Leiden University

ICT in Business and the Public Sector

Life Cycle and Development Best Practices of AdTech

Name:	Ruben R.C. Visser
Student ID:	s4032365

Date:	May 1, 2025
1st supervisor:	Prof.dr.ir. J.M.W. Visser
2nd supervisor:	Drs. N. van Weeren
Company supervisor:	W. Schmall

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Einsteinweg 55
2333 CC Leiden
The Netherlands

Abstract

Background:
Advertising technology (AdTech) plays an important role in the digital economy of today. AdTech serves targeted online advertisements to diverse online systems. However, its unique life cycle stages and challenges are not explored, particularly in comparison to the standard software life cycle. AdTech includes Demand-Side Platforms (DSPs), Supply-Side Platforms (SSPs), Programmatic Advertising Platforms and other systems that must handle real-time traffic, scalability, extensive metadata storage and regulatory compliance.

Aim:
The aim of this study is to characterise the software life cycle of AdTech. This will be done by identifying the differences from the standard software life cycles. Next to this, the aim is to propose best practices tailored to the specific needs of AdTech. A key objective is to map these best practices with the AdTech software life cycle by categorising them into its stages. Next to this, the research seeks to understand whether the findings are also applicable to related domains, such as the related Marketing Technology (MarTech).

Method:
A Design Science Research methodology was used in the research, with the focus on creating a tailored AdTech software life cycle model and a catalogue of best practices for each life cycle stage. The study incorporated interviews with industry experts and an exploratory case study on branded content software to identify challenges and validate the artifacts. Insights were further evaluated using the Technology Acceptance Model (TAM) through stakeholder feedback.

Results:
In the exploratory phase four interviews with AdTech experts have been conducted, along with a two month-long case study. Based on the findings from the interviews and case study, two artifacts have been developed: the AdTech software life cycle model and a best practice catalogue consisting of seven best practices tailored to the AdTech software life cycle. These artifacts have been evaluated through a demonstration and a questionnaire based on the Technology Acceptance Model and with in-depth interviews.

Conclusion:
A software life cycle model for AdTech has been identified, along with seven key best practices, both of which have been evaluated by an extensive group of AdTech experts. The AdTech software life cycle consists of four stages: Initial Development, Scaling, Evolution and Replacement. The best practices include: Develop AdTech in Components, Develop AdTech That is Simple to Implement, Scale in the Cloud, Monitoring of AdTech Resources, Prioritising of Development Tasks, Leverage CI/CD in AdTech and Replacing AdTech in Components. Each best practice is mapped in a specific stage of the AdTech software life cycle.

The AdTech software life cycle model has been recognised as a valuable model for explaining AdTech software and its development. Also, the best practices have been recognised as valuable while some were highlighted as must-haves in AdTech development. However, more research is required to validate and expand the best practices catalogue, as well as to explore whether other software domains share characteristics with AdTech that necessitate a tailored software life cycle. Additionally, investigating the continued relevance of standard software life cycle models, such as those proposed by Rajlich and Bennett [37], could provide further insights into evolving software.

Acknowledgements

From the start of this thesis, I have received invaluable insights and support from my first supervisor, Professor Joost M.W. Visser. Your knowledge of software evolution has been very insightful in guiding my research and without your expertise, I would have faced many challenges. Therefore, I would like to thank you for the support and I hope that my work has also provided new insights for you.

I would also like to thank to Drs. Niels van Weeren, my second supervisor. Your guidance and perspectives has helped me very much, particularly your insights into MarTech. You also helped me with connecting with a MarTech expert, which I am very grateful for.

Next to this, I am grateful to every participant who contributed through interviews or in my workshop. Without their expertise about AdTech, I would not have been able to capture the full picture. Next to this, I would like to thank Adswag, for allowing me to conduct a case study within their organisation.

Finally, I would like to thank my dad for your support during my studies and for reviewing my thesis with me. Your encouragement has been a blessing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advertising software, also known as AdTech, plays an important role in the digital economy by enabling targeted advertisements to reach consumers across various online platforms. Unlike regular software, advertising software faces distinct challenges, including real-time traffic management, large-scale metadata storage, dynamic scalability to accommodate fluctuating user demands and compliance with strict regulations. Due these complexities, it is expected that the life cycle of advertising software may differ from standard software stages, indicating the need for a separate investigation. If such differences exist, it may imply that developers and managers of AdTech software lack guidance, as standard software life cycle models may not sufficiently reflect the realities of AdTech development.

1.1 Problem Statement

At the beginning of software development have developers viewed all work following the initial delivery as software maintenance. However, as software development evolved over time, this perspective was no longer enough [37]. To address this, Rajlich and Bennett [37] introduced the software life cycle model which include five software stages: Initial development, Evolution, Servicing, Phaseout and Closedown. However, it remains unclear whether AdTech follows the same software life cycle as standard software. Due to this lack of understanding, it is also unclear how to best navigate AdTech through its different software life cycle stages. Especially for managers and software engineers, navigating the entire life cycle of AdTech is challenging without clear guidance. To bridge this gap, best practices for software development need to be defined. By providing practitioners with actionable guidance in the AdTech field.

Software development best practices are often designed for specific domains or methodologies. For instance, Meso and Jain [31] emphasise the use of Agile methodologies, while Assal and Chiasson [3] focus on security in software development. However, there are no established best practices for developing AdTech. The lack of domain specific guidance creates a gap in existing software development guidelines, leaving AdTech teams without clear guidelines to follow.

In this study, AdTech refers to "all technologies, software and services used to deliver and target online advertisements" [48]. The AdTech stack consists of the following key elements: Demand-Side Platforms (DSPs) for automated and real-time ad inventory purchasing, Supply-Side Platforms (SSPs) for managing and optimising ad inventory sales for publishers, Programmatic Advertising Platforms, which integrate DSPs and SSPs, for automating and optimising the ad buying process by advanced algorithms and real-time optimisations, Data Management Platforms (DMPs) for audience data collection and targeting and tools for ad verification and fraud prevention [44]. In a blog post, Gilmanov [17] also includes Ad server platforms, Ad networks and Ad exchange platforms in the AdTech stack. In this study, we include these components within the scope of AdTech, as they also play a crucial role in the AdTech ecosystem. See Figure 1.1 for a visualisation of the AdTech stack from Gilmanov [17].

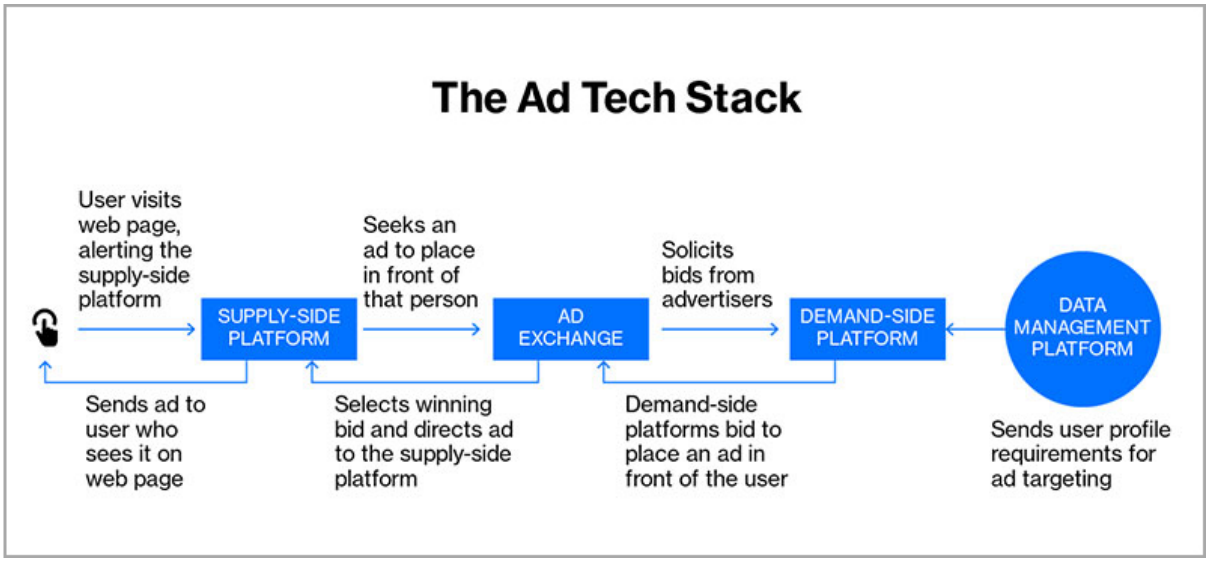


Figure 1.1: The AdTech Stack from Gilmanov [17]

AdTech software can be divided into two layers: foundational infrastructure, such as Demand-Side Platforms and Supply-Side Platforms and software that facilitates specific campaigns by leveraging this infrastructure. This research will consider software built on top of the infrastructure as features that leverage its capabilities.

