It has been simmering, and it has had the potential to become...

**Thomas Agerbæk Ruby:** I don't think from his perspective, I just don't think this... it hasn't been very important compared to the other things. I mean, not being able to produce cameras for your customers. And after I started, I was the first with Veo Cam 2, which was delayed by over six months. They had sold them, they had taken money from the customers.

**Ulver:** And that was the same story with the three.

**Thomas Aserbæk Ruby:** Yes, and a little the same. Not as bad with the three, but it was much worse than the first time. So, seen in that perspective, something like this can easily feel like less important things on my desk. So I can easily understand if I haven't seen it summed up. But that doesn't change the things that play into this kind of project. So it's not that you discuss, then you wait for a project or a green light [likely 'go'], whatever. Will there ever be a right time for something like this?

**Ulver:** No, there isn't. I've been thinking a lot that for me, this project feels a bit like being a teenager. You go from being a child to entering an ugly period, where it's something you just have to get through. This thing about changing systems to something that is much more scalable to work on. On a much higher basis. And I don't think there are chances as a company, you have to go through this period. Almost no matter what. I think there are very few companies that just solve it really easily. So that thing about being able to talk about that decision was bad, that decision was bad, that decision was bad. But they are just decisions you have to go through, to be able to come out on the other side and then have learned that now we know exactly how to handle these very complex scenarios.

**Thomas Agerbæk Ruby:** There are very few who think that kind of project is fun. That's not what people think. We make cool cameras, cool recordings, and all that.

**Ulver:** It's somewhat necessary under this kind of thing for many people.

**Thomas Agerbæk Ruby:** Exactly, it's like that teenager period that you just have to get through. If you don't do it, you'll never make it in some way.

**Ulver:** What do you take away from this project? What is one of the biggest learnings when it comes to scaling your IT projects?

**Thomas Agerbæk Ruby:** I think there are two things, I would say. One is change management. How important it is to constantly focus on change management. Especially in an organization with such high turnover. I think I wrote down ten names of people who were involved in that project. Who are not here at all today. So change management, the fact that there were so many changes, on the resource side, that is crucial. And we haven't had that. There hasn't been anyone responsible for change management. Yes, Christian Houen a little, but it hasn't really been a dedicated thing. And the other learning is this balance between internal and external. There are also large projects on new technologies that you don't know internally. This thing about using external balance between us, because Veo is a company where things have to

happen quickly. So that thing about first having to hire someone yourself, you can just forget about that.

**Ulver:** It seems like it will take too long.

**Thomas Agerbæk Ruby:** It takes too long. I wouldn't even wait. We needed the solution last week. Exactly. And that is... That can be dangerous.

**Ulver:** Yes.

**Thomas Agerbæk Ruby:** So, in hindsight, we probably should have found someone who could act. Internal experts on Zuora. I mean, a... A competitor to Subscription Factory. Whatever. Someone with some Zuora knowledge who could help ensure the quality of the solution.

**Ulver:** Something that can also be said is that, at the end of the project, the team kind of ended up having much of what the team would have needed from the start. There was a Product Owner for both Salesforce and Zuora. There were Developers. It was much more... I mean, the resources were there in the end. So, of course, if you had started there, you might have cut a year off the timeline. And not only that, it would also have been much more stable all the way through. And did you kind of feel that the thing with the external consulting company and you, they don't understand Veo as such. You have to spend a lot of time sitting and teaching them. "What does Zuora do?", they can then say. "Yes, but this is what Veo does. And we must have it exactly this way," etc. So, you can kind of say, I think that the product would have ended up differently if it had gone the other way with someone like Jesper involved from the start.

**Thomas Agerbæk Ruby:** I don't think that the product or the solution itself, overall, would be very different, but I think the quality would be much better.

**Ulver:** Also everything that is behind it, workflows and everything that is connected.

**Thomas Agerbæk Ruby:** The quality of it actually working.

**Other speaker:** Yes.

**Ulver:** We saw the consequences of it towards the end with, I don't know where, towards the end as you also experienced it with Subsavers also joining. We all became part of Builders. And then the entire focus of Builders was that thing about, now we just have to, because so many problems had started to arise everywhere. There was a problem with the cameras not being delivered, at the same time as we have had a huge migration period, where customers said that now we don't have our entitlement, or now we don't have what we paid for, etc. So there was just, I would say from when you finished, probably September-October or something like that, and from there it was just a period of firefighting until December, where it was just, now all engineers have to sit and do tickets if they have time, and sit and help customer support if they have time, etc. What could then be seen in the effect was that Trustpilot went down. I think we had a score of something like 1.9. All the way down in the red numbers, and everyone was just

writing angry, angry, angry messages and stuff. If there was an all-hands or a Q3 kickoff, then... then it was also just so clear to Henrik. Now it's Fosbury that matters. It is the overall priority. So the focus did come, it just came too late, you could say. And then it also helped, of course, that we were kind of absorbed by a huge team, Builders product, with a lot of engineers. And now you can say that Veo has somewhat come out on the other side, where Business IT no longer exists. We have all become part of Builders, and everything we were working on has been divided into three different teams. It ended up happening that the resources naturally came along by being absorbed into other departments. And then suddenly there are six engineers looking at it instead of one or two. So now I think that sort of... I mean, the last pimples are gone. You have kind of grown. Now you are kind of ready for that part of it. Of course, there are still some... I think maybe 99.8% of sales happen in the new setups, and there is still some strategic account management happening in the old system. That's also something to remember, that the project did succeed in the end.

**Thomas Agerbæk Ruby:** I hope so.

**Ulver:** Even though it was ugly along the way. So that's also what I'm looking at a little bit, a post-mortem that has been done of the project. And that is also the first point, that's what you have to remember to say, well, it succeeded, we did it.

**Thomas Agerbæk Ruby:** What you're saying now is also very interesting in relation to where it all started. Yes, it has been a huge period with... Product didn't want anything at all to do with Zuora or Salesforce or anything.

**Other speaker:** Not at all.

**Ulver:** And that is also quite naive.

**Thomas Agerbæk Ruby:** It's a choice. It's a choice. It doesn't matter whether you call it business IT, or Builders, say what you call the work that needs to be done. But it is a problem if you change religion and strategy while you are running. It can be very difficult. I think it's a bit like, if you have to sum it all up, then changing the fundamental strategy while you are running with a project like this, that is, and you also see that with all the people who are no longer there and have quit and all that. It has cost.

**Ulver:** How do you think one could avoid that in a place like Veo? When you take on a project that takes two and a half or three years.

**Thomas Agerbæk Ruby:** Yes, I think so. If it had been more anchored from the start with Henrik, and a strategy for how we run the business, and who does what, so that we don't reorganize the entire departments that are working on it, while we are developing the project.

**Ulver:** So you have a lack of what [one] calls governance, meaning there is no strategy from the top about how we handle our sales systems.

**Thomas Agerbæk Ruby:** Yes, I would say the organization of the company. The organization has changed drastically during this project. With what people are hired, in what departments, and doing what. It has been under constant change throughout this project. And there have been a lot of people involved. It's not a good foundation. And there I would dare to claim, at least the time I have been there, it has been more chaotic than in other [places]. I have been in startups before, or scale-ups before, but there is simply not the same turnover. No.

**Ulver:** Where do you think it comes from, that there was so much turnover?

**Thomas Agerbæk Ruby:** I don't know.

**Ulver:** I've only seen it from the outside, but there was a COO who liked to fire people here and there.

**Thomas Agerbæk Ruby:** Yes.

**Ulver:** I never really talked to Ditte or her. I have no idea about her professional integrity, etc. But there was certainly not such a stable area there.

**Thomas Agerbæk Ruby:** It's no secret that those of us who were under her, we often looked each other in the eye and also talked a lot by the coffee machine and so on. We just stood there bewildered on the sideline and looked each other in the eye. You can also say that the CFO and accounting manager, who were involved at the beginning. They are no longer there either, they have also been replaced. So there are many core resources around the project that have been replaced. Christian Jönsson was the head of all sales before, right? He was also stopped, right?

**Ulver:** Yes. So there has sort of not been the stability at the top that is needed?

**Thomas Agerbæk Ruby:** No. In my opinion, there has been a lack of fundamental basis and baseline on what it is, what it is with the resource management [likely]. I simply think it is too weak. And... Otherwise, you need a much stronger project organization. Then it doesn't work that you have one program manager, Christian Houen, who has to handle everything and try to make this work across everything and everyone. At the same time as you are undergoing a change in strategy/direction from product, because Jacob Frost is stopped, and Kruse has a different perspective on things. It's an impossible task. Christian Houen has been without a chance. You can say a lot about him. I felt a bit for him sometimes.

**Other speaker:** But he has also been without a chance. Completely.

**Ulver:** What was his role?

**Thomas Agerbæk Ruby:** He was program manager.

**Ulver:** Program manager for Fosbury.

**Thomas Agerbæk Ruby:** He was hired solely to drive this program. Or project. Or whatever I should call it.

**Ulver:** And do you know why he was fired?

**Thomas Agerbæk Ruby:** Uhh... I don't know, but it is my very clear conviction that it was Kruse who didn't think he was doing a good enough job. Okay. I have heard something about, why the hell do you put 28-year-olds in charge of driving such a complicated project.

**Other speaker:** Because it's a complex project.

**Ulver:** Yes.

**Ulver:** That's again the thing... Because it's a complex project. Yes, there you can just hear the prioritization.

**Thomas Agerbæk Ruby:** And Kruse has obviously not agreed with the choices that have been made. And Ditte's choices, and also not Christian's role, etc. And that is not healthy for any project if such an important player does not agree with the setup and the resources, etc. So I think the only thing that could have made it better, was a stronger project organization. When you don't have a more stable anchoring in the departments in the resources, when there are so many moving targets around and people disappearing in every way, then the only thing that can work, would be if you had a much stronger project organization that did nothing but this project.

**Ulver:** Yes. That will, I mean, it comes back again, touches back a little on that thing about, I mean, prioritization of resources and lack of responsibility from top management. Can you, would you say that? Or prioritization at least, right?

**Thomas Agerbæk Ruby:** They probably underestimated how important it was to prioritize some project resources or resources around this. And some anchoring around such a project.

**Other speaker:** Yes.

**Thomas Agerbæk Ruby:** Underestimated the complexity of it. Felt there were other things that were more important.

**Ulver:** When you look back on it, the project, what is your feeling? Do you feel that everything changed almost every day? Unexpected things arose constantly? Or what is your feeling when you look back on it right now?

**Thomas Agerbæk Ruby:** On the entire Fosbury project? Well, so as not to repeat myself, it is... The project as a whole, the most significant thing is the change from the starting point, the background for launching the project, the resources and the plan that was back then, compared to where it has ended up now. It is 180 degrees, it is a paradigm shift. And then combined with all the changes there have been on the player front, among those who have been players in the project, combined with the fact that the vendor, the big vendor, thought it was too big a task for

them too. Too complex a task for them. It just becomes together a challenging experience that is difficult, and that we have all had to struggle a lot with.

**Ulver:** Where, if I just need to understand precisely from that top management-governance perspective, were there regular meetings between you and Henrik and Ditte, for example?

**Other speaker:** No.

**Thomas Agerbæk Ruby:** No? I think I have seen Henrik in meetings a maximum of a handful of times approximately. Okay. Out of all the, what is it called, steering group meetings and steering committee meetings there have been.

**Ulver:** What was your feeling about his knowledge about the project?

**Thomas Agerbæk Ruby:** Very little. He was involved a couple of times when things started to heat up. Because we all know that change implementation and so on is important. But it's obvious. He hasn't been part of the project. He has hoped that he could delegate it. He should also be allowed to do it that way. But, yes, what should I say? It's always easy to be wise after the event.

**Ulver:** Is there anything you want to add? Is there anything you feel is unsaid? Something you have thought about? It is very exciting for me, because it is the first time I am looking at the project from the outside, you could say, or in its entirety, and can speak more freely and honestly with everyone about it, because I know that if you are there in a professional perspective, there are perhaps some things that you are neither entitled to hear nor talk about nor ask about, etc. So it is very exciting right now to understand the whole project.

**Thomas Agerbæk Ruby:** What I think should certainly be an important part of the story, is where it all started. Because it was a completely different setup. It was completely different people. It was a completely different strategy. A completely different plan than where it is today. And that journey there, where you start such a large and important project on one basis, and then suddenly everything is completely different.

**Ulver:** I think that is an incredibly important perspective. Yes. And also part of the development that you have to go through.

**Other speaker:** Yes.

**Thomas Agerbæk Ruby:** I think so. And again, back to something like change management. How the hell do you ensure that everyone, because in change management, it's about making sure everyone is on board with this [project]. You don't have any opponents. No. How the hell do you ensure that? Then someone comes in from the right, new people. What the hell are you doing there? Why are you doing it that way? They haven't been part of the decision themselves. And that, when people themselves haven't been involved in making the decision about one thing or another, or the way to do it, it is volatile, I have learned. Volatile. So when almost 90%

of the people who were involved in making the decision at the start, are not involved at the end, but all sorts of people come in who haven't been part of making the decision, then it's really, then it's difficult.

**Ulver:** It's just about really not underestimating the project you are embarking on, and then making sure that perhaps the people, I mean, have the right people involved in the project from the start, before you even consider, is it now we press the button and get started. Because you have just learned and seen from this project that it's just something we start now and then we fix it along the way. And then we build the train tracks as we go. That has just created problems or challenges.

**Thomas Agerbæk Ruby:** If the family you start with agrees on, "Now we agree, now we'll really get started with this." If that family is scattered to all sides, and it's all sorts of others who have to take over, that it's tough going, it's certainly a poor foundation to run such a large project on. And you can say a lot about Veo, but just the culture around creating a work environment, around such a project, where people want to stay and be, and not just get fired, etc. It's not good. They are not good ingredients for such large projects. But like everywhere else, new people just come in. And you find solutions, and of course you figure it out, as you also say now. It can just be more or less painful. And the very worst examples with the state's payroll systems, it ends up going completely wrong, because it is so complex, and you don't quite have the same agility as a private company, so there it can go completely wrong. It's a place like Veo, where you find a solution in one way or another.

**Ulver:** I don't think I have much more.

**Thomas Agerbæk Ruby:** On the positive side, I would probably say that if Veo is going to, which they probably will within a foreseeable period, sell or present itself on the market, then it is in something like technical due diligence, which you typically do to prepare tender material, etc. Just like when you sell a house, etc. It will look good that you have dragged yourself through something like this. Yes, precisely. You can write that you are running on something like Zuora and Salesforce and such connected platforms. That's the kind of thing where they say, "We can build on this." Whereas if you are in something home-built and all sorts of things, everyone knows that can be difficult.