AdTech has unique characteristics that differentiate it from standard software. To start, it must manage significant amounts of real-time traffic from the DSPs and SSPs. Because of these amounts it needs to dynamically scale to handle fluctuations in demand, driven by factors such as campaign performance, user interactions and traffic spikes [52]. AdTech also needs to store large volumes of metadata [51] to use this later for real-time targeting, personalisation, optimisation of ad delivery and performance analysis [11]. Next to this, is AdTech often embedded within websites by publishers [35]. AdTech also operates in a highly regulated space with complex data privacy requirements (GDPR) [42]. These unique requirements and challenges point out the need to identify the AdTech software life cycle and fill in the gap in existing guidelines. To address these challenges and provide clear guidance, this study seeks to explore a fundamental question:

The Research Question

How can the standard software life cycle model be adapted to fit AdTech, and which best practices help bridge the gap in existing guidelines throughout this cycle?

1.2 Objective

The objectives of this research include defining the software life cycle specific to AdTech while highlighting how it may differ from the standard software life cycle. Next to this, the objective is to map best practices to various stages of the AdTech software life cycle as identified in this study. This will involve drawing from existing studies and developing tailored best practices to address the unique needs of AdTech. Additionally, the objective is to define the characteristics of AdTech. With these insight we can discover whether the life cycle could be applied to other domains with similar characteristics, such as MarTech.

Currently, there are standard software life cycle models and sets of best practices for all software development. However, the proposed AdTech software life cycle and best practices provide specialised solutions. This tailored framework delivers actionable insights to help professionals navigate the dynamic life cycle of AdTech. So, by providing the framework this study aims to enhance both theoretical knowledge and practical guidance for practitioners in the domain.

1.3 Research Questions

To answer the main research question, we address the following sub research questions. The questions focus on understanding the unique characteristics of AdTech software, its life cycle stages. The last research question explores necessity for tailored best practices to navigate its development.

RQ.1 How can the AdTech software life cycle be modelled accurately?

This question explores the entire life cycle of AdTech software. Understanding its structure is key to identifying AdTech software stages.

RQ.2 Does the AdTech software life cycle differ from the standard software life cycle?

The standard software life cycle should be applicable for many industries. However, it might not address the demands of AdTech. This question investigates whether and how the AdTech software life cycle differs from standard life cycles.

RQ.3 What best practices are effective for the development and maintenance of AdTech, and how can they be mapped to the software life cycle?

Best practices streamline software development, but AdTech's unique demands might call for tailored best practices. This question examines whether the AdTech field requires new best practices.

1.4 Overview of the Thesis

This thesis applies the Design Science Research Methodology to examine the unique challenges of AdTech and to develop a software life cycle model with tailored best practices for AdTech. Chapter 2 covers the background and related work by exploring concepts of software life cycles. Next to this, covers the chapter existing best practices and the characteristics of AdTech. Chapter 3 outlines the methodological approach, introducing the used Design Science Research methodology and explaining its application in developing a tailored AdTech software life cycle model and best practices.

The results of the artifact design activity is discussed in Chapter 4, which includes the tailored AdTech software life cycle model and best practices catalogue. Next Chapter 5 evaluates these artifacts using stakeholder feedback and the TAM, for measuring the artifacts usability and effectiveness in practice.

The outcomes and broader implications are discussed in Chapter 6. Next to this, will the applicability of the artifacts will be discussed with the limitations of this study. Finally, Chapter 7 summarises the contributions of this research and offers recommendations for future work, including potential extensions to related fields such as MarTech.

Chapter 2

Background and Related Work

Standard software life cycles models for standard software reflect observed stages that guide decision-making throughout a system's lifetime [27]. However, the demands of AdTech software may require adaptations to standard software life cycle models. This chapter examines previous works that align with this thesis, focusing on software life cycles, best practices in software development and introducing concepts specific to the AdTech domain.

2.1 Software Life Cycles

In software engineering, software life cycles are commonly used to outline the various stages and their sequence during the development and execution in a production environment of specific software artifacts [45]. Software life cycle models are generally meant to be both descriptive of what typically happens in practice and prescriptive of what ideally should happen in practice [7]. Defining the software life cycle has been the focus of numerous studies [45, 37, 16, 12], with some concentrating specifically on the initial development stage and others addressing the entire cycle. The software life cycle from Rajlich and Bennett's [37] addresses the entire life cycle and consists of the following stages:

- **Initial Development**: Engineers develop the system's first functioning version.
- **Evolution**: Engineers extend the capabilities and functionality of the system to meet user needs, possibly in major ways.
- **Servicing**: Engineers make minor defect repairs and simple functional changes.
- **Phaseout**: The company decides not to undertake any more servicing, seeking to generate revenue from the system as long as possible.
- **Closedown**: The company withdraws the system from the market and directs users to a replacement system, if one exists.

In addition to the software life cycle definition provided by Rajlich and Bennett [37], other definitions exist. For example, IEEE [23] defines the software life cycle as "The period of time that begins when a software product is conceived and ends when the software is no longer available for use". This definition by [23] identifies eight stages: The concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase and optionally a retirement phase. In this context, 'checkout' refers to "Testing conducted in the operational or support environment to ensure that a software product performs as required after installation."[23]. In comparison to Rajlich en Bennett, the IEEE life cycle provides more focus on the development before initial release than on the continued development after release.

2.1.1 Software Development Life Cycle

Software progresses through the software life cycle stages by developing and maturing over time. To steer through the software life cycle stages different software development methods can be applied. Each method offers distinct approaches for managing the process. In this context, Saravanan et al. [38] provided

a comparative analysis of development focused stages within Software Development Life Cycle (SDLC) models. The study identified the three most used software development methods: Agile, Waterfall and Scrum.

The Waterfall model consists of five phases: Requirement Analysis, Design, Implementation, Testing and Maintenance. In contrast to the Waterfall model, focuses the the Agile methodology on iterative development through stages such as Timebox Planning, Requirements Elicitation, Detailed System Design, Coding, Development and Testing, Demonstration and Retrospective Meetings [29].

As software development got bigger, it also became more complex. Customer requirements changed more frequently, making it difficult for traditional SDLC models to keep up with market demands. Sequential approaches like Waterfall lack the flexibility needed to accommodate rapid changes [29].

To response and adapt faster, a new methodology was introduced called Scrum. Scrum is not a structured lifecycle model but rather a software development method (SDM) within Agile [26]. The methodology focuses on iterative and incremental development. It centres around roles, artifacts and ceremonies such as the Product Backlog, Sprint Backlog, Sprints, Daily Scrum, Sprint Review and Sprint Retrospective.

Although the Waterfall model and the Agile methodology are different in their structure and processes they both share the same goal: delivering a high-quality product in a predictable, efficient and responsive manner. They involve similar fundamental activities, gathering, analysing, designing, coding, testing, releasing, maintaining and retiring. However, they are different in how these activities are carried out [34, 32]. Especially Agile, serves as an umbrella for various methodologies, with Scrum being the most widely adopted framework [29]. These methodologies (Waterfall, Spiral and Iterative) fall under Software Development Methodologies, aiming to improve the quality of software products and to map, maintain and control these products as a general product [40].

In the literature review from Saravanan et al. [38] on software development methods reveals that no single SDM approach is universally applicable. Each method has strengths and weaknesses. Stakeholders should evaluate the methods and decide which method fits bests for their context.

2.1.2 Choice of Software Life Cycle

For this study, we will investigate the software life cycle of AdTech. The software life cycle includes all stages of a software's existence, including the Phaseout and Closedown stages, whereas the SDLC's final phase is Maintenance. Additionally, the SDLC primarily focuses on the development process from a development team's perspective, while the software life cycle examines the technological perspective, focussing the software itself. This perspective is particularly important for documenting its technical traits.

As a guideline for this study, we will decided to chose the software life cycle model from by Rajlich and Bennett [37] as our point of departure. This model provides distinct stages beyond initial development, addressing aspects where the software life cycle defined by IEEE [23] falls short. Next to this, it offers sufficient documentation. The software life cycle model not only outlines the key stages, from initial development to closedown, but also provides contextual insights into stage characteristics such as staff expertise, economics and software architecture at each stage.

2.2 Best Practices

For this research, we define best practices as routine applications of knowledge deemed superior to alternative approaches. Best practices are often developed outside the organisation, such as industry standards, academic research and adopted within an organisation to improve performance and to stay competitive within an industry [20].