**Ulver:** Yes, but then it must be after we have just been through what we have been through.

**Thomas Agerbæk Ruby:** Yes, so if a buyer comes and says, "Okay, now I can realize value, there's a billion in this, and we need to scale it times 10," then it is incredibly important that there is a scalable setup.

**Ulver:** Yes, part of the foundation. And we also talked with Sam and Øivind that the business IT itself, or whatever we call it, is very much the backbone of a company. But then again, you only notice your backbone when it hurts. Yes, exactly. It is only then that it is prioritized. It's not *before* it might start to hurt. If you keep walking this way, you might get back pain in 10 years. But then it's only in 10 years that we work on it. You can then also say, [it] has very

metaphorical things, so that's what we focus on. And then the backbone, we'll look at that when we get there.

**Thomas Agerbæk Ruby:** And that is also a classic when working with that kind of thing. I have stood at, I don't know how many all-hands meetings down there, where Sam and Øivind are the most clearly involved. Like Sam, he embodies all the emotions about it. But every time we have to stand and applaud ourselves, how skilled we are at selling, and new features in the product. No one has ever appreciated this kind of work. No one wants to hear about it. No one wants to deal with it, because it is not sexy, fun, customer-facing, even though it actually is, it's just a bit behind the scenes. Super classic with this kind of thing. And therefore, management can also probably be prone to prioritize it a bit too low. I don't know if you have read about it, I have heard the story about Skagen Watches.

**Ulver:** No.

**Thomas Agerbæk Ruby:** They did an ERP project. They sold their business for 1.5 billion to Fossil Group. They were about to go bankrupt, had to mortgage their house and everything, because they had gotten themselves into an ERP project. Then it was outsourced to someone who actually knew something about ERP. It ended up that for 12 months they couldn't invoice their dealers. They were about to go completely bankrupt. The company was about to close and everything, because of a bad ERP project. None of them wanted anything to do with it. They make cool watches, man. It's damn cool. No one wanted anything to do with that ERP. And it almost ended up burying them. Just barely. Then they got it up and running, and got some resources in who could help make it work. So the story is happy. But they sold it [later?]. And probably a bit the same with Veo. It's just another example.

**Ulver:** A different project.

**Thomas Agerbæk Ruby:** It's probably a bit the same with Veo. Exactly. It's just an example of underprioritization. Whether it's that guy who figures something out with ERP or something. You fix it.

**Ulver:** It's also, if you read, for example, this book Digital Maturity by Pernille Kræmmergaard, which is a textbook for Master's level, or also for leaders about digital maturity, it also says in the entire first third of the book, about how you as a leader today, cannot be a leader without understanding technology and digitalization.

**Thomas Agerbæk Ruby:** You can just forget about it. It says, yes, exactly. You will just be overtaken, and it's only a matter of time before your company has buried you.

**Ulver:** It says on the first page of the books I have read. It's the first thing.

**Thomas Agerbæk Ruby:** And there Veo is probably also a bit influenced by the technology they find cool and live off, and the product and the cameras. But this digital maturity we're talking about, if you read Kræmmergaard for example, it's your entire setup. It's also precisely

the way your back-end systems work, and how you get money in, you name it. Your ecosystem and all that.

**Ulver:** I don't think I have anything more.

**Thomas Agerbæk Ruby:** So it's a super interesting task, I think, in a contemporary context.

**Ulver:** And it is too.

**Thomas Agerbæk Ruby:** In modern companies.

## Interview 3

**Øivind / Thomas Master's Thesis - 2025/03/20 10:05 CET - Transcript** Attendees: Thomas Agerbæk Ruby & Øivind Larsson

### Transcript

**Thomas Agerbæk Ruby:** So let's just start a little bit about yourself, your role, the project. Yep. What were your main tasks and advice? Yeah.

**Øivind:** So, just taking it one step back, I started at Veo the first of September as a systems engineer in business IT. That was coincidentally two days before they kicked off the Fosbury project. So when I came in, everything was decided on what we wanted to use, and then I was thrown into starting to understand the different systems that we had actually chosen and to figure out what kind of integrations we needed and how they worked and how we should get them to work. So, I think one of my first things was looking into Adyen, our current payment gateway. We're currently using [Stripe or another system? context needed]. Because in the start, we didn't have access to Salesforce. We just started getting externals in to help us build it, right? But then once we started getting access to that, it was one of the first thoughts I had was actually building the checkout in Salesforce. I remember...

**Thomas Agerbæk Ruby:** Did you feel that you had clear expectations and direction from the beginning? The project was just there?

**Øivind:** I think it took almost two months until I was actually included into something like that, where I actually started getting tasks and getting concrete work that I was supposed to do

towards the project. One of my first things I looked into with Adyen was a live streaming thing because they could support that. Yeah. And then the guy in the line that owned it didn't work here anymore, and then suddenly it all fell apart. So, okay, that didn't happen.

**Thomas Agerbæk Ruby:** And how did your role evolve during the project as it progressed?

**Øivind:** Yeah, I would say my role kind of fell into every little pocket of what we needed to do, because I think everybody from the start underestimated how huge these systems are. So, my role description was only to manage the integrations between the systems and ensure that it all worked together. But then it evolved into, suddenly, I was a Salesforce admin because we didn't have a product owner. Then Arena came in, and I became the Zuora admin. And then Jasper came in, and then I kind of fell back to what I was supposed to do in Fosbury. So, I was a support agent for a while.

**Thomas Agerbæk Ruby:** And did you feel understaffed during the project?

**Øivind:** Yes. Yeah. 100%. Yeah.

**Thomas Agerbæk Ruby:** So, you felt like there was a lack of maybe priority for the project?

**Øivind:** Yeah, I could say that. I think it had a good priority. I think we underestimated the amount of work and the amount of change management that came with it. Mhm. And from my point of view, management thought they would copy-paste our current setup with new systems so that they could scale. Mhm. They kind of wanted Shopify, just better. Yeah.

**Thomas Agerbæk Ruby:** And when we're talking about migrating to a new platform, obviously it was underestimated how big the project would be, how long it would take. So, what were the main technical challenges that you faced during the project?

**Øivind:** A lot. Payments was a big one. We had huge problems because of requirements from Veo that [wanted] money upfront, but not for hardware – that only needs to be authorized – but software you should charge upfront. And us, then we're completely novices in this field, and then I had to build this whole thing from scratch using a little bit from Zuora, and we just had to figure out on the go whether it worked or not. And when it comes to credit cards, when we sell to people all around the world, it's impossible to test for every scenario on every bank and every card issuer that exists out there. So, once we went live, we had some huge issues with issues like the fraud, for example. So, that was one. And then the second one is also a little related to this, but it was around our requirement of, "You have to activate your software yourself," so that every subscription has to be in a pending state. We didn't take into account that Zuora is not meant to work like that, kind of. It's supposed to be invoice, then payment, and you're already active by that point, not payment, activation, invoice. We had that a little turned around. And then let's think about one more thing about... [I] wrote it, but there was several things like just

the fact that we had nobody in-house that knew Salesforce and Zuora was a big thing as well. Yeah.

**Thomas Agerbæk Ruby:** So, the way the project was, like the main technical experts were outsourced, were external. Yes. So, how did you feel about that situation? Was there any complications from that setup you used?

**Øivind:** Yes, we had a lot of back and forth with the consultants, but the biggest problem was, from my point of view, that me being technical, I wasn't introduced to the consultants until way into the project. I don't remember exactly when, but I remember that we had, what was it... project manager. So, we had Rasmus [likely Name Redacted] and Ulver [likely Name Redacted] in our team. Christian was also working on this. Those three were the ones that dealt with the communication with our consultants, which meant that no technical people were in the room when they talked to the consultants. So, a lot of the decisions that they had gotten from their stakeholders were then translated in a non-technical way to the consultants, who then were like, "We can do it," because Zuora can do anything. But they also will try to say that you should probably push back on this, but then it was more like, "No, we need this." And then when they implemented the work and it got handed over to us that were actually technical, we were like, "This doesn't make sense at all. What is going on there?" Yeah. And then it all points back to the consultants doing bad work, which is the standard thing. So, it's a lot of wasted time going in between, lost in translation, because a lot [of] middle management and the people got some requirements from top management that they had to explain to the consultants, and then it was just this letter that went both ways. Yeah.

**Thomas Agerbæk Ruby:** How would you have done it differently if you designed the project today?

**Øivind:** I would 100% have a technical leader always in the room when these decisions were taken with consultants, whether or, most likely, we would need someone experienced with the systems that could actually challenge what our consultants proposed and challenge the requirements that we came with, just to say that, "This doesn't make sense. You should choose this system if you want it this way," just an example. That's probably the biggest thing I would do differently. And I would also earlier start the conversation around how to implement and start knowledge transfer a lot earlier. Because we don't have a technical person in the room... for example, me and Sam, we could have been set on doing Zuora certifications from day one if we wanted to, because for two months we dealt with mostly Shopify until we were handed off the project.

**Thomas Agerbæk Ruby:** Do you feel there were enough opportunities for knowledge sharing and internal learning, for example? Were you given time to check these new systems out and learn about it before just being thrown out into it, or...?

**Øivind:** No, I learned as I went. That was the biggest learning. I had was... often we went live in production so that I could sit in Zuora and see how everything worked, and every request I got I just had to do trial and error to figure things out. Yeah, that was the first time I was properly learning or having time to learn because I had to.

**Thomas Agerbæk Ruby:** And maybe touch upon it a little bit, what... but did your role play, or did you play any role in decision-making during the project?

**Øivind:** Not on critical requirements, but when it came to implementation specifics, then I did have a good say in what... or I got personal requirements, and then how to implement it, that was kind of on me and Sam to do ourselves. And at one point, I was the only one that wanted to do anything in Salesforce, so there I had full autonomy of everything that was supposed to go out. And I wasn't... so not involved in decision-making on a high level, but on the technical side, yes.

**Thomas Agerbæk Ruby:** And during the project, there was a lot of change in terms of the people who were responsible for getting it done. How do you feel like... did any leadership change requirements, and was there any back and forth and new starting from scratch again, or was there any situations where you thought about, "Whoa, there's a lot of change here"?

**Øivind:** A lot of change, but not really [affecting requirements fundamentally]. From my point of view, when Christian was running this, he tried to have a clear view of starting from epics, like having everything written down, and then do the whole thing to release it. What I saw when that management changed after he went was that nobody went back to look at those [original] features at all. They didn't figure out any success criteria or whatever. They had been tasked with getting this project done. So they didn't care about that. They were like, "What are the blockers? Why aren't we done?" That was the only mindset they came in with was, "We need this done yesterday. What is missing? How can we get here?" So, yeah, we had one like that. And then we had something else that came in when the migration was ready, but we were a little hesitant to pull the trigger. Then we got some new management in that was like, "Let's do it now. Let's just migrate everything. Let's do it. Come on. Let's just do it like we're waiting for." So, it's like always there's a rush of getting it over the line so that we didn't have to deal with it anymore.

**Thomas Agerbæk Ruby:** [Someone] felt like he knew that when I talked with, for example... he felt like there was a big problem with the new people coming in who didn't know, who weren't there in the room at the beginning, and then they came in with a completely different view because they didn't know everything about it, and then they said, "We should just do it this way that I've been doing it for the last 10 years," or, "No, we should do it this way that I've been doing it." So, did you feel any of those views on the project?

**Øivind:** Yes, I get that comment, because I remember every time we got someone new in, we had to explain the system. It was always like, "But why aren't we doing this?" And then we're

like, "No, this is why. We've been at this point before." And then we did spend a lot of time explaining all of this and then coming up with their solutions and then us having to shut it down because of previous requirements. Yeah. But I think that's completely normal, and I would rather have new management come in and try to put some question marks on our requirements or what is going on, rather than just running with it and expecting everything to be perfect. Because I've mentioned this before to the other people I interview. The way I see the project is, it is a project that all scale-ups will get through at one point, and it's kind of like a teenager period where you're going from being a child and everything is easy and new, and then to actually mature, you have to go through the teenager period where it's very ugly. You get pimples, and we have phases, and you have to get through [it]. Nobody wanted to do this project. It's not like you're just waving your arms in the air and just being happy about this getting done. But it's a necessity for a company to get over to the other side and actually grow even more and be able to scale even more. So, yeah, basically what I'm feeling, one more, is this project was a necessity for Veo to get through. Of course, there were other projects as well that also [focused on] just staying alive, getting money and stuff like that. But if you complete a project like this, then you are very well set for the future. Yes. So that is one of the learnings I've already had.

**Thomas Agerbæk Ruby:** But would you feel that if you were to do this today, would you spend more time on creating a team that is going to be there for the project?

**Øivind:** I don't care that much about people going in and out of the project. I think we need more emphasis on documentation and proper handoffs and actually having a shared vision so that the people are not expendable, but if somebody goes on paternity leave or something like that, we don't lose four weeks of work because that person is the only one that could do something. Like, we need to have this setup that all tasks in the project can be picked up by anyone, that coming in and coming out is with not that much overhead. We ended in a state where if I had left at some point, we would have been set back too many weeks or months to even get something done. And then Sam left as well. So I was also alone at some point. So it was impossible for us to get anything done. Yeah. So I think that would be my main priority is to do the due diligence before we start a project, of having all use cases [and] requirements defined, set in stone, and if somebody comes in and says, "Hey, shouldn't we do this? This is a really smart idea," we can actually say no and keep working on the project as is. Yeah, because the scope changed for the Fosbury project so many times over the course of it because nobody knew what had been decided. So, making a clear strategy and scope before starting a project like this, which was maybe, if you look back on it, it wasn't done because it was highly underestimated, and they thought they could do it in half a year and it would just be plug and play, and then we go and we have consultants who will fix it and everything. Yeah. So that is a clear learning too.

**Thomas Agerbæk Ruby:** But we also talked about in the beginning having internal knowledge and so from the beginning, and finding an arena from the beginning, even though it takes more time to actually start the project. Yeah.

**Øivind:** Yeah, I would 100% do that. Or either that [or hire] someone that's very used to running projects like this. They have experience in running several projects like this. They know the pitfalls. They don't underestimate it. They know how to deal with consultants and stakeholders and get everything online that could run it. From a technical perspective, I would have loved if we had somebody that knew Zuora from the start. But I have a feeling, if you think about lower management, they would have loved someone that was strong and experienced in running projects like this as well. I think we lacked in both those aspects.

**Thomas Agerbæk Ruby:** So, obviously the requirements changed a lot throughout the project. Do you feel like the developers, you and Sam, were given enough time and resources to adjust to changes, or was it just reactive all the time? Today we found out that now we have to do this, or now...?