The selection of 'best' practices should be guided by clearly defined criteria [8, 36, 49]. In this study, we adhere to the principles proposed by Wu, Liu and Bretschneider [49]. Their empirical findings show that judges of 6 government departments, 1 from an international organisation and 16 from top universities and research institutes prioritise innovativeness, sustainability and replicability alongside effectiveness when recommending best practices [49]. While Wu et al. [49] hypothesised that importance would also be a factor, their results did not confirm this. Therefore, they excluded the importance principle. The final principles in this study are as follows [49]:

- **Replicability**: It refers to the quality or state that can be done in exactly the same way as before, or produced again to be exactly the same as before. Best practices must be generalisable and applicable to other locations and conditions.
- **Effectiveness**: It refers to the extent to which the expected goals or outcomes are achieved.
- **Sustainability**: It means the quality of being able to continue over a period of time.
- **Innovativeness**: This criterion refers to the novelty of the measures and mechanisms developed.

The best practices derived from this study will be analysed and discussed using this set of principles. We will assess them whether they have been properly applied in their definition and selection. The results of this assessment can be found in Chapter 6.

2.2.1 Software Development Best Practices

Best practices for software development have been studied before, Jain and Suman [24] provided a systematic literature review about software development best practices. They categorised best practices into the following areas: Requirement Engineering, System Design, Coding, Testing and Integration. This structure aligns with the stages of the SDLC stages. This approach includes the initial development stages, but overlooks later stages, such as maintenance and retirement stages. These final stages are critical for a complete a full software life cycle perspective.

The best practices from Meso and Jain [31] are structured differently, they used three dimensions to structure the practices: Product Dimension (Artifact), Process Dimension (Development) and People Dimension (Software Team). Each best practice is formulated separately for each of these dimensions. By utilising this approach, the best practices address the needs of the software product, development process and team performance.

Whited and Hanna [46] used a structure for documenting best practices that provides clarity and guidance for implementing them. This structure can be applied to best practices from various domains, not just software development. Each best practice is organised using the following elements:

- **Title**: The best practice.
- **Brief Description**: Provides a brief overview of the practice.
- **Additional Details**: describes details such as the implementation of the best practice and other necessary information.
- **Objective**: What the outcome should be after applying the best practice.
- **When to Apply**: Describes when the best practice is relevant.
- **Cost Implications**: Covers the costs aspects associated with the best practice.
- **Conditions for Successful Application**: Describes when the best practice is successfully executed.
- **Cautions**: Describes risks and challenges in applying the best practice.

By using this structure the best practices are insightful presented and made easily understandable. This is crucial for stakeholders who wish to apply them in a real-life setting. In addition to the best practice documentation approach proposed by Whited and Hanna [46], there are also other documentation approaches. Alwazae et al. [2] developed a Best Practice Document Template (BPDT). The BPDT organises best practices in components such as: Summary of BP, Representation, Requirements for Applying BP, BP Actor, BP Properties and BP Implementation. For documenting the best practices in this study, we have used the best practice documentation approach from Whited and Hanna [46]. This best practice documentation approach is the most comprehensive. A more detailed justification for this choice, along with the justification for selecting the best practice categorisation approach, is provided in Chapter 3, "Best Practices Documentation Approach".

2.3 AdTech

In the problem statement we described AdTech, which refers to all technologies, software and services used to deliver and target online advertisements [48]. The AdTech stack includes DSPs, SSPs and Programmatic Advertising Platforms, which enable automated, data-driven ad transactions in real time. Additionally, Ad servers, Ad networks and Ad exchange platforms are highlighted as also included as components of the stack [44, 17].

While figure 1.1 highlights all the platforms in AdTech. Figure 2.1 illustrates the structure of the display advertising ecosystem. It shows the flow of data, ads, servers and the flow of revenue. Different studies have illustrated this ecosystem in various ways. Some studies use terms like publisher ad server and advertiser ad server instead of SSP and DSP. Next to this, view others the SSP, DSP and ad server as a single entity, given that some platforms provide all these services.

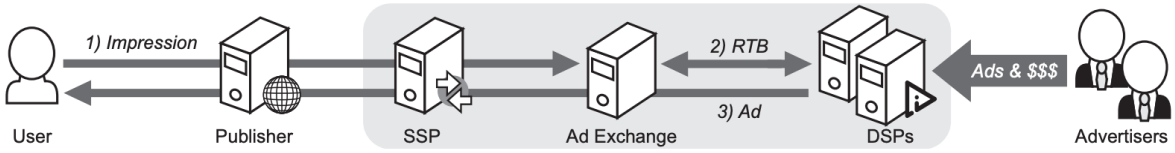


Figure 2.1: The display advertising ecosystem, impressions and tracking data flow left to right and revenue and ads flow right to left [5].

2.3.1 The Online Advertising Ecosystem

In the online advertising ecosystem three primary stakeholders are involved: publishers (websites) supply ad inventory (the available impressions publishers can allocate to ads [28]), ad networks (SSPs and ad exchanges) facilitate the real-time bidding process and advertisers (brands) buy ads from ad inventory [43]. In the online advertising ecosystem many software solutions play a crucial role. [30] paper contains a brief introduction to this topic. In the mid-2000s, publishers started exploring ways to monetise the "remnant" inventory they were unable to sell through ad networks. Ad exchanges were introduced as a solution for this, they filled these slots in real time by taking bids from various advertisers by utilising multiple advertising networks. This process is known as "real-time bidding" (RTB). Over time, ad exchanges expanded beyond remnant inventory. This change led to the development of several intermediary business models within the exchange ecosystem:

- SSPs are analytics and automation companies that specialise in optimising publisher revenue by directing ad impressions to the most lucrative ad exchanges [4].
- DDSPs are analytics and automation companies. They manage and executing ad campaigns for advertisers[4].
- Data Management Platforms (DMPs) sell ad targeting data to advertisers in real time, often basing their targeting recommendations on tracking data, information purchased from publishers and offline consumer databases [30, 22].
- Ad servers are used by publishers on the supply side to manage advertisements. These ad servers differ somewhat from those on the demand side in their technical functions, as they are designed to broadcast the available inventory [22]. Consequently, ad servers on the supply side handle tasks such as administration, delivery and tracking, including managing ad placements and monitoring clicks on online advertisements [33]. In contrast, advertisers use ad servers to deliver digital ads and streamline campaign management across various media and publishers [22].

As shown in Figure 2.1, we can see that both SSPs and DSPs facilitate the buying and selling process of ads. However, they are clearly different platforms. SSPs collaborate with publishers to manage relationships across multiple ad exchanges, aiming to maximise revenue. In contrast, DSPs support advertisers by evaluating the value of individual impressions and optimising bid prices. Next to this, many companies provide products that span both categories [5].

2.3.2 AdTech and MarTech

In the same domain, MarTech also plays a critical role. Although MarTech serves a different function. AdTech focuses on optimising ad placements and ROI through tools like programmatic advertising and real-time bidding, relying heavily on data [44]. MarTech, on the other hand, encompasses technologies like campaign automation and customer journey mapping to improve overall marketing activities and customer experiences [44]. In this research, we briefly compare the two domains and explore if AdTech's software life cycle aligns with MarTech's. This section can be found in Chapter 6.

Chapter 3

Method

In this research, a qualitative approach will be adopted, applying a "Research by Design" [10] approach. The central artifact produced will be the AdTech software life cycle model, accompanied by a catalogue of best practices for developing AdTech, categorised according to the various stages of this life cycle. The best practice catalogue will provide a guideline for each life cycle stage. The research will involve two rounds of interviews with stakeholders in the AdTech industry (Exploratory interviews and In-depth interviews). Next to this, an exploratory case study will be conducted to investigate real-life challenges in the industry.

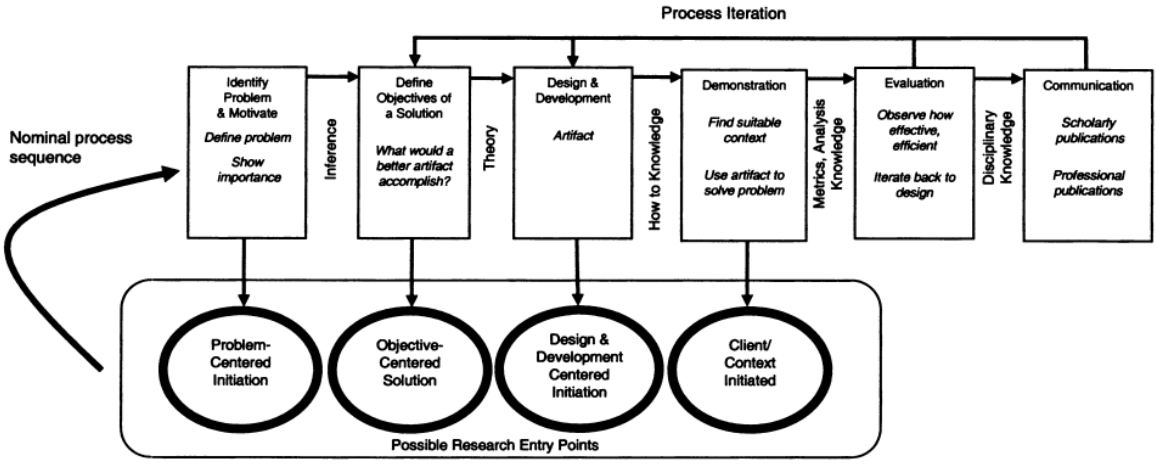


Figure 3.1: Design Science Research Methodology (DSRM) Process Model by Peffers et al [25]

3.1 Methodological Approach

The methodological approach for this study follows the Design Science Research Methodology (DSRM) by Peffers et al. [25] (see Figure 3.1). The methodology starts with identifying the problem and defining the research objectives, which were in detail outlined in Chapter 1, "Problem Statement". This chapter discusses the challenges in AdTech.

The solution and research objectives are discussed in Chapter 1, "Objectives". This chapter elaborates on the gap in the literature. By exploring whether AdTech follows the life cycle of regular software or requires a new approach. To investigate this, the first research objective is to identify the characteristics of AdTech and define its software life cycle by breaking it into distinct stages. The second research objective is to assign tailored best practices to each stage. By addressing the specific challenges and opportunities within AdTech software life cycle. Finally, the research objective is to determine if these findings can be applied to related domains, such as MarTech, which has similar characteristics as AdTech.

By addressing this gap the research aligns with the solution objective, which is to provide both theoretical understanding and practical guidance for AdTech development. The solution objective of this study is to develop a tailored best practices that delivers actionable guidelines to help professionals develop AdTech more effectively. To meet the AdTech practitioners needs, the framework should support and be accepted by AdTech experts. Additionally, it must also meet non-functional constraints, such as ensuring usability and adoption which align with the principles of the Davis [13]. The principles focus on ease of use and perceived usefulness in the industry. By doing this, this study adds theoretical knowledge and provides guidelines for practical application, leading to more efficient and effective AdTech development.

Following the DSRM, insights are gathered and a theory is crafted based on these insights. The insights will be gathered through interviews with industry experts and a case study which is discussed in Chapter 3 "Data collection". This theory has been crafted by doing data analysis with coding which is discussed in Chapter 3 "Data analysis".

In the design and development phase have used this theory to create artifacts tailored to AdTech. The artifacts developed in this study include a tailored software life cycle model for AdTech and best practices for each stage of the software life cycle. Detailed descriptions of these artifacts can be found in Chapter 4, "Artifacts".

To validate the artifacts, a workshop was conducted with stakeholders who work in the AdTech field such as Software Engineers and Product Owners. The session provided an audience to present the developed artifacts. By demonstrating how they address challenges in AdTech software development. Stakeholders were encouraged to give feedback about the practicality and applicability of the proposed life cycle model and best practices. This feedback was used to refine the final version of the artifacts, ensuring they are both theoretically refined and practically relevant for AdTech professionals.

Evaluation of the artifact is conducted using the TAM. Validation interviews were conducted with stakeholders and AdTech users. The interviews were focussed on factors such as perceived usefulness, ease of implementation and overall impact on AdTech development. Participants provided feedback on how well the proposed solutions fit with AdTech's requirements, their practicality in real-world applications and areas for improvement.

The final phase of DSRM is communication through publication. This research will be made publicly available in the LIACS Thesis Repository (2024–2025) of Leiden University. Additionally, infographics were created to visually illustrate the AdTech software life cycle and the AdTech best practices. By showing the artifacts in the form of infographic it should the finding should be more accessible. These infographics can be found in Appendix E.

3.2 Case Study Approach

An exploratory case study on branded content software development will be conducted to investigate the AdTech development and its challenges it faces during its life cycle. As an exploratory case study, it will serve as an initial investigation to derive new hypotheses and build theories about the AdTech software life cycle [14]. Based on the findings from both the interviews and the case study, a theory will be created along with AdTech best practices.

The case study takes place at Adswag, a Dutch technology company which specialises in digital advertising solutions. The office is located in Amsterdam, the Netherlands. During the case study, was Adswag developing branded content distribution software. This software allows advertisers to scale and distribute sponsored content (such as branded articles) across a network of over 200 publisher websites with minimal manual effort [1].

The case study on branded content software is chosen as it adds value to the research since it covers AdTech development, as outlined in the "Problem Statement" chapter. It adds value to the study by demonstrating how issues such as real-time traffic management, scalability, data handling for targeting and regulatory compliance in practice within the context of this branded content software. The branded content software serves as an proper example of AdTech since it operates as a programmatic advertising platform. The software facilitates both the demand and supply sides of the value chain.

3.2.1 Case Study Documentation Approach

The case study documentation approach was conducted using direct observation logs, which took the form of reality-based evidence capturing events as they occur or contextual evidence capturing the context of events [50]. The observations were drawn from SCRUM meetings, such as daily stand-ups, sprint planning sessions and additional meetings. This type of observational evidence is often useful for gaining additional insights into the topic being studied [50]. When a researcher is writing observational field notes, it is important to record the key details such as the date, time, location and a detailed description of the subjects involved. Including observations of emotions, direct quotes and specific actions can improve the context and depth to the data [47]. With this in mind, the logs of the observations have been structured to include what occurred, who was involved, when and where it happened.

By applying this documentation approach, most logs recorded problems in the "What" column and corresponding solutions in the "How" column. This structure enabled a systematic identification and analysis of each problem. These documented issues were used in the research to identify challenges specific to AdTech software, which are reflected in the artifacts. The identified solutions contributed to the development of best practices.

3.3 Best Practices Documentation Approach

In Chapter 2, "Software Development Best Practices," multiple ways for documenting best practices were evaluated. In the study by Jain and Suman [24], best practices were categorised based on a SDLC. Structering the best practices in a life cycle has also been done in this study however, instead of following the SDLC, which focuses on the development phase of software, the categorisation here considers the entire software life cycle. Given the scope of this study, which includes the entire life cycle of AdTech, the ordering is based on the software life cycle stages rather than the SDLC stages.

In the study by Meso and Jain [31], each best practice is described in three dimensions: Product, Process and People. However, this approach lacks some elements, such as a detailed description and guidance on when to apply the practices. Because of the absence of these essential elements, this approach has not been chosen for this study. Also the approach from Whited and Hanna [46] has been discussed and considered. This structure includes elements such as a title, brief description, additional details, objectives, application guidelines, cost implications, conditions for successful implementation and cautions. Since this method meets the requirements for creating a detailed catalogue of best practices, it has been used in this study.

Finally, the approach of Alwazae et al. [2] is discussed, which is based on their Best Practice Document Template. While many elements of this template align with the approach from Whited and Hanna [46], it is much more detailed, comprising 33 distinct points. However, given the extensive effort required to complete this template in its entirety for every best practice and the time constraints of this study, along with the fact that the approach from Whited and Hanna [46] sufficiently meets the needs of this research, this approach has not been used as the chosen framework. However, it has served as a helpful reference for filling in the template from Whited and Hanna [46].

3.4 Hypothesis

To address the research question: "*How can the standard software life cycle model be adapted to fit AdTech, and which best practices help bridge the gap in existing guidelines throughout this cycle?*", four hypotheses are formulated. These hypotheses shape the research process by directing the creation of the artifacts and serving as a foundation for their evaluation within the DSRM framework. In the first hypothesis (H), the software life cycle of AdTech is expected to be different from traditional software. The rapid pace of the advertising industry could mean that AdTech has a more iterative development process.

> **H_1**: The software life cycle of AdTech is shorter and more iterative compared to the standard software life cycle, due to the fast-paced advertising industry.

The second hypothesis discusses the financial and demand related factors that may influence the AdTech software life cycle. AdTech systems may require a certain level of demand from advertisers before development begins to justify the significant upfront costs.

H_2: AdTech must ensure a reliable demand from advertisers and other consumers before development begins, as its robust infrastructure requires significant upfront costs and resource investments.

The third hypothesis discusses the potential similarities between AdTech and standard software development life cycles. It suggests that while some practices are needed, the core life cycle stages may not differ significantly.

H_3: The software life cycle of AdTech does not differ significantly from the standard software life cycle, although it requires specialised best practices for managing privacy regulations.

The fourth hypothesis discusses that, despite the specific nature of AdTech, its life cycle might closely align with standard software life cycles, with only minor differences.

H_4: The software life cycle of AdTech does not differ significantly from the standard software life cycle.

3.5 Data collection

To evaluate the hypotheses, data has been collected in three phases, focusing on mainly qualitative methods to assess relationships within the AdTech domain and its life cycle, as well as the applicability and effectiveness of the proposed catalogue of best practices.