**Øivind:** Very reactive. We had periods where we could actually work on something ourselves, and then there were periods where whenever management actually took a look and saw, "Shit, what's going on here? A lot of payments are failing!" That's when, for example, [Name Redacted] got involved because then he got pushed from up, and then he pushed down, and then that's like top priority, reactive. So, for example, from my point of view, the Salesforce checkout, I was never able to return to it ever because of so many reactive changes that we had to do. So, I wasn't able to polish it to the way that it was supposed to be. So, right now, I'm still joking around that we're live with the POC [Proof of Concept].

**Thomas Agerbæk Ruby:** How were changes communicated? Was it always just [you] were told something from top and then doing it, or was there clear communication throughout?

**Øivind:** No structured communication. No structured communication. I usually go to Oliver or Christian at my desk if they needed something. That's usually how it went. If I was lucky, they had some task; else I had to make it myself. Yeah.

**Thomas Agerbæk Ruby:** And did you collaborate with other teams within Veo to do the project?

**Øivind:** Yeah, I collaborated with, then, the Club team, which is basically... my hero was the main contact point for the Entitlement service. Yeah, that was the big thing that I did, at least cross-team. Mhm. And then I know [we] collaborated [with] logistics on the hardware. I wasn't that involved in that one.

**Thomas Agerbæk Ruby:** And did you feel that they knew what the project was?

**Øivind:** If I remember correctly, I think they just had a vague "I don't want to get involved in this" kind of thing. But then Entitlement service was a big part of the Fosbury project, yeah, so they had to have some knowledge to understand what is going on, and they were also going to

own Entitlement service. So, they had a huge part in actually setting it up, getting it ready, ensuring fulfillment: "When should they get the payments? How should they get...?" All those things came from that. Yeah. And then we just built the integration to provide them.

**Thomas Agerbæk Ruby:** So, before the project began, I was told that if you had to look up a customer, there was no way of figuring out what product they actually had and what subscription they were on, and everything was just like a big mess. And also, they had never thought about a customer buying a camera again, having two cameras, or buying more than one camera in the first [place].

**Øivind:** So, yeah, so basically there was also just a lot of getting entitlements done. Or who came to that conclusion, how are we going to structure it? If I remember correctly, this was the Club team, Builders back then. They had complete autonomy over this and how it should be decided. They came up with all the features. They came up with all the entitlement limits, like the upload minutes and upload... I'm pretty sure Sunna had a say in it as well because they wanted to monetize a lot of this, the upload minutes. Make it usage-based. And they also wanted to make live and live analytics minutes, and then they wanted to upload all usage into Zuora and bill them accordingly for what they have used. That was luckily pulled out of scope because [we] were already trying to get anything done like that. But I remember it was Dina from the Club team that came with all the requirements regarding entitlements, and she used to be the product manager of the Club team. So, yeah, she was that, and then my hero was the tech lead. So, those were the ones that drove this. Yeah. And I think Koo had a say as well. At least I saw he had a comment or something in the Notion documents. I'm pretty sure he was involved.

**Thomas Agerbæk Ruby:** Yeah. Have you ever thought about why you were the only team actively working on this? Some of the other teams were very hesitant to take any ownership of, for example, finance never wanted to come on board on the project, and builders only wanted to work on the product and not on any of these backend programs that we were dealing with. Mhm. So, have you ever thought about a more appropriate way to get these other teams involved?

**Øivind:** I haven't thought about it in that way, but after we moved to builders, I see how much they could have helped. Does that make sense? When I was in there, it kind of made sense because we were the ones responsible. Mhm. But now I see that, yeah, builders had this capacity and was able to help. Finance, on the other hand, I completely understand. They're swamped in work. So, I get that they didn't want something new. They didn't have the luxury of being able to look into new stuff and send it on correctly, which is an understatement on their behalf, but they also can't get people. So, I don't blame them, to be honest. When it comes to builders, I am a little afraid that it was a little political thing as well, because I know for a fact that Alex worked a lot on trying to get business IT into builders way before it even happened, and I know there were some on our side that pushed back and didn't want that. So, I know Schultz has spoken into it as

more assumptions, but you think that's why builders wasn't that helpful when we were working on this?

**Thomas Agerbæk Ruby:** From my point of view, it could be anything.

**Øivind:** Like they had their work, they didn't know. It wasn't like we were communicating openly. So, yeah, I think... was that... there was a guy called F... yeah, Jacob. Yeah. And they were kind of internally battling which way to go and have different opinions and philosophies of how to deal with the project. That makes sense. Yeah, because Jacob, I know, pushed back at any point that builders suddenly got involved in business logic. That was his big thing. Probably shouldn't care about... completely separate. So, that makes sense if they were on both sides of the discussion.

**Thomas Agerbæk Ruby:** So, I think I want to talk to Alex as well. Yeah. To hear about his thoughts about... there were definitely political aspects, and also could expect a little bit more about what did they want and what was... wonder why the change was eventually it came that we got absorbed by builders.

**Øivind:** Yeah, let me see. If you could change one key aspect of the project, what would it [be]?

**Thomas Agerbæk Ruby:** Change one key aspect. That's a good question. Are you looking for a specific requirement or just...?

**Øivind:** Just say... just anything you feel like if you could have been God and said, "I'll just snap my fingers," and then this is the main thing that you're looking [for].

**Thomas Agerbæk Ruby:** I would snap my fingers and then the requirement of money upfront would go away.

**Øivind:** Because it would simplify the checkout. We wouldn't need the checkout flow. We could use Zuora's out-of-the-box invoice and payments. And it would just simplify so many things that have been a problem at the moment. But yeah, when you get into it, you hear about sales commission, you hear about finance getting money. Yeah. That it's difficult when it's still a company that needs money, and upfront, exactly. That cash flow is very important, but it would have simplified so much for Veo from the start.

**Thomas Agerbæk Ruby:** Yeah. How has your experience with the Fosbury project shaped your approach to working in large-scale projects?

**Øivind:** I think from before, I always heard about these [stabilization] projects or whatever you want to call it, scale projects, they never work. It always goes to shit all the time. Yeah. I was like, "Yeah, that's because it's a public sector. They don't have anything and only use

consultants.” And then I start here, and I figure out that it’s basically just the same as I’ve heard about it, even in the private sector. So, from my point of view, I think it’s an extremely nice experience to have that I’ve been a part of this because I’ve seen a lot of the faults, but I’ve also seen when it can go correct, when builders absorbed us and all the people jumping on the sinking ship or whatever. And now it’s actually kind of going good. Even though it’s their lives... yeah, we have found the duct tape and put it in the right places at the moment. So, it feels okay. Yeah. But just to summarize it, I can see why companies fear these kind of projects, and I understand that you really need to have guardrails in place for this to work correctly. If scope starts changing in the middle of the project, then you’re basically doomed. You’re fucked. Yeah. Yeah. And one thing to notice is also that the project ended up successful. The main goal of changing systems happened. Yes. But then there was [a] very different road. Lots of little noise that went on to the final destination. And also just it didn’t take half a year, as they would have, as they said in the beginning.

**Thomas Agerbæk Ruby:** Is the project fully done?

**Øivind:** I’m speaking from my team’s perspective. We kind of consider the Zuora we have now as... we are starting to view it as almost legacy already, and we want to rebuild it. So, you can say that Fosbury is done, but there might be a new one coming if we get our way: 2.0. Yes. We’re never going to call it that. But no, we’re coming in with fresh eyes, and we want to challenge the requirements and the business requirements that we set and see if we can make it in a simpler way.

**Thomas Agerbæk Ruby:** Then you’re going to change the requirements throughout the project a million times. Good. We’re going to do everything that we did before. We’re just going to start now. Do it. Exactly.

**Øivind:** Yeah. But yeah, just to give you... yes, no. I think I consider it done. Yeah.

**Thomas Agerbæk Ruby:** Is there anything you want to add? Anything you feel is missing out of your story [about the] project?

**Øivind:** I don’t think so. I want to add that I think a lot of it was a success. It’s not only crap that comes out. It was a little crap being the engineer of it for [a while], but at the end, right now, it is going surprisingly stable. Yeah. Even with all of that bad whatever you want to call it that we have. And e-commerce, for example, was a huge success. They managed to get it right. Yeah. And that was also part of the project. Need to give some props there. Is that due to the internal knowledge actually being applied and getting help from other engineers in the company? I think for e-commerce, why e-commerce was a success was: one, they had a product manager that was very strict in defining use cases and putting them to guardrails, and he set realistic timelines, and if they weren’t done, he pushed the timeline. And second point is, the e-commerce project got

moved to builders because business IT couldn't handle it. So, they actually got a lot of people involved in it that could help with it. Yeah. So, I think already you see that if moving to builders would have [happened] before, maybe we would have been in better [shape]. So, utilizing the general resources is a good idea.

**Thomas Agerbæk Ruby:** Useful still. All right. I don't know if I had more. I don't think we've been around a lot of it. It... it's just going to be an article uploaded or something.

**Øivind:** Exactly. This guy said that... this guy... internal war. Yeah. Just make tabloid. Exactly. Yes. Right.

**Thomas Agerbæk Ruby:** All right. Bye bye.

*Meeting ended after 00:33:31*

## Interview 4

**Jesper / Thomas Thesis interview - 2025/03/27 10:32 CET - Transcript** Attendees: Thomas Agerbæk Ruby & Jesper Nilsson

### Transcript

**Thomas Agerbæk Ruby:** So that is going... If I can see ourselves... no participant... I have muted everyone now, so whenever you speak, it's... I don't get in the way at this point. All right. So the way I want to start is just briefly a little bit about your background at Veo and how you started. So if you can briefly introduce yourself and your current role in relation to Veo.

**Jesper:** So, I started at Veo in April 2024. Previous to that, my first job before Veo, I worked as an ERP consultant in the finance area. So, yeah, and then from there I moved to Zuora, working in-house at Zuora, not in-house, but as a solution architect, which is essentially a consultant for other businesses. So, my daily job was essentially meeting up with companies like Veo, implementing Zuora, either improving the current solution or implementing it for the first time. Yeah, so it's very interesting to come to Veo because obviously at that point in April... yeah, let's get there, I guess. But worked at Zuora, so a very suitable job, I think, for coming here. And at the time also, we didn't have any kind of knowledge in-house that had previous knowledge, at least. And now from April 2024, I also started as a Zuora product owner. So, essentially the job description is to own Zuora, making sure it's a good solution.

**Thomas Agerbæk Ruby:** Yeah, when you joined Veo in the beginning of April, what was the state of things on your end?

**Jesper:** So, at that point, everything was more or less already in place. I can't remember what the first terms of migration work and not migration... not fully like... yeah, the migration, I believe... I can't actually recall when it was live, but it was live when I got there. There was only not a lot of customers, so a small percentage of customers, but there were still customers live on the current solution. And the current state, the solution was essentially built by a third-party vendor for Zuora. And so a lot of... I remember coming here seeing the solution, and I thought it was probably not the worst, but the most complex solution I've ever seen from my work at Zuora, and I was there for around one and a half years. So, yeah.

**Thomas Agerbæk Ruby:** What were your initial priorities when you joined? Did you receive a proper handover, or did you have to pick things out yourself?

**Jesper:** Yeah, from a Zuora perspective, how typically handover of projects would be a very structured way of working. So, you would have a solution design document that would explain exactly how the solution is going to work. You would have integration documentation, usually at least integration documentation, that tells you all of the integrations to surrounding systems. So, it should be on a high level very documented. But coming to Veo, there was actually none or a very limited amount of that. So, what I did to get onboarded was just going through the system and having a lot of meetings talking to Øivind and Sam at the time, who were involved in building the system. So, yeah, some documentation, but not like these high-level explanations.

**Thomas Agerbæk Ruby:** So, you basically had to find out how to manage and put out some fires yourself? And talking of fires, when you started, would you say that there were fires anywhere, everywhere, or was it [just the] initial beginning to what we saw later in 2024?

**Jesper:** I think at the time, so, me starting here, I had ideas of how we could improve the solution, and this was actually what I was hired for. But then I spent probably the next months, if not half a year, just firefighting. And so, the more customers we brought on, so I think when I joined there was a very small portion of customers in Zuora, still working in the old system most of them. So, it was kind of under control. We knew that it was, for example, entitlements had some issues, and we had other issues, but it was still manageable with us three employees that could actually work, or four, whatever, that could actually work on the solutions. But as we kind of introduced more people, it became much more harder, and at that point, we actually had fires, I would say, everywhere. But it's always been quite difficult, I think, to maintain and understand.

**Thomas Agerbæk Ruby:** And you mentioned that when you came in, you saw this complex way of using Zuora, or this maybe inefficient way of using it. Could you dive a little more deep into what you mean? Yes, yeah.

**Jesper:** So, from my experience, how Zuora is usually worked is to use as much out-of-the-box features as possible. And you would build the solution as [it was] always meant to work, which is generally you place an order, you invoice for that order, you collect payment, you send all the transactional data, or do revenue recognition within Zuora or in an ERP system, etc. And then most of the business logic and workflows are handled outside Zuora because Zuora uses a specific coding language called the Shopify liquid, which not a lot of developers know. And so, maintaining that in-house if you don't have the knowledge and teaching people to actually handle it, it's very difficult to maintain. So, usually businesses would implement Zuora as the standard out-of-the-box, where you would configure a product catalog, you would have your subscription management, so invoices can recurringly be created on whatever is on that subscription, payment collections through native integrations to payment gateways. But if you want to collect payment before invoicing, then usually that logic or whatever it may be, a lot of the companies that I know would bring that outside Zuora because they usually have the in-house knowledge of maintaining that code. When we got here, or when I got here, I saw all of that logic living inside [Zuora]. So, it was clear that this was something that was asked to do, and the consultants doing their job accepting these requirements, how to build it inside, because that's... and so I think there was a lack of [internal expertise] present at that time.

**Thomas Agerbæk Ruby:** Yeah, so once you joined, were there any quick wins or immediate changes that you were able to do to stabilize the situation?

**Jesper:** I mean, I think one of the first things that I was involved in was just making sure... so because a lot of this custom logic and complexity in Zuora, a lot of the fixes also happened in [Zuora], and the most urgent one at the time was related to entitlement issues where we basically moved from one solution version to version two. And so, all of that was built by this third-party consultancy firm, and my job was essentially making sure that that was progressing. I was involved in some testing, etc., but it was very early in my days at Veo, so obviously other people were involved as well. But that went better and better, still with issues, but that was one kind of firefighting that did improve a lot from, I think, the day I started.

**Thomas Agerbæk Ruby:** I want to talk with you, and you talk a little bit about the business rules or what we call it, that we have our way of doing business is basically the exact opposite way of how Zuora is intended to be used. Yeah. Can you elaborate a little bit on that? Yeah.

**Jesper:** So, I think, and also the impression I got when I saw how this was built for Veo was, and that was what I've heard after the fact, is a lot of how Shopify or the whole system worked, whatever Bold, that was basically they were trying to move all of their economics they had back there into Zuora. So, they challenged some things, but I get the sense that they were just trying to bring whatever was there into Zuora and we do some additional improvements like aligning subscriptions and all that stuff. But Zuora has its best practices that should be followed, and that's not what we actually signed up for. And I know from my experience, I did also get this

exact request from other customers, which is a customer or a consultancy should always push back on that and urge them to follow best practices, which is probably what they tried to maybe initiate, but yeah, I think the requirements were also quite strong at the time.

**Thomas Agerbæk Ruby:** Yeah. What I've been told is that the management people that started the Fosbury project thought that they could take Zuora and just it would be plug and play. We can just match one to one what we have here and just in a bigger... a bigger program or a bigger solution. And I think that was the expectation from the C-suite or high up as well. And then when they explained it to the external consultants, they pushed back and said, "We don't say no to projects, but we wouldn't recommend doing it this way," but it ended up still happening.

**Jesper:** And yeah, yeah, also my thought is also that during the sales cycle all things were promised that maybe weren't [feasible], and I think the project from the start was probably not doomed to fail, but it was already starting [with challenges]. It was underestimated the size of it.

**Thomas Agerbæk Ruby:** Yeah. What I've been told and also the initial time frame was half a year, and that it would be going... So, a lot of during the project, some lack of prioritization from C-suites, higher-ups. So, it was basically just IT director down that had to manage everything without the right funding or the right internal knowledge, etc. Yeah. And then I think also one thing that I've heard that that did probably incorrectly was, you mentioned they didn't involve the right people. So, having these discussions and not involving stakeholders from all the teams... finance should be involved. There should be engineers from the team actually implementing this involved, lead architects. Marketing might be, but at least product managers from different teams should be involved, at least know what's going to happen and maybe request or be aware of what features or functionality will be brought along and not. And I think like you mentioned, they tried to do a plug and play so that whenever we start [using Zuora], we will have everything we have today, which is not necessarily the case or not usually the case.

**Jesper:** I would say there was also a political game going on. Finance, for some reason, did not want to charge the Fosbury project at all. They liked the way it was, and the people who were running finance back then didn't want to be involved at all. And also builders engineering, they didn't want to touch business IT. They only wanted to do product, yeah, not business IT at all. So, there were pushes from other sides as well, and then it was just laid upon business IT to figure this thing out. And that says a lot, I think. Yeah. If the internal system is probably the most important one, yeah, bringing revenue. Okay.

**Thomas Agerbæk Ruby:** So, back to the more of the Zuora... when you joined, how involved were you in decision-making about Zuora workflows, pricing, or subscription logic?

**Jesper:** In decision-making, not so much, I would say. I think from what I've seen, companies that are mature, larger in size, they're not as quick at delivering new solutions, all that stuff,

which is obviously a con, but they're usually more structured in involving engineering in decision-making as well. And so, I think Veo has a lot of ideas that we want to iterate and move fast on. And so engineering, at least at that time, I think it's getting better now, but at least at that time was very like... I don't know if necessarily 'evil' is the right word, but they would make their decision and then just push it down to engineering who have to build it. So, I think in terms of decision-making, I wasn't involved. I think a lot of my time also at that point went into firefighting. So, it was not like we were making necessarily strategic, smart choices for the system, because we didn't have time or the resource to do so, which also kind of... it fixed the issue in the short term, but we knew that unless something's going to happen or the management do a reprioritization, we will just keep doing what we're doing today. Yeah.

**Thomas Agerbæk Ruby:** Basically half a year for you to start developing on?

**Jesper:** I think half a year, something like that. Which is when we got absorbed by builders. Yes. That's where we got resourced to stay above water. Yes. So, what happened was, we were in business IT under a very small team. We were at the time, two full-time employees and a student worker. Because Sam left also, critical time. Yeah. So me and Øivind were basically the ones handling it. Then I know on Salesforce side, but we had one engineer and two specialists essentially trying to handle all of the systems and bring... and Shaw was not at the team at that point when it was at its worst, when we also migrated most of our customers. So, we were having around 20,000 subscribers at a time, I think, or maybe even more in Zuora with three people handling a system that did not work. Yeah. And so you would obviously see that it's not going to work. Where was I going? Yeah. So, what happened was, we got absorbed by builders. I think management started to realize that the internal systems are not working. Yeah. Which we tried to plug many, many times. But at that point, we got absorbed by builders, moving into a more technical point of view ownership. And from there on out, they just brought us a lot more resources. I don't know how many more we are today. We're probably... I don't know. Yeah, 500% times the size or something.

**Thomas Agerbæk Ruby:** Did you feel like there was a lack of prioritization from the C-suite/management for this project?

**Jesper:** Yes. Yes. I think there was a lack [in] implementing this project. I wasn't here from the beginning, so I don't have all the context, but it sounds like lack of resources and knowledge in-house. And I think so, once we actually implemented the solution that was built by third-party people, there was a lack of resources in the team handling it, yeah. So, I think those two things probably brought it to the level of the failure of the project.

**Thomas Agerbæk Ruby:** I think in terms of governance, decision making, were there anything that felt... were there anything where you were standing and thinking, "Why is it like this?" where you were not involved in the decision and then suddenly you're just told, "Now we do it

like this," or, "Now things change." Yeah. Do you feel like the communication were off in that way?

**Jesper:** Yes. I think there are a few different things to this point. But some example, or one example, is that we would have top management requesting solutions or even communicating solutions to customers or saying that "You will get this or that," that you can choose between this or that. And sending an email out to the customer, the customer would respond. But then we would have to build it, and we were already under a lot of pressure trying to just maintain the internal support. And so, yeah, I think there were a lot of decisions that were made that wasn't... that that could have been handled differently. And if it were handled differently, it would make our lives easier and probably everyone's lives, including customers. And then in terms of the larger solution, yeah, I think there were a lot of decisions that probably would have been questioned, but I also think that if [it] were initiated together with builders and engineers, I think a lot of the issues that we actually found would have been resolved, or we would have pushed a lot of the logic outside Zuora instead of having it in Zuora.

**Thomas Agerbæk Ruby:** So, if you were to start this project today, and you were in the team from the beginning, how would you handle it? What would you have done differently? Yeah.

**Jesper:** Yeah, I would have a large buy-in from management, making sure that they would bring all the necessary stakeholders in the room and have them allocated resources to this project, and that would include everyone. So, I think even like talking to the consultants, driving the project and requirement gathering, all that stuff doesn't have to be a massive team, but you should have a team in-house which are involved and knows the changes going to happen. And if there are workshops that touch on finance, there should be a stakeholder from finance involved. If it's like... engineers should always be involved probably in all meetings because there's always talks about integrations and where do we want to keep this business logic or whatever. So, make sure that there's commitment from management. We identify stakeholders, and at least one stakeholder from each team is the owner for this project in their area. And just make sure that people are onboarded. And that will probably be step number one. Yeah. And then you would go into these workshops and all that stuff. I think that's important.

**Thomas Agerbæk Ruby:** I think what you mentioned before, that finance were not involved. I know engineers in the team that were handling this in business IT were not involved. It was only driven by a few number of people, and so it's very difficult for them to think of everything and make the right decisions. Even... I have two questions that I wanted to [ask]. First is, would you still use external consultants?

**Jesper:** I mean, in general, yes. I think external consultants, depending on who they are, but I think external consultants helping you implementing a new system would normally have a lot of experience that you wouldn't have in-house. So, I think they can come with a lot of best

practices and knowledge of how to solve certain issues and cope with... because Veo knows how Shopify was run and how that solution works, so they would obviously move towards that solution because they're already in that mindset, and I think external consultants can come in and break the mindset and tell them that this is how it should work or "I would recommend this approach," etc. But it should never be them accepting requirements and be the owner of making sure that requirement is solved. I think it should be owned by the business, and they could be the one solving it, but the business should be the ones validating, making sure it's built in a way that's understandable and maintainable by the business. So, they should be more in an advisory role than actually a building role. But they... and I think they can build as well, as long if there's no resources to build, then that's fine. But I think for example, Zuora workflows being a very Zuora specific thing, if you want that, then you should have employees in the business or in the company that actually learns how that works so that they can take the ownership. Once the implementation, once the project is done, you shouldn't need to reach out to consultants to make fixes. You should be able to have the knowledge in-house. And I know Zuora has the Zuora University helping you to learn this stuff, and so it's not obviously possible to learn how to deal with internal Zuora configurations, so it could be both for that configuration, building a lot of stuff, but it should be something that also the company itself can handle them or hire people. That's my opinion.

**Thomas Agerbæk Ruby:** You said you had two follow-up questions, but I cannot remember them. Is there anything you think worked surprisingly well under the project?

**Jesper:** I mean, I think this is more of post-project, really, which is when we moved to builders. So, when the project was initiated, I was not here. Even at the first when they went live, I wasn't here. Then there was a lot of firefighting, which didn't... like, we survived. I think we still hit our revenue targets and stuff. So, it's obviously not that horrible, but it didn't work in a way that we wanted it, which was still not good. I could also not do the things that I knew that Zuora could do because I was firefighting. So, it was actually only when we moved to builders where I saw the first light in the tunnel. And I think that's when it went surprisingly well, where we got a lot of resources given to us. We could allocate, and we started to get on top of all of the tickets that we had. Yeah, so I would mention the move to builders and getting new resources on board.

**Thomas Agerbæk Ruby:** What advice would you give to someone stepping into a chaotic project midway as a product manager, like I did?

**Jesper:** And it's... I think at the point we were, was... I don't know if you typically end up in such a position, but for me at least, just taking... I think the mental strain... when the complexity and the size of the project... you could even see, "I have five different issues now. This issue number two is actually pretty bad, but issue number three is even worse. So, I need to prioritize issue number three." So, this is the kind of work that you have to do all the time. And so it was not fun. It was not encouraging or anything. So, I think just day by day, and just committed to

making sure that it works. But I think my tips would literally [be to] take [it] day by day, and voice your concerns. Yeah.

**Thomas Agerbæk Ruby:** Yeah. Do you feel like you learned something from the project?

**Jesper:** I mean, personally, 100%. Professionally, I think you learned a lot around... I don't know if politics is the right word, but the company, the way the governance around the company and stuff like that, definitely. Yeah. And I think it's taking a natural turn to a better place, where we become more part of the technical team. And now we actually behave and work as a team that I think should have happened from the start.

**Thomas Agerbæk Ruby:** Do you feel like we were in business IT team back then, but do you feel like IT is part of the governance now, and those business systems are part of high-level decisions more than they were when you joined, or just before?

**Jesper:** Yes, definitely. I think because before business IT were under the operations division, and I should probably know that, but then we moved to engineering or builders under technical management. So, I think yes, definitely. Don't know, or maybe a lack of technical knowledge on the old handling management handling? Yes, part of that knowledge is introduced now.

**Thomas Agerbæk Ruby:** Yes, I think that part is introduced. And I also think that nowadays, because we have a lot more resources, with the people working in this, we have more mandate to say, because it's also not as much firefighting, but I think we have a lot more voice in terms of decision making or pushing back even. Yeah. So, I think the governance in general is definitely better, and I think we have a lot more... Yeah.

**Thomas Agerbæk Ruby:** I also have an analogy that I've thought about since I'm writing this project, and my thought is that when you are transitioning from a startup where you do the easy, dirty decisions that brings in money, to becoming more of like a mature company. I kind of look like... as every company is going to have their own Fosbury project, because this is part of ensuring you're transitioning to systems that will be able to sustain and help you do business for the rest of your lifetime, maybe. So, the way I look at it is that you are a child, or a baby was child or baby before that, then turned to a teenager, and when you're becoming a teenager, you get explosions of hormones everywhere, you get growth weird places, and a lot of back and forth. And then as a company succeed[s with] this project, then you become a grown-up, a mature person that is able to live on. Would you agree with that analogy? That what happens if you don't succeed? Then you never get away from puberty.

**Jesper:** No, I 100% agree. I think that's a very accurate way of saying it. We know also, or me joining at the time also, we know that when Veo was a startup and how it was made, sales was everything. Even when I worked here in the beginning, sales and marketing could do whatever they wanted, and whatever was pushed down was pushed out because we just need to make

sales, we get money in the business so that we can survive. And I think that's a very natural approach in the early stages. And then once you move towards more of a scale-up, I think that's very much like the teenager phase what you mentioned, where there is a lot of pain points, a lot of growth happening, where you need to employ new people, you need to make sure you have a good onboarding so people understand what the values and what we're trying to achieve. And there's a bunch of things that's happening in that stage. And I think as you grow more mature as a business, you also move away from giving all the power to sales and moving towards the backend of the business. So, then finance has more things to say because compliance becomes more [important]. Technical departments have more things to say because implementing new features or promising things to customers will impact more customers, so it needs to be done right and in a scalable way. So, it's very much that. And I think Veo is... since I joined, I think Veo has gotten a lot better from that being finance/sales driven towards having a technical understanding. So, we are definitely moving towards the more mature stage, I think. And one other thing that should probably be mentioned is implementing Veo, or even when I worked as ERP consultant, like introducing new systems, you will have pains. There's no way around it. I think we probably experienced more pains than what was necessary or if it was done right, but people will always complain about those systems, unless there's something I haven't seen. So, I think it's natural that this happens, and I think there's always a cost that you can have from a system change because there's always something that's going to fail or timeline is going to get delayed, whatever. So, yeah, I think it's natural, but I do think that this project was not handled correctly from probably the start.

**Thomas Agerbæk Ruby:** Just have a few wrap-up questions or quick questions. Just one decision you wish had been made earlier involving more stakeholders.

**Jesper:** One decision that saves...

**Thomas Agerbæk Ruby:** One tool or process. Is this where I'm at? No, could we...? Yeah. Yeah. Yeah. And we talked about it a little bit. But, biggest surprise or wow moment in the project? Positive or negative or anything. Can be whatever.

**Jesper:** I think just seeing the workflows for the first time was probably a big wow to me, just trying to understand. Because I knew Zuora, but coming to Veo, nothing was... Yeah, it was just a ton of customizations that you needed to understand, and a lot of edge cases that were built some kind of band-aid solution that you needed to understand. So, I think...

**Thomas Agerbæk Ruby:** So, it was another system pushed into...?

**Jesper:** And a lot of business requirements that should have probably been pushed back on or had dealt with a different way. And we are looking into a new Zuora version two solution now

where we are moving towards what this initial solution probably should have been to a large degree, at least.