- **Exploratory Interviews**
  The first phase involved conducting exploratory interviews with stakeholders in the AdTech field. These interviews goal was to collect qualitative data to understand the relationships between the software life cycle and AdTech development. Next to this, the goal was to identify any domain-specific challenges.
- **Exploratory Case Study**
  The second phase involved an exploratory case study focused on branded content software development. This case study served as a means of data collection in the form of logs. This case study provided an opportunity to gain insights into the current development process. By gathering qualitative data from observations, documentation and stakeholder feedback has been analysed to identify key insights and patterns relevant to the software life cycle.
- **Validation Interview for Theory Understanding**
  The validation interview was an interview with an expert from the exploratory interviews. The interview was to confirm and refine the findings from the exploratory interviews. It incorporated a member validation approach, as described by Seale [39], where participants are asked to review the researcher's interpretation of the data to ensure it accurately reflected their experiences and responses. In this process the participants confirm the accuracy of individual interview transcripts and validating the themes identified during the analysis. By applying this method too, the study aimed to enhance the credibility and reliability of its findings.
- **Workshop**
  In the workshop, participants were asked to review the developed artifacts, including the defined AdTech software life cycle and the catalogue of best practices. The artifacts were demonstrated and after the demonstration, participants were asked to fill in a questionnaire about the artifact. The questionnaire had a qualitative approach based on the Technology Acceptance Model (TAM) by principles Davis [13]. In addition, the workshop provided an opportunity for participants to give feedback and share their opinions on the artifacts with the group.
- **In-depth Validation Interviews**
  The last interview round were in-depth interviews with AdTech developers. Here the goal was to evaluate the AdTech software life cycle and best practices catalogue. Participants assessed their relevance, practicality and effectiveness in solving key challenges. Feedback focused on satisfaction, recognition of problem-solving and acceptance of the artifacts. It was also an opportunity to receive feedback.

3.5.1 Interview Guides

Two interview guides were created for the exploratory and validation interviews. With the exploratory interview guide to examine the key stages of the AdTech software life cycle. This interview guide is structured on basis of the software life cycle stages by Rajlich and Bennett [37]. The guide also included targeted questions about certain AdTech topics, including scalability, technical challenges, regulatory compliance and maintenance strategies. By structuring the interview questions this way the aim is to capture AdTech's entire software life cycle with its challenges it faces. In addition to questions about the research topics, the interview guide also includes general questions about the interviewee's role, background and experience. During the interview process, the guide was refined to ensure all information could be captured for this study. The interview guide is provided in Appendix A.

The workshop questionnaire followed a quantitative approach based on the TAM [13], using a 1-to-5 rating scale. Participants rated the artifacts on perceived usefulness, ease of use, user satisfaction and behavioural intention to use. Optional open-ended questions provided additional qualitative insights. The questionnaire is provided in Appendix C.

The in-depth validation interview guide was created to evaluate the artifacts using the TAM [13] on a qualitative approach. With open questions focused on perceived usefulness, ease of use and user satisfaction with the proposed AdTech software life cycle and best practices. Participants provided feedback on the artifacts' ability to address AdTech's challenges, their practical applicability and the likelihood of adoption within their organisations. The validation interview guide is provided in Appendix D.

3.6 Data Analysis

In this study, each interview was transcribed to capture the participants' perspectives accurately. These transcriptions, combined with detailed observations from the case studies, provided a rich dataset for analysis. For the data analysis, a combination of two methods was used: the Qualitative Research method described by Seale [39] and the Grounded Theory method as Glaser describes [18].

3.6.1 Coding

For the coding process we used the Grounded Theory from Glaser and Strauss [19] with approach from Grounded Theory Online [21]. The approach starts with identifying the substantive area by defining the domain of interest and selecting the relevant substantive population. The next step is gathering relevant to the area of study, followed by open coding. During open coding, sentences from each interview transcription are analysed and assigned a descriptive category name that captures their essence.

The coding process in this study is called "Intermediate coding" [6]. This process uses the researcher coding in two ways. First, to develop individual categories by connecting categories with subcategories and creating a range of properties with dimensions. Second is to link the categories together [6]. This process continues iteratively, with new labels being created as long as unique properties are identified in the data. Once no new labels or properties emerge and the same ones are consistently observed, theoretical saturation of categories is reached [19]. This signifies that "no new properties emerge and the same properties continually emerge" [18].

Throughout this process, memos are written to document theoretical insights, relationships and methodological decisions [21]. Once the core category is established, selective coding and theoretical sampling are conducted to focus on the core category and related categories [6]. By this analytic emphasis, theoretical sampling helps researchers to craft robust theoretical categories. So, theoretical sampling occurs later in the analytical process and helps researchers identify variations within the category(ies) and the relationships between them [41].

Memos are then sorted to identify theoretical codes that organise substantive codes and shape the emerging theory. The final steps involve integrating the theory with existing literature through selective coding and writing up the developed theory. The code book from this research can be found in Appendix G.

3.6.2 Analysis Validation

For the validation of the codes and the theory, the Qualitative Research method described by Seale [39] was applied, involving a validation interview to ensure the accuracy of the data and its interpretation. This approach uses member validation, which means that participants review their interview transcript and the researcher's interpretation of it to confirm that it accurately reflects their experiences. Although this method is considered the "gold standard" in qualitative research, it introduces a separate validation process [15]. The Grounded Theory approach is different, this approach outlined by Glaser [18] includes validation as an integral component of the research process. Through constant comparative analysis and theoretical sampling, validation occurs continuously during data analysis, rather than being treated as a separate task performed after the analysis is finalised [15]. While both validation approaches are possible for this research, we chose for the separate validation interviews since the experts could not be continuously involved in the research because of their job obligations. This approach allowed us to schedule a dedicated time to validate the findings.

In addition to the validation interviews conducted with the interviewees from the exploratory phase, the developed theory was also validated by two additional AdTech experts. These additional insights provided more credible research findings.

Chapter 4

Design

This chapter introduces the findings of the study. The chapter begins with an overview of the interviews, followed by a summary of the codes and code groups from the data analysis. Next, each artifact is discussed in detail.

4.1 Overview of Data Collection

In this study, 10 AdTech industry experts participated. A total of 9 interviews were conducted, including 5 exploratory and 4 validation interviews. Two participants took part in both phases. The interviews involved 7 participants from 5 different companies—three operating within the AdTech industry and two within the software development industry. Next to this, a workshop with 8 participants was held to validate the artifact, of whom three had not participated in the interviews. Next to this, a case study was conducted focusing on the development of an AdTech solution: Branded Content at an AdTech company. All experts are kept anonymous due to privacy reasons. Below is a small introduction to the experts to provide insight into their expertise and job experience.

Exploratory interviewees

- **Expert A**: Head of Technology at an AdTech company. Expert A has over 15 years of experience in the AdTech industry and worked across four different companies. Skilled in developing AdTech solutions and leading project teams.
- **Expert B**: A Senior Software Developer at an IT servicing company. Expert B has 15 years of experience in software development with 6 years specialising in AdTech solutions. Skilled in developing back-end and front end applications.
- **Expert C**: A Sales Director at a digital media and technology company. Expert C has 10 years of experience in the AdTech industry and has worked at two AdTech companies. Expert C is skilled in building strong client relationships, AdTech sales and requirements engineering.
- **Expert D**: A Senior Full Stack Developer at an AdTech company. Expert D has 6 years of experience, including 4 years of experience in the AdTech industry. Skilled in building AdTech applications on top of existing AdTech infrastructure.
- **Expert E**: A marketing technology professional with over 25 years of experience, including extensive expertise in analysing and mapping the marketing technology landscape. Expert E is a founder of a MarTech company, specialises in assisting organisations to navigate the complexities of MarTech stacks. Next to this, is Expert E recognised for proficiency in marketing technology management.

Additional interviewees for validation of the artifacts

- **Expert F**: A Product Owner in the AdTech industry with 7 years of experience. Expert F has previously worked as a TV Planner in an AdTech system and as an Online Campaign Manager. In these roles the expert was responsible for managing DSPs, video players and AdTech platforms.

Expert F served as a Programmatic Demand Manager, specialising in integrating SSPs and DSPs for publishers. Now focused on project management in AdTech development.

- **Expert G**: A Business Development Director with 23 years of experience in the AdTech industry. Expert G has an extensive background as a Solution Architect and Product Manager in programmatic advertising. Expert G is specialised in data propositions and consultancy.
- **Expert H**: A Full Stack Developer with 10 years of experience, the last 6 years focused on AdTech. Expert H is specialised in reporting systems and data processing using AWS and BigQuery. Currently working at a software house.
- **Expert I**: A Sales Manager in the AdTech industry with 9 years of experience in selling ads, including audio and video ads. Expert I is skilled in AdTech-driven ad sales and optimising ad transactions in SSPs.
- **Expert J**: A Data Engineer at an IT servicing company, with 6 years of experience. Expert J has been focussed on AdTech for the last two years. Skilled in building cloud infrastructures.