**Thomas Agerbæk Ruby:** Good. I think that's it.

*Meeting ended after 00:35:46*

## Interview 5

**Alexander Grosse / Thomas Master Thesis Interview (Fosbury Project) - 2025/04/10 14:55 CEST -**

**Transcript**Attendees: Alexander Grosse, Thomas Agerbæk Ruby

### Transcript

**Thomas Agerbæk Ruby:** Then we can always talk about that. Nothing going on there. But yeah, I'll start the recording and if we could start a little bit about talking about just you briefly in your role and... when you joined Veo as CTO.

**Alexander Grosse:** Yeah, I joined two years ago, leading product, design, engineering. So all the disciplines which collaborate on a daily basis. And as you know, we call ourselves builders.

**Thomas Agerbæk Ruby:** Yeah. Yeah.

**Alexander Grosse:** So we're builders first, and then discipline second. Obviously, being a Veo employee is very much first, then builder, then engineer, but whatever.

**Thomas Agerbæk Ruby:** At what point did you get involved in the Fosbury project once you started at Veo?

**Alexander Grosse:** Very early. When I came in, "There's this Fosbury steering committee, you should show up." That was a few weeks after I joined.

**Thomas Agerbæk Ruby:** Yeah. What was your original or initial assessment of the situation back then?

**Alexander Grosse:** Yeah, I mean, so I'm pretty old, so I've seen quite a few things. I felt from the beginning that this was not on a good path. There was this steering committee... everybody thought we would solve everything and some slide where then all the engineers [are saying] "it's not working, it's not working, that doesn't make any sense." And all the feature development regarding plans, pricing, whatever, on hold, because Fosbury will solve everything. I've seen this before. So I had a bad feeling, to be frank, but I was not deep enough to understand it with a lot of facts. Right? This is just based on former experience.

**Thomas Agerbæk Ruby:** All right. So, sorry, just pop up here. What were your first impressions of how Veo had handled the... let's talk about, for example, the decision-making structures. And we can talk about something like governance structures in terms of the Fosbury system. Do you have any impressions of who had made the decisions back then?

**Alexander Grosse:** So one of the Lucas asked me, "Hey Alex, could you help me clarify...". Then there was some kind of decision which needed to be taken. Then I walked around, like, I found other people, and I asked, "Who can make that decision?" Everybody: "I don't know." And then I went to Michael, the former COO, and I asked him, "Who can make this decision?" He was just like, "Are you seeing me?" He made... and that was for me the final event where I thought, okay, this is not going anywhere. So decision making was completely unclear. I think it was like this monster. Nobody knew how to tame it. Nobody had or took accountability or responsibility for it. It was a pretty horrible project.

**Thomas Agerbæk Ruby:** Yeah. I've spoken with a few people already. I've spoken with Harsh, Jesper, and I've spoken with Ulver as well. Yeah, we also talked about, yeah, the project...

**Alexander Grosse:** You spoke to all. Interesting.

**Thomas Agerbæk Ruby:** ...how he felt about it and what he felt could have gone different back then. But we also talked about, when you joined, he felt like a change, having... You were quite intent on absorbing the IT business department into Builders quite quickly, where maybe some other people in the company were more hesitant on that. Can you elaborate a little bit on that?

**Alexander Grosse:** Yes. Yeah. I mean, I've never seen something like this business IT department before. It was, to my knowledge, that people before didn't want to have anything to do with money. So everybody who touches money goes into a different department, and we're working on the same product. Scaling up and down is basically impossible if you have... Øivind was more or less the only engineer the whole time, right? So how can you scale up or down, right? It's impossible. So for me, you can have it as a separate department, but maybe when there's 20, 30 people and there's an inner structure and an inner... you can change work based on priorities, right? They were completely, absolutely lost. And when they moved into Builders, or before that was Subsavers, you suddenly could see there was a time when I got green light, "Now I can do something," and then 20 people helped roughly, right? You saw a lot of people doing L3 tickets, a lot of people working on tech. And for me, it's like, I cannot believe how we can have the biggest project in company history and put it on a department which cannot scale up. Zero sense to me, right? And obviously, they were like, "I don't want this," and "Build this," and "Alex," or whatever, but I mean, it's not in the best interest of the company, right?

**Thomas Agerbæk Ruby:** No. Did you ever speak to anybody about how the project started?

**Alexander Grosse:** Yeah, I heard some stories, but they were wildly contradictory.

**Thomas Agerbæk Ruby:** Yeah, I think, yeah.

**Alexander Grosse:** So it was before my time, and I heard it was based on Z. I don't know. I would speculate now, and I...

**Thomas Agerbæk Ruby:** When I talked to Ulver, he said one of the major problems he personally thought about the project was the amount of people who touched the project, who started the project and were then laid off, or there was all this change management. All the people who started the project never saw it through to completion. So nobody really knew who started the project, why we were doing it the way we did. So that was one of the major concerns from him, and if he would have done it differently today, that was the main thing he would look at was keeping the same people or as close to the same people as possible throughout the majority of the project. Is that something you agree to?

**Alexander Grosse:** Don't. So... It wouldn't touch my top five, to be honest. I think the top things are, for me: defined scope and work in steps, not like, "We have this one release which will solve everything, and while we're at it, we also change X, Y, and Z," and it becomes this monster project. Then, staff it with people who know what they're doing. I don't want to talk badly about people, but some people were just in the wrong position. So Christian Houen... I mean, I think he's a nice guy. But he was in the wrong position, right? You cannot expect somebody with his background to more or less lead such a project. He was not set up for success. And then you obviously need to staff up with enough people. And if I take these three, that would be my top three. And if this is the same people over four years... That was nearly four years. I don't even think that's possible. Consistency is not bad, so I'm not saying this is a bad idea, but it doesn't touch my top five.

**Thomas Agerbæk Ruby:** No, basically made it a company project, or...?

**Alexander Grosse:** No.

**Thomas Agerbæk Ruby:** Okay. Did you bring any new frameworks to help stabilize and redirect the project when you joined?

**Alexander Grosse:** Builders project. I think that's the main thing I did, right? And I tried to... I asked Ditte several times, "Hey, can I give you people? I think you need more people. Should we join forces?" whatever. And then I only succeeded when really the shit hit the fan. And I basically went to Henrik first and said, "Hey, shouldn't we do something like Subsavers to solve that?" And then Ditte said, "Okay." And then you saw then a lot of people came. I mean, at that time we needed to throw people at it because you know that our situation was pretty dangerous.

**Thomas Agerbæk Ruby:** Yeah. Yeah.

**Alexander Grosse:** But there was no framework. It was more like, "Fuck, let's do it." It was way too late for frameworks.

**Thomas Agerbæk Ruby:** Did you feel that the prioritization of a project, especially from... and Henrik, did you feel there was a lack of that when you started? Prioritization for the project?

**Alexander Grosse:** To be very frank, I don't think people really understood what the project was about.

**Thomas Agerbæk Ruby:** Yeah. No.

**Alexander Grosse:** And I think if those two had understood, I mean, Henrik said afterwards, he's a very self-critical person. So he makes a mistake like every human being, but he then tries to learn from it, right? Which you cannot say about everybody. He says he always thought, "That's easy, right? That's... building a camera is hard, selling it is hard, but everybody can do that." And then I mean, to a certain extent, it's not completely wrong, but if it goes wrong, the whole company is in danger. And I don't think those two really realized, also Michael, what kind of huge and important project this is. And then they did a mistake to rush the migration because the new system was not in a state to be migrated to, which we saw. And there's countless bugs and errors, and we're still discovering things which went wrong back then. So we migrated into ruins more or less, right?

**Thomas Agerbæk Ruby:** Something that when I've talked to Jesper and Harsh and Ulver is that they kind of regret how much we relied on external consultants. For example, they described that when the project began, there was an expectation that we could take our old system and put it into new systems and it would just be plug and play and work. I think that's also one of the reasons why, for example, Henrik and a lot of people underestimated the project because it was announced as if it was going to be easy. It's going to take maybe half a year. We have external consultants who can do it for us, and then it's plug and play, and then obviously the ball rolls from there. What is your opinion on the way that this happened?

**Alexander Grosse:** I mean, for me, it's a key sign that somebody who understands software and has been through a project should have been part of that steering group. I mean, it's hilarious to think that, right? I mean, say if we had picked the right consultants... it's one thing to rely on externals, right? But then if you rely on bad externals, then you're really screwed, right? So really good externals would have raised the right topics.

**Thomas Agerbæk Ruby:** Yeah, exactly.

**Alexander Grosse:** ...that we say. Right, we need to redo a lot of things that were done in Salesforce because also sometimes consultants don't have the right context. And I wasn't involved in that, so I cannot really say. I'm now speculating again, but I imagine they got some briefing, and then they did something which could have made sense in a different context, but not in our context.

**Thomas Agerbæk Ruby:** One of the main problems was that the consultants understood how Zuora worked, but they didn't understand how Veo worked. And also when the partnership were discussed with the consultants, they actually said that they would not recommend that Veo uses this solution. Of course,

they don't say no to a project, but they felt like this was out of their water, and it was a very complex solution that probably didn't work the way Zuora was intended for. So they were even pushing back a little bit. Yeah.

**Alexander Grosse:** Yeah, I don't know about that one. So that's good, because I think all the business requirements... Because everybody came with some business requirement and said, "Okay, you want this? Okay, you want this? Okay, let's do it here." And overall, this whole thing didn't make sense anymore, right? So you need to have a person who designs a solution which is consistent and flexible and as simple as possible, which we cannot say about ours. A lot went wrong, yeah.

**Thomas Agerbæk Ruby:** Yeah, we learned a lot.

**Alexander Grosse:** It's the worst project I've ever seen.

**Thomas Agerbæk Ruby:** It's a project that nobody likes to do. But would you argue that in a high growth environment like Veo, or in any high growth environment, for example, a project like this might come up at one point or another?

**Alexander Grosse:** Yeah, likely, right? It's very, very unlikely that in the early stages you design a solution which holds for a long time. So, yes, on a lot of levels, right?

**Thomas Agerbæk Ruby:** Do you think in any way, with a different approach, that the outcome we saw could have been avoided?

**Alexander Grosse:** You mean the Fosbury outcome? Yeah, absolutely. We were very close to the worst-case outcome, very. I mean, and... I don't want to calculate how much money this cost us, not even told. Even if I exclude opportunity costs, which are very tough to calculate, the sum is pretty impressive. I would first call it differently.

**Thomas Agerbæk Ruby:** So, if you were to start the Fosbury project today, so the equivalent to a new Fosbury project today, how would you proceed? Yeah.

**Alexander Grosse:** I mean, basically what I said before, right? First, get people in. Let's take, say we should migrate again to a different subscription platform, which I don't hope so, because now we're on a monster, and that should help be with us for the rest, at least for the next 10 years. But really understand the problem, seeing can it be solved a different way. The migration should always be the last resort. It always has a lot of risks, and nobody wants to do it. It's not fun. If it nobody will give you a high five. If it works badly, you will be in deep shit. And really get some people who really understand the problem, really see if it can be solved in a different way. But if something like that needs to happen, then define steps in which the migration goes. Maybe I'm just making stuff up, maybe we shouldn't, right? We needed to change payment system because we use Shopify Pay, and Shopify Pay can only be used with Shopify. So maybe change payment systems first, does it work? Right? And then before we then migrate subscription

platform, do extensive testing. Right? We moved into the most buggy system I've ever seen. And I admire Øivind for his work on the migration. But in reality, he should have worked on the final system to make it ready, not on the migration. The migration was half a year too early. Yeah, get the right people. But the key thing is for you... get the right people in, because if you have somebody who has been through a project before, some things need to be learned. You can be really, really smart, but some things you need to experience to understand them. And there are complex projects with constantly blowing up scope, and the wrong people involved. You need to see it once at least. Could also be on a smaller scale, but the pattern...

**Thomas Agerbæk Ruby:** Yeah. Yeah. One of the issues I know that when it was the business IT team that handled the project was that they had a really difficult time to get some of the other departments on board. For example, the finance were not really interested in taking up a new project. And also, of course, at one point, Builders were also hard to... They... or...

**Alexander Grosse:** I know I talk with Jakob Frost, the former VP of engineering. Yeah. Yeah.

**Thomas Agerbæk Ruby:** ...I know I talk with Jakob Frost about what was it called first exactly? He said that he didn't want Builders to touch anything to do with business and only product.

**Alexander Grosse:** But that's how Business IT came, right? I mean, when I joined, they were involved in the steering committee. There was David, the former CFO. I think the things you're referring to were before my time.

**Thomas Agerbæk Ruby:** Exactly. How did you in any way try and involve these types of stakeholders when you joined? Or...? Okay.

**Alexander Grosse:** But I know that Frost had kind of an allergy against everything which had to do with finances or business. And...

**Thomas Agerbæk Ruby:** And did you feel like there was, when we saw the fire happening in 2024, that there was any need to rebuild any morale, trust in the company back then? For example.

**Alexander Grosse:** I... I mean, honestly, back then, trust in the company from the employees into the company...

**Thomas Agerbæk Ruby:** Yeah. Is there any...?

**Alexander Grosse:** I mean, I mean, my initial thought was just, "Let's solve this crisis." And there was a lot of work happening, right? I was sitting at 7:00 a.m. on a Sunday solving tickets. I think how you solve the trust issue is that we invest enough that we don't get into the same situation this season, like July, August, September. If we get into the same trouble again, we're really on it. And I think we need to show the organization. That's why we also shared the public post-mortem, right? That we're learning from it.

Because the old-timers told me every renewal season was a shit show, and seemingly nobody learned from it. But now we're a bigger company, we have more people, we can also invest more. And that's the key, right? If the renewal season goes smoothly, I'm happy. But there's still the big test.

**Thomas Agerbæk Ruby:** Would you describe the Fosbury system as a transformation, or a move from startup chaos to maybe more like a mature setup, governance in general? Wait a minute.

**Alexander Grosse:** I mean, I have a hard time associating anything Fosbury with mature or professional. It might have been an attempt at that. And I think in reality, it's right because we cannot build these systems ourselves. Maybe we can, I don't know, but I don't think it was a stupid decision to do it. How it was done was the problem.

**Thomas Agerbæk Ruby:** Yeah. An analogy I've had in my head since I started working on the project is that I've seen this before, the transition into these new systems. We could see Veo as more like, of course, a younger company, and therefore also maybe a child. And then where things are... they're not easy, but they're a bit similar because the business cases are different. Back in the old systems, the first thing they tried to solve was, "Okay, how do we sell to one camera to one person?" Then suddenly, "Shit, that person wants to buy another camera! How do we see what they have already?" So suddenly, more and more complex business rules, could you say, came up. And then [the] system is kind of... the Fosbury project is kind of like becoming a teenager. Everything gets a little bit ugly. You're growing in different ways. You get pain points in different areas. And once you've completed the Fosbury project and transitioned into these very large, scalable systems that we can use for a number of years, you have matured a lot. But you agree in that analogy?