4.1.1 Code and code groups

One of the data analysis goals was to identify codes in the transcripts. In total 171 codes were identified, including 151 unique codes classified into seven distinct code groups. Figure 4.1 shows the distribution of code groups. The largest group of codes is *Initial development of AdTech*, followed by the groups *AdTech in general* and *Evolution of AdTech*. The smallest code group is *Regulations*, likely because most interviewees have expertise in developing AdTech rather than in its legal and business aspects. The complete code book can be found in Appendix G.

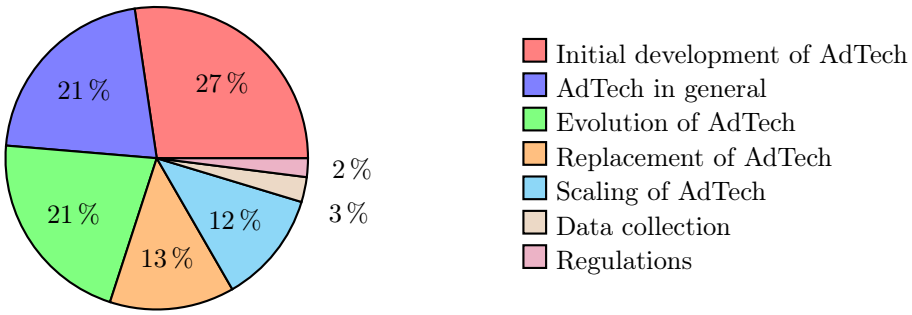


Figure 4.1: Distribution of codes across categories.

4.1.2 Case Study Results

The branded content case study was conducted as part of the exploratory phase of this research and spanned two months (January to February). The goal of the case study was to gain initial insights into the development process of Adswag's branded content software. During this period, a total of 19 log entries were recorded. The complete log book can be found in Appendix F.

4.2 Artifacts

This section introduces the developed artifacts in this study. The AdTech software life cycle model is created to provide more insights into AdTech's life cycle. Unlike standard software models, which lack specificity for industry specific challenges, the AdTech software life cycle model focuses on the demands of AdTech.

The main goal of the life cycle model is to provide a clear tool for navigating AdTech's life cycle. It emphasises understanding of the progression from the software's inception to its eventual conclusion. By achieving this goal, practitioners could better understand its scalability, evolution and efficiently allocate resources.

For best practices, the goal is to offer actionable guidelines that streamline AdTech development. These best practices are specifically created to address challenges in AdTech development. By providing these

artifacts our research fills in the current gap in specialised guidelines for this domain. This ensures that industry practitioners have clear, targeted strategies to enhance their systems.

In Chapter 6 "How the Artifacts Address Key AdTech Challenges" we discuss in detail how the artifacts address the challenges in AdTech.

4.2.1 Traits of AdTech

In the Problem Statement Chapter we discussed the traits of AdTech, distinguishing it from standard software systems. The most important trait from AdTech is that it must manage large amounts of real-time traffic generated by DSPs and SSSPs. This trait requires AdTech to have advanced capabilities for processing data at high speeds to facilitate real-time bidding and ad delivery. Because AdTech needs to deal with large amounts of real-time traffic AdTech resources must scale dynamically. Causes for of large amount of real-time traffic could be good campaign performance, user interactions with the ads, or other unexpected spikes. AdTech also relies on storing and processing large volumes of metadata, essential for real-time targeting, personalised ad delivery and ad performance analysis. Managing these operations while adhering to strict performance expectations adds to AdTech's challenges. Next to this, AdTech operates within a highly regulated space. AdTech must comply with data privacy laws, such as the GDPR. In addition, AdTech is installed within publisher websites, to do this AdTech requires simple integration with all publisher systems.

The traits we could identify from related work were also reflected in the case study and interviews. The case study showed AdTech's need for near real-time data processing, highlighting the trait of high-speed data processing. It also revealed the importance of seamless integration with publishers, as noted by the Head of Product, who stressed the need for a simple and but also robust integration process with all publisher websites. In the interviews, the importance for real-time traffic management was a much discussed topic, highlighting the requirement for scalable systems which need to be capable of handling high volumes of traffic. Additionally, regulatory compliance was pointed out as a concern, with experts raised the importance of adhering to GDPR and industry-specific regulations. While not always explicitly stated, the handling of large volumes of metadata also played a role in the software from the host of the case study. In particular in relation to real-time targeting and performance tracking. However, not all AdTech systems from the host's case study required metadata storage for targeting. In the case of the branded content software, targeting was done by selecting ads based on the context of the website, aligning ad content with the site's domain.

Through interviews and a case study, additional traits of AdTech have been identified. One remarkable trait was its need to continuously evolve. AdTech's environment faces many external factors such as trends, technological advancements and regulatory changes. These factors often influence one another, creating a domino effect within the industry. For instance, when new technology enters the market and is adopted by AdTech, regulations may later restrict its use, necessitating further adjustments within the systems. In contrast, regulatory changes can drive technological innovation, requiring AdTech to adapt to align with new standards.

In addition to these external pressures, have the AdTech companies intense competition. Publishers can easily switch to other AdTech companies software when competitors introduce new features that other doesn't have. This adds another factor of pressure, as AdTech companies must not only respond to external factors but also continually need to innovate to remain competitive. Modular architectures and reusable components are important to facilitate these shifts, since this makes seamless incorporation of new functionalities and the ability to adapt possible.

AdTech Requirements

Before AdTech companies decide to develop an AdTech solution, certain requirements must be taken in mind. Without meeting these requirements, the AdTech solution may face challenges such as a lack of trust or transparency issues.. Based on insights from the interviews and case study,the following requirements have been identified as critical to the success of an AdTech solution:

- The AdTech solution must comply with data protection regulations, including the GDPR, which governs the processing of personal data within the European Union. Next to this, must AdTech comply with industry specific regulations, such as those set by the KSA (Kans Spel op Afstand)

for betting advertising in the Netherlands as explained by Expert C. These laws impose specific requirements on cookie consent, tracking technologies and targeted advertising practices.

- The AdTech solution needs to be simple. The AdTech industry is already complex. Next to this, have many stakeholders, such as publishers, advertisers and agencies, not always have technical expertise. This makes it essential for AdTech to be accessible and easy to use.
- The AdTech solution needs to deliver what it promises. AdTech solutions must ensure that all the impressions promised actually get delivered. This means that these promises must could be fullfilled with the aviallable ad inventory. This ensures that delivery expectations are realistic and that campaigns run within the actual capacity of publishers' ad spaces within the campaigns time frame.
- The AdTech solution must provide publishers with control and monitoring capabilities. Publishers want to be able to to track their performance, manage inventory and optimise revenue.
- The AdTech solution must offer a transparent payment process. Since advertisers construct varied pricing structures for their campaigns and may adjust how costs are passed on, the platform must clearly present all associated costs. This ensures that clients have full visibility into pricing breakdowns, their margin and cost allocations. By having a transparant payment process advertisers can see that the goes fair and is predictable.

While these requirements have been discovered in this research, there may be many more yet to be discovered for AdTech solutions. In Chapter 6, we discuss the limitations of this research and explore how these requirements might also apply to other software solutions beyond AdTech.

4.2.2 The AdTech Software Life Cycle

In Chapter 2 "Software Life Cycles" various standard software life cycles are discussed, each aiming to describe the life cycle of regular software. However, after conducting extensive research, we concluded that these conventional software life cycles such as the software life cycles from Rajlich and Bennett [37] and IEEE [23] fall short in capturing the unique characteristics and requirements of the AdTech software life cycle.

For instance, AdTech systems must handle a lot of traffic from the earliest stages of their life cycle. This is a trait that standard models do not account for. Additionally, the AdTech industry faces constant changes driven by factors such as market dynamics, regulatory changes and technological developments. Moreover, AdTech continuously evolves because of various external factors such as regulatory changes, competition and new technologies. These never ending changes place a much greater emphasis on the Evolution stage than in standard software life cycles.

Because the existing software life cycles for regular software are insufficient for AdTech, we propose a new software life cycle. Our software life cycle model is specifically tailored to address the demands and challenges of AdTech. Understanding these stages and their transitions can provide valuable insights for both business and engineering stakeholders.

Software Stages

By coding the transcripts seven distinct groups have been revealed, as described in Chapter 3 "Coding". Among these, four groups could be directly linked to an AdTech software life cycle stage: Initial Development of AdTech, Scaling of AdTech, Evolution of AdTech and Replacement of AdTech. These stages reflect a time-ordered progression of AdTech software, from its inception to its eventual transition or decommissioning. In Figure 4.2 we show our proposed AdTech software life cycle model. It shows the four stages with the transitions between them.

- **Initial Development**. Engineers develop the first version of the AdTech software. Additional software is also implemented on the publisher's systems to enable ad serving on publishers side.
- **Scaling**. The resource infrastructure gets configured to dynamically scale its resources to handle increasing traffic demands, maintain continuous uptime and accommodate significant impacts.
- **Evolution**. Engineers further develop the AdTech's capabilities to meet emerging user needs, evolving market trends, maintain a competitive edge and ensure compliance with regulations.

- **Replacement**. Outdated AdTech software is retired and replaced with a new system through careful planning to minimise disruptions.

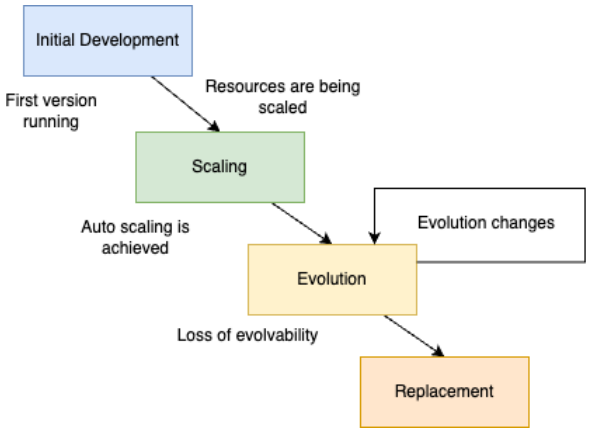


Figure 4.2: The software life cycle of AdTech

In the next sections, we will discuss the AdTech software life cycle stages in more detail.

1. Initial Development

In the Initial Development stage, the AdTech software is either created from scratch or developed using existing infrastructure, depending on the project requirements and available resources. During this stage developers need to ensure that the software can be simply integrated with multiple third-party systems. For example, publishers must implement the AdTech software directly into the code of their websites to ensure proper functionality.

Because not all publishers have a technical background as explained by Expert C, the AdTech needs to be simple to install. The low effort integration encourages broader adoption and reduces the likelihood of integration errors. Furthermore, the software is built in modular components. By starting with a modular architectures, components can be easily reused. Next to this, reduces it complexity for future updates.

Testing plays a critical role during the Initial Development stage. Testing of new functionalities is often conducted with AdTech users, such as agencies that have a strong relationship with the AdTech company. The development team could also choose to extended it to all users. Software updates are released frequently based on insights and testing. Software updates are released frequently based on insights and testing. By iterative development and testing the first version gets iterative improvement. This cycle continues until version 1.0 is finalised, providing a stable product that the sales team can confidently present to clients.

2. Scaling

After a first working version of the AdTech software is realised, scalability measures becomes crucial. A single client can significantly challenge the system's capacity. To tackle this challenge the development team needs to make sure the infrastructure can ensure reliable ad severing.

To meet these demands, the resource infrastructure is designed to scale dynamically. The resources should adopt to fluctuations in traffic and accommodating growth in real-time without performance degradation. Cloud-based solutions play a critical role in this process, because cloud-based resources often offer automatic scaling functions for resources. By leveraging these functions the needed capacity is made available instantly when needed.

Since AdTech systems are expected to operate continuously, downtime is not an option. Scalability measures prioritise reliability to ensure uninterrupted operation and to handle the varying demands of multiple clients.

3. Evolution

AdTech's environment constantly changes by factors such as new trends, regulatory changes, emerging users needs and technical changes. To stay relevant AdTech developers constantly need to evolve AdTech by adding new functionalities or changing existing functions.

The rapid environment of the AdTech demands also constant innovation. If AdTech fails to innovate it can result in outdated software that no longer meets client needs. Ultimately causing clients to look for alternatives. It could be said that AdTech in this stage is never truly finished, as the rapidly evolving AdTech industry demands constant updates and improvements to stay relevant, functional and competitive.

In addition to this, AdTech needs to keep up with technological advancements, regulatory changes and other external factors. If the software must ensure compatibility with the latest tools, platforms and comply with compliance standards. Regulatory adherence is essential to building trust with clients and avoiding legal risks.

3.1 Evolution Factors

In this study, the following evolution factors could be identified. The factors include external pressures but also internal demands that drive AdTech to continuously evolve to stay relevant.

- **Ad-blockers:** "One major factor is ad blockers. If they put us on a blacklist, then every Chrome extension that blocks ads will start blocking our requests." - Interviewee Expert D
- **Acquisition & Integration:** "Lot of AdTech will be sold to bigger companies and then integrated in to their own AdTech stack. It maybe will not always be completely replaced but integrated in other AdTech." - Interviewee Expert F
- **Competitors:** "If you don't innovate for a year, it can happen quickly. Due to competition, innovation from the rest of the ecosystem" - Interviewee Expert C
- **Customer Needs:** "Yes, for example, initially nobody asked for it, but now advertisers want click-through links in the article. Like if Brand X wants to sell its beer, they add a link, a user clicks it, we record an event, and Brand X can track it using UTM parameters." - Interviewee Expert D
- **Different Client Profiles:** "Initially, we were focused on one client, but eventually, as we implemented it with publishers, we realized they could also be clients, and the client definition became much broader." - Interviewee Expert C
- **Fast-Outdated Software:** "More than you think. It's never finished. I think maintenance is always more, a lot. Maintaining is one thing, but today's AdTech is already outdated." - Interviewee Expert C
- **Innovation of the AdTech Platform:** "That's where the innovation is—simplifying the branded content workflow. The product itself is already innovative in that it cuts out an entire flow by bundling everything into one platform." - Interviewee Expert D
- **Maintenance:** "If you collect a lot of data and handle high traffic, someone has to maintain it because, for instance, you can quickly exceed the amount of space on a disk or run out of memory. Many things can go wrong." - Interviewee Expert B
- **New Technologies:** "Because there are new technologies, you also have to connect to them." - Interviewee Expert C
- **Regulations:** "If regulations change, technical solutions also change accordingly." - Interviewee Expert A
- **Technical Changes Affecting the Ad Market:** "Technical changes, for example, browsers changing cookies." - Interviewee Expert A
- **Third-party Dependencies:** "AdTech usually integrates with many third-party companies. These APIs evolve, introducing frequent changes. You have to maintain them because they always give a migration period, but in the end, they can just disable some functionality you used before. So you have to keep track of it." - Interviewee Expert B

- **Trends:** "People follow trends in this market. Everyone wants the latest technology." - Interviewee Expert A
- **Younger Market:** "The market is getting younger, meaning younger people use our platform... We want the platform to be user-friendly for that demographic. Hence the UI/UX designer. So the design of our app might keep evolving, but on the back-end side, not much changes based on the market or client needs." - Interviewee Expert D

4. Replacement

If AdTech software becomes outdated because of factors such as new regulations, increased competition, lack of innovation, or other industry changes, then replacement comes in as a necessary step to stay competitive or to avoid potential losses.

When AdTech systems become trusted legacy systems with deep integrations into the clients systems, it can hinder evolvability. This forces AdTech companies to decide it is time for replacement. This replacement is often a difficult task, because of the many integrations AdTech systems have. Next to this, are there still running campaign on the existing system. To replace the systems, extensive planning is needed, any disruptions could result in significant costs. If the AdTech is developed in modular components, it can significantly reduce the complexity. Another way of replacing outdated AdTech systems is by transitioning client by client to the new systems. To execute this approach two AdTech systems needs to run at the same time. Transitioning clients to a new systems is also a slow process since the clients have running campaigns and integrations with the current system.

4.2.3 Stage Characteristics

During the software life cycle characteristics of the software and the development team change with each stage. This includes shifts in staff expertise, software architecture and economic considerations [37]. For AdTech, this also applies. In this paragraph, we discuss the characteristics that could be identified in this study.

Staff Expertise

During the Initial Development stage, staff need strong expertise in web development to create the first version of the software and integrate it with publishers systems (websites). In the Scaling stage, the focus shifts to building scalable software and data engineering. Experience with cloud technologies is a must, since it supports the development of scalable data storage and other infrastructure solutions. When the software enters the Evolution stage, staff must stay up-to-date with the latest technologies and AdTech trends. By staying up to date with the latest trends and technology, the development team is able to evolve the AdTech timely. Finally, in the Replacement stage, staff must possess knowledge of the legacy software to replace or update complex components without disrupting existing operations.

Software Architecture

The Initial Development stage is the most important commitment to defining the software architecture. This architecture will serves as the backbone of the AdTech system. During the initial development is the architecture is designed to support the the system's intended functionality and to handle later evolution. This architecture will significantly influence the ease of future evolution and providing a roadmap for further development efforts, which align with the principles discussed by Rajlich and Bennett [37]. Next to this, is the software architecture is favourably designed using a modular approach. By having a modular architecture updates and replacement can be more easily done without ad serving disruptions. In the Scaling stage, the architecture undergoes intensive upgrades in resources to handle increased traffic and data demands. This ensures the system stays robust and reliable. In the Evolution stage, the modular software architecture designed in the previous stages will be extensively used for updating and replacing existing components and adding new components to the architecture. Finally, in the Replacement stage, the architecture is gradually phased out, with certain components repurposed or reused in the development of replacement software.