**Alexander Grosse:** You don't bother, at least in a company this size...

**Thomas Agerbæk Ruby:** All right, let's see what haven't we talked about. So I don't know if this is something we've already talked about, but you mentioned you've never seen a business IT department before.

**Alexander Grosse:** I've seen it in Nokia or something, right? But not at the state of the company.

**Thomas Agerbæk Ruby:** Did you feel like your initial thought was that this is an unnecessary department for where we are? Yeah. Yeah.

**Alexander Grosse:** Again, because you need to have a certain size to be able to scale up and down. And I mean, you're probably aware that a few people from business IT moved to Builders initially, Isak and Christoffer and David. But they basically didn't rehire any engineers, which is still a miracle to me, right? So they had Harsh, and then Harsh had it and also moved to Builders, but also didn't rehire for him. Then Øivind was on his own. Poor Øivind. Yeah, a lot of management mistakes.

**Thomas Agerbæk Ruby:** I think at one point it was Øivind and then it was me and then Jesper. Arena was there as well. But it was a hectic period, to say the least, back then.

**Alexander Grosse:** Yeah, but sorry that I didn't mention you, but I mean...

**Thomas Agerbæk Ruby:** No, it's okay.

**Alexander Grosse:** ...you're part-time, right?

**Thomas Agerbæk Ruby:** Yeah. Yeah. But for me, of course, when I was part of a smaller team, it was very interesting because there were so many hats to put on all the time. So for me as a part-time student, I learned incredibly much back then from all different... Of course, the situation was not ideal, but for me it was an exciting time. Not having any responsibility and just learning...

**Alexander Grosse:** I mean, I'm pretty sure it was also an exciting time for Øivind, as an extreme case...

**Thomas Agerbæk Ruby:** ...how to do.

**Alexander Grosse:** ...but it shouldn't have been this way.

**Thomas Agerbæk Ruby:** Is there any myths or any things about the project you think can be corrected, or do you think that when you look at it now, is it a success where we are right now?

**Alexander Grosse:** No. I mean, one story I heard is that the initial promise was to make sales, the life of the commercial team, better, and... I think maybe we are now on par with... I don't know exactly, right? But we certainly haven't made their life dramatically better, and then after three years... I cannot... That was a failure, the project, by the way. We're still cleaning up from it.

**Thomas Agerbæk Ruby:** Right. Let me see. A quick question. If you were to advise another CTO at another fast growing company facing a similar transformation project, what are your top two, three lessons?

**Alexander Grosse:** I think I mentioned that before, the scope, right? Enough people. Don't underestimate it.

**Thomas Agerbæk Ruby:** Yeah, I think that's also what I hear from everybody else I've talked to, and that seems to be the primary learning from everybody involved in the project.

**Alexander Grosse:** Yeah. I mean, these big projects also have this pattern. "Why are we at it? Shouldn't we change X?" And the answer to that should always be no, because then scope blows up, complexity goes up, and the thing becomes undoable. That's good.

**Thomas Agerbæk Ruby:** All right. I don't know if I have much more on my table.

**Alexander Grosse:** That's good. I can fill the rest of the 30 minutes with other work.

**Thomas Agerbæk Ruby:** Yeah, that's good. But thank you for your time, Alex. It was a pleasure.

**Alexander Grosse:** So, yeah, thanks for surviving this project.

**Thomas Agerbæk Ruby:** Yeah, it's fun. It's fun from an academic point of view, because there are all these best-case uses and we have completely ignored everything in this project. So there is a lot of learning, especially from an academic point of view as well, and there's also a big gap in these types of projects.

**Alexander Grosse:** If you study basic project management, these are the points. I don't think anything new came out of the project. The lesson from this is: ignoring the fundamentals is not a good idea.

**Thomas Agerbæk Ruby:** Page one. Yeah. Yeah. But it is very interesting because there is a gap in high growth environments, in scale-up companies, in terms of academic leadership. So it's definitely something new to bring to the table. Please. All right. But thank you, Alex. Nice talking with you.

*Meeting ended after 00:29:51*

## Interview 6

Timo / Thomas - Subscription Factory Master's Thesis Interview - 2025/04/30 09:28 CEST -  
Transcript

### Attendees

Thomas Agerbæk Ruby, Timo Zuidgeest

### Transcript

Thomas Agerbæk Ruby: There we go.

Timo Zuidgeest: I think it's good to do retro perspective.

Timo Zuidgeest: By the way, I've got a hard stop at 10. I think you as well.

Thomas Agerbæk Ruby: Yeah, that's perfect. Yeah. Yeah.

Timo Zuidgeest: Thank you for sharing the questions. I answered them already because that's something I like, otherwise it's going to be too much open conversation and I think it were pretty important or really valid questions that you were raising. And secondly, so what I said as well is that we never actually closed the project. It was more or less just a candle that stopped burning at some stage. And I think more generally, it was a bit rough this project. Really bumpy from everybody's side.

Thomas Agerbæk Ruby: Yeah. Mhm.

Timo Zuidgeest: I think from Zora's side, from our side, but also I think from a vio side and I think this is also reflecting I think the answers that I will share with you to start with more generally I think a project is always something that you're doing together so it's not a vendor's responsibility to deliver of course it is in the hard deliverables but at the end doing a project foss that's a teaming effort and from a subscription perspective at the end it doesn't really matter if you're a vendor customer business

Timo Zuidgeest: it, finance, we don't care, you have to do it as a team because whenever you making the hard borders between the different companies,...

Thomas Agerbæk Ruby: Yeah, thank you.

Timo Zuidgeest: then it will never work. But I think that went better. So how would you like to so I've got your question for me so I can just read them up and give you some explanation and of course when you've got any questions in between.

So how did subscription factory get involved with the Fosbury project at Veo? I think of course there was a need to streamline recurring businesses. I know Veo contacted Zuora as an vendor of subscription billing solutions. we are in Zuora partner for more than a decade already. We've done 65 70 projects in the Nordics in Europe in MIA mainland also in thou in US. So we were pretty soon involved as well in this possible project.

Thomas Agerbæk Ruby: Yeah.

Timo Zuidgeest: I think it was a short list between Zora and Salesforce billing.

Timo Zuidgeest: I think that what it was and I think at the Veo decided to go for Zuora and some other partners were involved as well. There was a partner also on the finance side. I think maybe they still working for you. I'm not really I cannot really remember his name anymore. And I think also Dyna Cap they were also involved for doing the Salesforce part the integration between Zora and Salesforce. So it was a starting point by Zuora and the way it works sometimes that Zuora is prime they said okay we would love to go for the project and...

Timo Zuidgeest: subscription factory help us will support us that's our partner actually to get things done that's where it started correctly correct yeah actually yeah they were running their recurring business I think it was a Shopify and some

Thomas Agerbæk Ruby: Mhm. ...

Thomas Agerbæk Ruby: so it was Veo who contacted Zuora and then you found a common ground. Okay.

Timo Zuideest: They were running their current business and i think it was Shopify and some other environments and the main goal was, well it was a big project it was not only about recurring billing that was one but also I think it was also meanwhile also more or less a salesforce project...

Thomas Agerbæk Ruby: Yeah. Yeah. Yeah.

Timo Zuideest: because one of the main things was also to streamline the sales organization again with CPQ quoting based on rules engines quoting based on all kinds of other things so it was a bigger project than only Zuora at the end wasn't a full redesign of the end to end stack. Maybe it was too big !

Timo Zuideest: Think at the end maybe there's too much to do everything in run maybe the first focus should be more on Salesforce and the first focus should be more in Zuora recurring billing packaging and so on fulfillment and then the next step was maybe to start then with the salesforce project it was everything and everything and everything together so it was quite a big project. So what was the original scope or mandate? The original scope of the project was to redesign. I said before you know it was to redesign the full end to end process. Yeah, it was combined single art for sales with recurring digital services. So that was really the part for So the cameras and also the services package it in such a way and also to be supported by a number of new systems and salesforce.com was one of them.

Timo Zuideest: and Zuora was one of them and everything should also be integrated with existing legacy applications your finance but also everything on the Zuora side and on and on the fulfillment side and yeah what I said before you know that the salesforce side and the sales support part that was already in product as such that was already a big project so it was a lot that had to be managed by Ulver and by Christian and that

Timo Zuideest: time and by the others as well. It was a lot of everything and...

Thomas Agerbæk Ruby: Yeah. Heat.

Timo Zuideest: maybe I think it was too much at the end. Did subscription.. So my check out on that one is that not particularly Veo but whenever you start with something like Zora that is already a big project because it needs involvement from sales for packaging and pricing for nurturing for acquiring sales for finance revenue recognition things finance fulfillment everything and everything.

00:05:00

Timo Zuideest: So even when you start implementing something like Zora that is already a big cookie that you need to eat and then whenever you are also including a redesigning for sales process then it's going to be too much I think and I think that's my check out of this project as well maybe it was a bit too much of everything in the early days. So did subscription factory use a standard implementation playbook

or was it tailored by Veo needs? I think it was more or less tailored by Veo needs the only thing is that because we normally as subscription

**Timo Zuidgeest:** factory we use a standard playbook and that's a playbook which is more or less based on Zuora nine keys that's product pricing acquiring nurturing cash collection building subscription management blah blah blah that are normally the nine keys that we touch whenever we are doing Zuora implementation. I think that was clear from day one that was also our starting point, however Veo yeah but the thing is before we started the project not everything was pretty clear yet it was more or less: We know where we would like to get Zora implemented. We would like to get Salesforce processes redesigned but let's go for a more agile approach. and agile actually killed us almost. I think everybody because lots of requirements were not clear from day one. ...

**Timo Zuidgeest:** It was step by step. So, it was a bit conflicting that normally we run the nine keys approach they Veo choose for the more agile approach and it was a more mixed up playbook, of two methodologies.

**Thomas Agerbæk Ruby:** So from your understanding was Veo able to see the end goal of this project when they started it or...

**Thomas Agerbæk Ruby:** was it more like we'll take it down the road and we'll learn and do. Yeah.

**Timo Zuidgeest:** In big words I think it definitely was that we need to streamline, we need to make our subscription billing and our subscription management processes more efficient.

Yes, we would like to have a system in place where we can easily also on-board archetypes, and other other propositions. And yes, we need to give the sales people all the necessary tooling, to be most effective, in their sales processes. So, that was more or less the bigger picture. And I think it was pretty faced out clearly. There was a phase one and a phase two and a phase three. I think all the decks were there. It was all good.

**Timo Zuidgeest:** But the thing was that whenever you were starting whenever you making a website or a commerce site then agile is fine because every two weeks you define your sprint and what your deliverables will be and you include your business and your stakeholders and you can have some chitchats about would you like to do more this way or that way more this way or that way but whenever you're implementing the full end process then it's not a matter of being agile because with agile you can go to the left you can go to the right you can go backwards forwards and so on and so forwards. Especially projects like this, you should do more a mixture approach. I think a more waterfall agile approach before you start doing any coding, you have to be pretty sure about your end to end processes. And that was not really the case. it was more or less okay, let's go for best practice. Let's go for what subscription factory thinks what needs to be done. let's go and by step. Actually, yeah, we were just following a certain journey.

Timo Zuidgeest: But I think looking back and I think I discussed with Christian & Ulver maybe we had to spend more time on definition how does it look like to implement the most effective fulfillment process how does it look because also things like product catalog the offerings as such it took I think at least five or six months before Veo decided finally how their market propositions should look like how the new bundling should look like so not everything was pretty clear from day one and that's not a problem but you need to recognize it as such so that impact also your timelines

Thomas Agerbæk Ruby: So when you were having these meetings with Ulver and...

Thomas Agerbæk Ruby: Christian, those were described as by my co-workers non-technical people. Do you feel like there was a need for more technical people in the room when we were doing these meetings? And

Timo Zuidgeest: I think there was a need for less technical people...

Timo Zuidgeest: because normally when you run a project, you run a project for finance, you run a project for business or you run a project for IT. It was definitely an IT project was an IT delegated project. So it I think received from the board guys we have to do something with salesforce and Zuora please handle it.

Thomas Agerbæk Ruby: Yeah.

Timo Zuidgeest: So our conversations with business with finance but also with sales were really limited so every time we had questions about okay what about the propositions what about your office yes we need to ask that we need to ask that yes we need to talk to finance yes we need to talk to what's her name beta I think beta she was from Sales i think, i don't know if she is still there but so I think that was also difficult for Ulver and Christian they were a bit squeezed between us vendors and also between business

00:10:00

Timo Zuidgeest: because they had one thing they wanted to do everything well for everybody...

Thomas Agerbæk Ruby: So there was a lack of internal collaboration on the project

Timo Zuidgeest: I think so yeah, but they were not really able I think also to bring in the business people into conversations directly with us or to bring in the finest people in conversations directly with us they were always being a bit of a proxy about I think so yeah I think so it was also a bit of a fear culture I think they didn't want to go back too much to business because maybe business guys we delegate this product to you so you have to run it. I think they had some hard times as well, it was difficult for them I think sometimes as well. so normally when you're running a project like this, it should be a multi-disciplinary project, from day one till go live, you should have every meeting where needed,

Timo Zuidgeest: Yeah, always the same people from finance, from marketing, from logistics, from I don't know who, and it was always Ulver and Christian and harsh and Rasmus and...

Timo Zuidgeest: Øivind that was our working space and normally we would love to talk with people from everywhere,

Thomas Agerbæk Ruby: Yeah. Mhm.

Thomas Agerbæk Ruby: So, was it hard getting the definitions of what was needed from Veo?

Timo Zuidgeest: It was not hard to get the first definitions from Veo because that was more or less clear. But I think the devil was always in the details,...

Timo Zuidgeest: You know, people started to think about everything when stuff was more or less implemented, especially when it was about, for example, designing the product catalog that wasn't ping pong, that wasn't an forward backwards between us and Ulver and Christian and the team.

Timo Zuidgeest: and then build for Ulver and Christians team I think also with sales again and it even stopped the project I think for a few months because Veo was not really able actually to get the full market propositions on paper. it took a while before everything became clear and the thing in this one as well as well that sometimes when we are running projects for other customers and we are facing that product catalog is getting too complex then we talk to the business as well hey guys I'm hearing what you want I see what you want but you decided to go for Zora at the end zora can do a lot...

Timo Zuidgeest: but don't make it over complicated in everything and everything.

Timo Zuidgeest: So I think in some areas and that's also our mistake of course or mistake that's also something that we maybe had to do differently that we had also to make more noise to say guys we fully understand the direction you're going to but keep it simple for day one Yeah,...

Thomas Agerbæk Ruby: Do you think there was a lack of understanding from view side of the systems they wanted...

Thomas Agerbæk Ruby: because What I know is that they were expecting a plugandplay solution that matched their old system and then we can just integrate Zora and Salesfor we can use it the same way and they try to force the old system setup into the new systems.

Timo Zuidgeest: correct. Yeah.

Thomas Agerbæk Ruby: Yeah. No.

Timo Zuidgeest: And not only system functionality also procedurials, procedures and way of working but also interaction with customers and also things like and maybe it's also my answer.

Timo Zuidgeest: So it was also about FO's fear there was always a sequence before somebody signs up then first we need to release a proforma invoice and then at least the customer understands what kind of

invoice they get that was the real invoice then the second thing is that there should be an incoming payment and then that incoming payment will be well kept and draw reconciliation with the performing invoice then the ordering process starts and then when final delivery has been done and when the final payment has been done then the final delivery will be done. that were quite some not best practice steps. There was always a bit of a fear with FO that they were shipping out equipments that before there was any payment done by the customer.

**Timo Zuideest:** And that's a valid argument, but you have to find the balance between, what is still doable in terms of procedures and your administrative organization and what makes it really complex. And that was difficult sometimes maybe a bit more pragmatic approach could help as well. But again, that was us between and Christian and the guys. We never spoke with business on that one. The only thing what tried to do was he understood from business what they wanted and he said okay what you want we will deliver what you want we will deliver what you want we will deliver but there was never a good conversation with business guys what you want maybe we can do it a bit more easy in the first phase just to get things up and running it was indeed more or less getting the ideal or maybe a bit overdesigned solution into Zora and yeah again I think we also had to make a bit

00:15:00

**Timo Zuideest:** more noise. We in Zor say, "Hey guys, maybe we should go back to the drawing table and say yes, we definitely understand what you want, but yeah, let's go for more pragmatic solution from day one at least to get Zora and..."

**Timo Zuideest:** Salesforce up and running and first practice and then optimize, then when everything is running and everything is going fine, then you can decide, okay, maybe these are the different areas, yeah, where we have to do a bit more automation or a bit more workflowing and so on and so on. No, just mainly over.

**Thomas Agerbæk Ruby:** Was there a point in time...

**Thomas Agerbæk Ruby:** where you talked to other departments at all or anybody else were involved other than Ulva?

**Timo Zuideest:** Yeah, mainly over Christian Ramus and Øivind and by the way, operational was we had a great relationship with these guys. They went to our office. We went there. from an operational perspective, I think it was great teaming.

**Timo Zuideest:** However, from a more strategic it was sometimes a bit unclear that we thought guys what we're doing right now is it really the way how we should forward and then wanted to please I think everybody and everybody he worked hard and he was involved not in the details more or less he said not really into the details but yeah I think that sometimes went there was not really a steering committee there was no steering committee between us or...

Timo Zuideest: and also Veo so we were not really talking high level we were only talking with high level management like Ditte for example. Just when something went wrong, or Yeah.

Thomas Agerbæk Ruby: And did you ever get an impression of...

Thomas Agerbæk Ruby: how much did you prioritize this project? was that something you ever thought about or

Timo Zuideest: She was managing on KPIs, she was mainly managing not even really on budgets, I think, but at the end of course as well, but she was also managing on KP personal KPIs.

Thomas Agerbæk Ruby: And did you feel that was that helpful that approach the Mhm.

Timo Zuideest: Guys, I need to get this life. it's my personal thing to get this life. it was pushing just to get things live. It's always helpful at the end, it's always helpful. Somebody should push it. but I think at the end we were all captured by the situation we were in we were halfway or 3/4 and we had to finalize things we were facing complexity in some areas and I think mainly also fulfillment that was a big thing the fulfillment workflows and everything to get everything shipped out correctly but also the edge cases

Timo Zuideest: So they were also really focusing on edge cases as well. what happens and then the question was okay how many times does occur monthly? Yeah a few times. So yeah it was also sometimes I think shining the light more on edge cases instead of shining the light on the commerce business on the happy flows and...

Timo Zuideest: as business as usual. I think at the end also it was also ongoing business. No, I think the video cameras they upgraded from a new version. That was also something of course that was impacting I think we had to do some migration again. ordered people were signed off or people had ordered an old version but the new version should be delivered. But it was also of course yeah business thing Yeah. As long when the project takes longer and...

Thomas Agerbæk Ruby: Yeah. ...

Timo Zuideest: longer you facing all kinds of business challenges as well during the project but that's fine I think. Yeah.

Thomas Agerbæk Ruby: if you could go back to the start of the project,...

Thomas Agerbæk Ruby: is there anything you would have done differently recommended

Timo Zuideest: So going back to the start of the project my recommendation should be guys before you start doing any coding and...

Timo Zuideest: whenever you're doing a project this don't make it too agile first of all just sit describe make your flows make your designs get everybody on board how would you like to set up your quoting

process your bundling of your products first of all more thinking in the early days and then execution after that it was now a little bit thinking start execution and...

Timo Zuideest: during execution still a lot of thinking and that's what you need to do differently I think in projects like this

Thomas Agerbæk Ruby: do you think com when you compare this project to anything else you've ever been involved in is there anything that stands out?

Timo Zuideest: You mean doing it differently? Yeah, that's what you mean.

00:20:00

Thomas Agerbæk Ruby: Is there anything about the first project when you think back is there anything that stands out? Is there anything that was done differently than you usually do?

Thomas Agerbæk Ruby: Is there anything because of the complexity of the project that you feel like this is not the usual case or is this quite a usual case for you guys to work on?

Timo Zuideest: This is quite a usual case.

Timo Zuideest: but what normally is different with some other customers is that C-level management is more involved and whenever some hard decisions should be taken about getting a project up and running then also C-level management stands up and they say guys stop talking about this complex finance process we're doing it this and this and this way guys I know that fulfillment is important but can we not simplify it in such a thing by not try to automate everything but also to use the human also to implement the human factor, with some procedural stuff and work and also some other business rules that you have to do internally and not everything should be done automated by any kind of system.

Timo Zuideest: That normally helps when a bit of C-level pragmatism is coming in as well to say guys okay stop stop this is...

Timo Zuideest: what we're doing first second third don't try to automate everything in every different edge case scenario because that will make the project more or less unantless what they should be involved...

Thomas Agerbæk Ruby: Mhm. So there was a lack of prioritiz prioritization from seuite level employees.

Thomas Agerbæk Ruby: Do you feel like that is that normally the case that they are also involved?

Timo Zuideest: because implementing Zora that's open heart surgery you are working on something that hits the full organization because it is about subscriptions it is about your revenue it's about your customer satisfaction it's about everything so that's something that I was really missing

Timo Zuideest: somebody that sometimes could make the final decision stop this we going to do this way and that could always and that was also and that's I think something that was missing as well being backed by his board to say okay let's go that way yeah correct and...

Thomas Agerbæk Ruby: Do you think that Veo as an organization were not fully grasping the size of the project and...

Thomas Agerbæk Ruby: what it meant to have a project like this going on? Yeah.

Timo Zuideest: no worries it wasn't difficult

Timo Zuideest: project we also had to do something better maybe making a bit more noise tell them guys do that workflow differently do it otherwise and also unfortunately we were also facing some limitations in Zora at that time because the selling process of the camera right now is we've got ordering functionality was also in Zora in the early days we started the project with vio however ordering functionality was not integrated with the CPQ functionality yet.

Timo Zuideest: So we couldn't really go for the line. So we had to do it in a different way. so we were also facing some things that were not really really great I think from day one.

Thomas Agerbæk Ruby: Okay.

Timo Zuideest: And I know we had to do the fulfillment workflow twice because it was really a monster of a process at the end but if then da da da.

Timo Zuideest: I'm not sure if you have ever seen these design documents, but maybe Jora workflow was not the best solution for doing the final fulfillment afterwards.

Thomas Agerbæk Ruby: All Is there anything you feel like is unsaid or anything you want to add?

Timo Zuideest: So okay just so based on experience what would you have thought growing startup start quick go live.

Timo Zuideest: So, normally I think when you're doing a project like this again, try to get something up and running quickly, as soon as possible and take it from there. Don't make it too big in the first phase. I said,...

Thomas Agerbæk Ruby: Mhm.

Timo Zuideest: we discussed already a bit more contact with business would be really helpful, I think. But at the end, we all worked hard, it was really good that we have a great relationship with Ulver, Øivind, and also with Harsh, because the teams worked really close together to get things done.

Timo Zuideest: No I said was really bumping roads fulfillment that wasn't the biggest issue I think for us at the end I think that's also something that we could do I think better in a more efficient way I think with

the team just to tell team guys stop it doing this way I think as subscription facing resour I think we were more or...

Timo Zuideest: we were more or less following Veo too much in their way of thinking I think we had to make more noise, about guys simple, keep it simple. I think we all made it a bit too over complicated in some areas. That's my lesson learned as well for this.

00:25:00

Thomas Agerbæk Ruby: And did you feel like you spent a lot of or...

Thomas Agerbæk Ruby: some time also understanding what is Veo as a business because there are some unusual things with having software. What do we sell, what invoices do we generate first? Yeah.

Timo Zuideest: That was difficult.

Timo Zuideest: That was the difficulty I think as well. Yeah, that's what I said. what do the whistle first? Do we need to send out a proforma invoice first and blah blah blah. So the business flows were doable. but sometimes even when you're facing a difficult business process then it would be great that somebody stands up guys can we please make it a bit less complex and...

Thomas Agerbæk Ruby: Yes.

Timo Zuideest: just do it that way what are the risks of making things more simple customers are vos customer you always need to think about what whenever you're running a project like this also replace or place yourself in the customer experience what they are experiencing and don't

Timo Zuideest: try to fill in all the requirements that you as a Veo thinks that you think how customers are thinking you have to sometimes step out and think okay whenever we are doing it this way that way what does it imply for the customers not really something okay fine do it that way and...

Timo Zuideest: what's always people always would like to fill in customer needs themselves instead of asking them Okay.

Thomas Agerbæk Ruby: Yeah. All right,...

Thomas Agerbæk Ruby: Timo, I don't want to take up any more of your time, but it has been very very interesting to talk to you and...

Timo Zuideest: Yeah. I've got the answers also in text,...

Thomas Agerbæk Ruby: Yeah,...

Timo Zuideest: so I will send it to you late later on today.

Thomas Agerbæk Ruby: that's perfect.

Timo Zuideest: Yeah. Yeah.

Thomas Agerbæk Ruby: It's been very valuable for me to also get good to get your perspective. I think that's something that will be really useful for my thesis.

Timo Zuideest: Take care, Speak to you. Whenever you got any questions left, let me know. Yeah. Okay.

Thomas Agerbæk Ruby: We'll do. Thank you for your time. Bye.

Timo Zuideest: Thank you. Bye-bye.

Meeting ended after 00:27:12

## Interview 7

### Øivind / Thomas - Zuora 2.0 Thesis Interview - 2025/04/30 14:27 CEST - Transcript

Attendees: Thomas Agerbæk Ruby, Øivind Larsson

#### Transcript

**Thomas Agerbæk Ruby:** So, I wanted to rewind a little bit as well. So, not only talk about Zuora, but also a bit about the restructuring and absorption time and the Subsavers time and all those crazy experiences you had from that time. So before we dive into Zuora 2.0...

**Thomas Agerbæk Ruby:** I want to talk a little bit about the period right after Business IT got absorbed by Builders. It was around September, October 2024, around there. The team gets absorbed. Can you walk me through a little bit about what started happening at that time?

**Øivind:** Yeah, so that's a good one. So, the first thing was that we basically got pulled into a meeting with, I know at least Alexander was there, more, probably also Sander, where they just told us that you've now been absorbed by Builders and the old structure no longer exists. Before that, we had already got notice that Ulver was on leave anyway. I think it was within his last two or three months at that time anyway, but he was let go earlier that day. After this, we were basically just told that we would move into Builders and we don't really have a plan for how you will fit into Builders yet, but we really need to get more people on board because we see you as very business-critical and we need more people on it. So after that, we basically went two, three months without having any proper team in place. It just started slowly forming in the background. We were still kind of the remaining of Business IT on the other side where

Subsavers was pulled in. So Subsavers was kind of the structure of that time, but when Subsavers was supposed to end, we didn't really know exactly where we were going.

**Thomas Agerbæk Ruby:** So would you describe it as it was a period where there was little to no clues about what was going to happen? Was there anyone whom you were reporting to at the time, or was it more just like waiting and seeing when somebody tells us news?

**Øivind:** So I don't remember if it was everyone, but I think most of the people that moved from Business IT got Alexander Grosse as manager. So I had one-on-ones with him until the end of the year, and that's when we swapped to getting Gert as my manager. So that was the closest manager, but then again, he has a lot of other things to do as well. So it was just this one-on-one and asking how things are going. It wasn't that direct follow-up that you can have, but if you're actually sitting beside someone and they're in your team, only working on stuff that you're working on. But we also had Subsavers, which is where, for example, Gert was the first one more or less that onboarded. Gert and some others, I remember I had a lot of meetings and very close collaboration with them in these days, and then we started this whole Subsavers service thing. I was pulling everybody into a room and then we started to actually teach everybody what the systems were all about.

**Thomas Agerbæk Ruby:** Exactly. So that... I want to talk about that. So how did Subsavers come about back then? Do you know? Was it Sander who started it?

**Øivind:** Yeah. So I believe it was an initiative from Sander, or at least from Builders, but Sander was the main character that we saw.

**Thomas Agerbæk Ruby:** What was your goal of Subsavers?

**Øivind:** Subsavers is still on, but phase one of Subsavers was to basically get ahead of the subscription ticket backlog, the support backlog, and to ensure that we get to a stable state where we can actually work on the system that we have. So that was all about breaking the backlog, or whatever you want to call it, but get those ticket numbers down, get it into a state where we control the inflow, we are able to respond within one day, whatever it was.

**Thomas Agerbæk Ruby:** And then you and Jesper were doing these internal training sessions with the engineers of Builders. How can you tell me a little bit more about those sessions?

**Øivind:** Yeah. So, yeah, after going to Builders, we were basically just told that, we're assembling a team, Subsavers, and you're going to teach them how to do this. So, we kind of just thrown into it. I remember Jesper and I sat together. It was like, "Yeah, what should we show them? I don't know. Let's just smash something together," and I think we'll just go along and the questions, they will drive the sessions. So the first two sessions, we actually had a little plan about what we wanted to do. After that, we ended up just doing in plenum two actual L3 tickets that the other ones had, where they got stuck, and then we did that all together. And for a while that worked, but then that stopped working because, I think, we were a group of 12 people at some point. People learn at different speeds. If we want to do every ticket and do it

thoroughly, then we had half an hour and we did three tickets. Some people were way ahead of that. Gert, he was already doing tickets on his own, whereas we had other ones that really weren't following along. So we had a very high gap, you could say, in the ones that weren't paying as much attention and getting the point across versus the ones that were really digging into this and taking it in full seriousness.

**Thomas Agerbæk Ruby:** And it was mainly just to get it on record as well. It was just for teaching about our business systems and Salesforce. Only about the acquisition flow, Salesforce to Zuora. Yeah. And everything that can go wrong in that period. And what impact do you think that had, both short-term and long-term?

**Øivind:** Short-term, huge, because we got extremely smart people into a room and we showed them how messed up the solution was, and everybody was mind blown by the amount of complexity and the amount of stupid things that we had to keep in our head just to maneuver in the wilderness, you could say, that we had. So short-term, we got a lot of traction on this. This is the priority, we need to fix this, this can't keep going. And we were able to showcase that to basically the whole Builders organization. Long-term, I think, yeah, this is both short-term and long-term, right, because long-term we're getting the buy-in to actually do it right now, which I don't think we would have if we weren't able to show Sander and Alexander Grosse about what are our actual problems, because they were also solving some of these problems during this phase.

**Thomas Agerbæk Ruby:** Yeah. And if we go back to the Business IT team at the absorption, what happened to the colleagues or the team members of the team? Did they get spread out to different teams, as far as you remember?

**Øivind:** That, that's what happened, right? Yeah, after a while. In the start, we were basically the same team with René trying to hold us together, but he had no clue what was going on at that point. And it also ended up with him also leaving the company because we didn't fit in, couldn't fit him in. And then basically the split was that Irena went to a different team, and me, Zsolt and Jesper stayed in one team, which was the split. Yeah, that was basically it. And then, of course, you and Fraser and Federica, which was supposed to join Business IT, kind of formed their own unit. So there was a while where the IT team were hanging in a state of uncertainty, and then there were three new teams coming into Builders where the different team members of Business IT got placed, right? So what was that? There was the one you're in now, which is called Revenue Platform, and then Business Optimization. I'm not sure what the L3 team is.

**Thomas Agerbæk Ruby Ruby:** It's called Federica.

**Øivind:** Oh, okay. So, it's called Federica's team.

**Thomas Agerbæk Ruby:** What are the Revenue Platform and Business Optimization teams, what are their responsibilities now?

**Øivind:** So, the Revenue Platform's main responsibility is Zuora and the billing system and then everything that comes with it. So you have Avalara, tax calculation, and you have Adyen for capturing payments. The main thing was that Monetization used to own Adyen, but we got Adyen because it's coupled with Zuora after Revenue Platform became a team. Business Optimization is working more with Salesforce and serving support, i.e., internal users. So they build support functionality in Zuora. They handle everything related to strategic accounts in Salesforce, and they ensure that all of those processes are in place. Which was previously something Business IT had to do as well as the Zuora. And of course, L3...

**Thomas Agerbæk Ruby:** ...which team?

**Øivind:** Oh, the third team. Yeah, the third team is the one that handles the L3 support, which I was talking about before. So they are the ones that are taking care of all the inflow of tickets and ensuring that we get through those. So they are kind of the frontline towards support. And then we have the two teams that are building on the system, where Revenue Platform is doing the core subscription engine and the billing system, and Business Optimization is doing the sales-facing functionality.

**Thomas Agerbæk Ruby:** So would you say that the new setup is more efficient in what you can produce or what you're doing?

**Øivind:** Yeah, this was actually my proposition that I gave to Alex. I remember whether he figured it out himself before me or not, I don't know, but I remember actually pitching this to him. But I think it makes sense to have two teams that serve the end users, which is Monetization for our commerce customers, sales, service, and Business Optimization for internal users, being sales, being support and the rest. And then kind of on top of that, we have a team that's fully dedicated to the subscription engine and the payment solution, so that's what they're only focusing about. And that revenue team, which is heavily dependent on the other teams, and vice versa. Commerce without us, the organization can't work without us. They're kind of bundling that together, and then having those ones that do the actual surfacing of the features that we build. So those teams are kind of representing everything that Business IT was doing before. Yes.

**Thomas Agerbæk Ruby:** And now would you say it's more controlled now? And is it under control?

**Øivind:** Compared to last period, it's controlled. If you think of support, then it's under control. We don't have a huge backlog of anything. Like, things are going smoothly. If you think about our willingness and possibilities to implement new features, that is not under control. And we're still in a stage where there's so much legacy and tech debt already that we have such a high bus factor of only one person doing Salesforce checkout and only two people being able to change Zuora workflows confidently. That means that if any of these three suddenly is not here anymore, then we lose a lot of internal knowledge that we really need to be able to operate.

**Thomas Agerbæk Ruby:** Are there any documentation structures in place or anything? Like if somebody was leaving, somebody else was taking over? Is there a manual ready for a person to take over this? Or is it still in a fragile state?

**Øivind:** And the simple answer is: there is nothing ready for anyone if things go to shit or whatever.

**Thomas Agerbæk Ruby:** Okay. That's all right. So moving on to what we initially talked about, or what I framed that this meeting was going to be about, is Zuora 2.0, Fosbury 2.0, I don't know, whatever we want to call it. So before you mentioned... may I want to go back to... you mentioned that there are still legacy systems. Is that legacy in terms of old Zuora-Salesforce stuff, or is it Shopify systems and old legacy systems and the old setup?

**Øivind:** Yeah. So I was talking about Salesforce is also legacy, but yes, there is still some Shopify legacy as well.

**Thomas Agerbæk Ruby:** So you could say categories of legacy. Multiple layers.

**Øivind:** Multiple layers. All right. But that then leads us on to 2.0, you call that. Yeah. So I've heard a little talk about it or mentions of it from you and Jesper. And yeah, basically let's talk about what's going on.

**Thomas Agerbæk Ruby:** Is it getting reworked into something new? Because the first system was supposed to give us a clearer idea of our customers in hand. It's going to make it easier for salespeople to do their work. And you could argue that we're not quite over at that point yet where it's easier for sales to do it. So what is Zuora 2.0?

**Øivind:** Zuora 2.0, in short, is simplifying the solution we have today. It's about providing the exact same capabilities that we have today, but we will build it simple so that when you go into Zuora, it's easy to understand what's going on, or ideally, have nobody go into Zuora because we can surface it in other places so that every intake or input into Zuora is controlled.

**Thomas Agerbæk Ruby:** And what triggered the need to revisit the current setup?

**Øivind:** Basically what I mentioned earlier about this bus factor, that we have these Zuora workflows that are immensely complex. They cost a lot of money. It's actually a huge number. Like, we're paying per workflow task. Just put that out there. You don't need to know that much, but there's a huge number of money that we're paying based on this. So, yeah, this was kind of like what we figured out. We have talked a lot. Yes, for me, I think Martijne has been somewhat involved, and Sean as well has been talking about, do we want to with the current setup make it a lot simpler, pull apart all the workflows, pull it into somewhere else, ensure everything works and whatnot. And every time we talk about this, we end up with, we need to do a big bang anyway because we can't just pull out some logic of our workflows. We need to pull apart our whole thing. And every time we get to that stage, we figure out that it doesn't make sense to build on the solution we have today. We should just build everything from scratch with new

assumptions and then try and see what kind of requirements can we remove because we were young and stupid and we didn't know Zuora. So there's a lot of things that were implemented in a complex way. An example of that is order line items for hardware, like separating hardware and software. That makes it so we can actually use pending activation on subscriptions instead of whatever we use today. So these kind of things we figured out that if we want to do all of this in the setup we have today, it's going to take a huge amount of time and a huge amount of resources and a huge amount of testing because it works. Whereas if we buy a new tenant of Zuora and we start building on that from scratch, completely fresh, there's no custom fields in it, and then we set up how we want it to be set up, and that's what we want to get to. So you can say 2.0 is literally a new tenant that we want to move to, basically because the old, or the current setup we have is so complicated and clustered and basically a mess. It's easier to just wipe it out, make a new one.

**Thomas Agerbæk Ruby:** Just how far have we come? When are you going to push the actual button? How is that going to work?

**Øivind:** So this is literally what we're talking about now, because our first goal was to do big bang, switch everything. We put that to management. They said, "Yeah, we are not doing that," because of the risk of switching everything all at once, and then suddenly we're talking new ways of working for sales and everybody else. So what we are doing now is a de-risking way, where we will in V1, which we're calling it now... For a lab one. We will introduce all of the critical components and still keep all the workflows running on them. But we'll introduce the critical components and ensure all process changes for sales, for support. They will now have those new ways of working in V1. So that when we eventually click the button of moving to V2, no processes will change. It will only be like the meat and the backend that will change, but what they do will be exactly the same. So then we remove that barrier of having to ensure everything works upon the migration, which we will have to do, but also ensuring everybody's trained at the same time and ready for the onboarding, right? So yeah, we will de-risk everything in V1, ensure everything works there. Then we will do a migration from our production tenant today to a new production tenant, so that we get a completely new product catalog which is ready for the future, you can say. And that's when we will just switch all of the processes to look at that new tenant instead, and there will be no process change. That's the goal.

**Thomas Agerbæk Ruby:** Sounds so easy.

**Øivind:** Yeah, it does. Fosbury did as well.

**Thomas Agerbæk Ruby:** So one of the last questions I asked you first interview was if you were going to do the Fosbury project again, how would you do it? And you answered something as far as I remember, keeping it internal and also have the correct people with the correct knowledge before you start a project like this. Do you feel like you're using some of the learnings from the first project now?

**Øivind:** 100%. We, Jesper, me, Gert, have already spent almost two months designing the solution, and we got some architects flown in to review our solution, which they basically gave a

thumbs up. They didn't have anything to say. We are now in the process of talking to Monetization, Business Optimization to get their buy-in, to say, "This is what we need to do. This is our design. How does this fit in with what you need to do? What kind of work falls on your shoulders with this?" And we're also talking with Sander to get the buy-in to actually have a feature freeze, to be able to go through all of this so that we don't get these sudden commercial initiatives thrown at us. We actually have the option of saying no and keep our focus on it. But essentially, we want to involve everyone, you could say. But we want to take it slow. We need to first ensure that we are aligned within our team, then align it with our closest dependent teams, and then they know which users they service, so they can also align with them. And then we will not start doing any work until the scope and everything is clear. We have the project timeline. We understand the risks that we want to de-risk and why we're doing it. But yeah, so already now it seems like we're in a lot better place, but you never know.

**Thomas Agerbæk Ruby:** And you've had some meetings with management and C-suite people about this. What is their opinion on the project?

**Øivind:** I haven't heard it directly from them. I've only heard from Martijne, Sander, and Sean. All of them are like, "We need this. This has to happen." We need to be able to iterate on this quickly, and we need to be able to have the observability of critical workflows and be able to ensure that everything runs as smoothly as it should. They're all like, "We 100% need this." What I have heard is that because we will introduce a new way for sales to work, of course, there is naturally some hesitation over on their end, which I understand, but the goal also is that we are literally going to kill Salesforce CPQ. Yeah, we are going to build our own CPQ inside of Veo, which is going to live kind of like e-commerce, where sales will go to an e-commerce page. They'll build something there, send a payment link that will send them to an e-commerce checkout. So we will disregard the old Salesforce CPQ, and you could say this is a completely new thing for sales. But what's the one thing that sales does not like? That's how slow Salesforce is. I think this is going to be a good change for them. They just need to see it. Yeah. The other thing they don't like is changes, even though they scream about going back to Pipedrive at the moment.

**Thomas Agerbæk Ruby:** Do you feel like there is more priority/attention from management and C-suite type people on this project than there was in the first project back then? Do you feel like there's been a change in the way of viewing these projects?

**Øivind:** Yes. At least from the short intro that we have now, and we have more people with different backgrounds overseeing this. We have Sean, who is the principal engineer. He can point out some technical issues no matter the context, you could say. You have Sander, who is really commercially driven, so he really understands how we should word it to others and how we should de-risk it. Then you have Mark Øland, who is mostly cost and engineering specific. So he will also be able to point at if we tell him that we will lower cost because we kill workflows by this. He will be on our side. And then Alex, who is the C-level who is overseeing this account. He will listen to all of those three, and if we have those, then he will understand what's going on.

Yeah, just this hierarchy that I can see now we didn't have before. We had Ulver and Michael and Ditte.

**Thomas Agerbæk Ruby:** No, we talked today with Tim. He said that was very KPI-driven and was pushing a lot.

**Øivind:** Yeah, because this was a problem on her table, but not necessarily... there was probably not an understanding of the systems and how to fix it.

**Thomas Agerbæk Ruby:** So this is definitely a different way of doing it, I would argue.

**Øivind:** Yes, I agree.

**Thomas Agerbæk Ruby:** Yeah. Okay. So we are now in April 2025. Would you say that the state of our business systems, and where we have come from? Has it gotten better since Business IT? Would you say that we are set for or in a better state now than a year ago?

**Øivind:** Yes. I think only directly a year ago, then I was already swamped in Fosbury support. Right now I don't have to look at support tickets. It's like I can get into work one day and actually have something I want to work on, and I'll be able to work on it. Just that in itself, to me, shows a great, great improvement. And also the overall feeling in the company that people now understand what is going on. They can understand that things are the way they are because they have actually been there, they have seen it, they have tried it themselves, and they can understand that when we say, "We can't do it like this, it will take us two months," then they say, "Sure, that makes sense." Whereas before, we will be like, "But it's a new system, it will work, just make it work."

**Thomas Agerbæk Ruby:** And do you feel like it has gotten better as well in terms of strategy and governance from the Builders side now that there are more long-term defined goals for the systems?

**Øivind:** We are more strategically involved in what we are doing as well. Yeah. Already now it feels so nice to be a part of it as well. We've spent two, three months now only doing this kind of strategic work, documentation work about making decisions. And everything right now is documented, just the fact that we have a decision log compared to Fosbury. It's a Builder's way of doing this because they have been on this probably longer than Business IT's mindset. Good.

**Thomas Agerbæk Ruby:** Is there anything you feel like hasn't been talked about enough, or even the reorganization, or the way we've approached systems, so on Salesforce or Zuora?

**Øivind:** I am, as an individual, always a pessimist, right? So, at least with these kind of things, and I've learned to be pessimistic, so I already now have this feeling about the timeline for Zuora 2.0, because we will need to set the expectations very low for this to succeed. If we now say that we will get this done by December, then I fear we will get into the same shit storm of Fosbury, where we had a deadline we didn't reach it and we push it one month, then one

month, and then we're constantly under stress. We need to give ourselves the time to do this. I think that's my main concern right now. I think that's it. I had a nice talk. Sorry to take more time.

*Meeting ended after 00:28:38*