Economics

By doing interviews with Expert A (Head of Tech) and Expert C (Sales Director), it is possible draw several conclusion about the economics of AdTech throughout its life cycle. In the initial Development stage, significant investment is required to create the first functional version of the software. Once there is a function version and the first publishers are on-boarded to the AdTech platform, agencies or direct brands can purchase advertisements. With these initial campaigns, the platform begins generating its first revenue.

If developed correctly, the AdTech system scales automatically during the Scaling stage, with costs balanced based on traffic. The difference between the campaign cost paid by the advertisement buyer and the cost paid to the publisher is the profit. This will be used to fund operational expenses and further development. In the Evolution stage, ongoing costs are directed toward implementing new features and adapting to market demands to maintain competitiveness. By the Replacement stage, costs decrease as the focus shifts to reallocating resources for the development of new software or elsewhere as needed.

4.2.4 Differences Between the AdTech and Software Life Cycles for Regular Software

The AdTech software life cycle is influenced by the work of Rajlich and Bennett [37]. However, what sets the AdTech software life cycle apart from traditional life cycles, such as those described by Rajlich and Bennett [37] and IEEE [23], are several distinct characteristics. While the initial development stage aligns closely with traditional life cycles, we found that AdTech software development requires installation within third-party systems. Installation with publishers' website is needed to create a fully functional first version. This requirement sets the Initial Development stage apart from the standard Initial Development stage.

In this study, we introduced a new stage: the Scaling stage. For AdTech software to become fully operational, it requires a dynamic and robust infrastructure capable of handling the immense demands of individual clients, who can generate millions of requests and data logs through a single campaign. For AdTech is reliability critical, since even minimal downtime can disrupt operations and compromise performance. Tackling these challenges is critical and have to be addressed before the software can progress to the next stage of its life cycle.

While the Evolution stage in the AdTech software life cycle aligns with the Evolution stage described by Rajlich and Bennett [37], the Servicing and Phaseout stages from Rajlich and Bennett [37] are not included in the AdTech software life cycle. Similarly, the same applies to the Maintenance stage from IEEE [23]. In the Servicing stage and Maintenance stage, minor repairs are made and small functional updates are implemented to maintain and stabilise the system. However, this is not applicable to AdTech software, which undergoes continuous development driven by evolution factors. If these factors are not enough tackled, AdTech software quickly becomes outdated.

Unlike in traditional software domains, there is no parallel maintenance of older software versions in AdTech. Instead, development efforts are focused on the latest version, which continuously evolves as we mentioned before. Consequently, there is no space in the AdTech software life cycle for stages dedicated solely to minor repair such as Servicing, as described by Rajlich and Bennett [37] and Maintenance, as defined by IEEE [23]—or for minimal activity focused on generating revenue from the system for as long as possible (Phaseout), as described by Rajlich and Bennett [37].

The final stage of the AdTech software life cycle, called "Replacement" aligns with the Closedown stage stage described by Rajlich and Bennett [37] and the Retirement stage from IEEE [23]. However, the final stage in the AdTech software life cycle places a strong emphasis on replacing the existing software with a new solution, as publishers cannot afford to lose advertising revenue during the transition. Due to this focus, we find "Replacement" to be a more fitting name for this stage.

4.2.5 Best Practices

In this section, we present best practices tailored to each stage of the AdTech software life cycle.

Initial Development

Develop AdTech in Components was identified during the exploration interviews. A recurring need came forwards to build AdTech using an modular approach. An example of this was provided by Expert A: "The market operates in cycles, for example: 20 years ago were the first ads, it's becoming more automated, then other sectors like retail [...] emerge. These sectors need to be rebuilt from scratch. So, AdTech can be reused. Therefore, it's wise to build in components"

Title	Develop AdTech in Components
Brief Description	AdTech systems should be developed in modular components to improve its reusability.
Additional Details	Developing in components makes faster iteration possible. Next to this, gives it more flexibility. Each component can be tested, updated, or replaced independently. This reduces complexity when scaling or evolving the software. This approach comes particularly in handy when the AdTech software can be applied in different domains.
Objective	Create modular software components that can be reused across different domains and AdTech systems, which improves adaptability and efficiency.
When to Apply	Start creating modular architecture as soon as the application's structure is defined. By starting with a modular based architecture is the architecture modular from the outset.
Cost Implications	Developing modular components may increase initial development costs. However, it will reduces future maintenance and adaptation expenses, which results in long-term cost efficiency.
Conditions for Successful Application	Successful implementation depends on the reusability of the developed components. When components can be easily reused in different AdTech software or in different domains, it can be concluded as a successful application.
Cautions	Avoid over-engineering components to avoid complexity. Additionally, documenting the components prevents integration issues in future stages.

Table 4.1: Detailed overview of the best practice: Develop in Components

Develop AdTech That is Simple to Implement was identified as best practice in response to a recurring challenge observed in the case study: the onboarding and configuring/installation of publishers. These issues caused ads being displayed multiple times on the same page. This problem aligned with the interview with Expert C: "Publishers often don't read all the documentation for implementation." Expert C further elaborated on the difficulty: "Often, the website owners aren't knowledgeable enough to implement simple things and get redirected to the next agency, which are usually small firms or agencies that aren't very quick or smart. This leads to going through many layers due to the lack of knowledge, which is very challenging." Expert C mentioned a possible solution for this problem: "Ideally, you should centralise control so you can set the pace and not depend on sixty different parties, as we experienced with dynamic branded content". While this solution is ideal, it is hard to realise in a real-world setting since the configurations need to be set on the publisher's website. Giving this level of control to an AdTech company can be too big of a commitment, as publishers need to accept the risk of potential downtime. A more realistic approach could be a more simple integration which fits all publishers systems: "If something works, it needs to be simple and applicable to many of your end-users."

Title	Develop AdTech That is Simple to Implement
Brief Description	AdTech solutions should be designed for effortless implementation. By minimising a complex implementation it should the reduce the technical burden of the publishers. An effortless implementation should reduce debugging efforts and helps with broader adaption.
Additional Details	A simple implementation includes minimal configuration steps, clear documentation and automated installation.
Objective	The objective is to ensure publishers can easily implement the AdTech software without needing any technical expertise. With the end goal of reducing setup time and dependency on third parties.
When to Apply	During the initial development stage, the development team should focus on making the implementation/installation as easy as possible. When new functionalities are introduced simplicity should remain a priority.
Cost Implications	Developing onboarding-friendly AdTech may require additional development effort. However, reducing implementation complexity decreases support costs, debugging and onboarding time.
Conditions for Successful Application	The best practice is successfully applied when publishers need minimal support with implementing the AdTech. The AdTech is as simple as possible to implement is a publishers system.
Cautions	The development team must caustion for oversimplification. Advanced users should not lose functionalities. Additionally, implementation guides must remain up-to-date as technology evolves.

Table 4.2: Detailed overview of the best practice: Develop AdTech That is Simple to Implement.

Scaling

Scale in the Cloud was identified as a best practice in response to one of AdTech's core technical challenges: handling significant amounts of real-time data without any disruptions. As described in Chapter 1, "Problem Statement", AdTech software needs to handle significant amounts of real-time data. To deal with this, AdTech companies need to find a solution for this challenge.

In the interview with Expert B: "I mean that for instance several years ago people tried to set up database on their own server but right now they prefer to use hosted solution that database it's something which is hosted by which is maintained by external company who is really take care of any parameters of this piece of infrastructure". To save costs, AdTech companies use automatic scaling functions in the cloud. "Most of it runs in the cloud and needs to scale automatically." - Expert A.

Title	Scale in the Cloud
Brief Description	AdTech should leverage the cloud for its automatic scalability function. By leveraging this function, resource overallocation and under allocation can be avoided.
Additional Details	Cloud providers can dynamically allocate resources based on real-time demand. By making use of the a cloud resources, systems can avoid over-allocation and under-utilisation.
Objective	To efficiently handle large amounts of real-time data, while minimising costs.
When to Apply	When workloads/traffic vary significantly, requiring dynamic resource adjustments particularly during real-time peak hours. Peak hours could happen during campaigns with significant reach, or when transitioning from static on-premises solutions to more flexible cloud systems.
Cost Implications	Reduces upfront investment and operational costs associated with resource overallocation but requires monitoring to manage costs effectively during peak scaling. Next to this, the potential for increasing cloud usage rates and dependency on providers could lead to high costs over the long term.
Conditions for Successful Application	A reliable cloud provider, applications designed for elasticity, proper cost monitoring tools, engineering team is skilled in cloud resource management and auto-scaling enabled in the cloud configurations to dynamically adjust resources as needed.
Cautions	Misconfigured scaling rules can lead to unexpected costs. Dependency on a single cloud provider may pose risks of vendor lock-in and long-term costs may escalate due to increasing cloud usage rates.

Table 4.3: Detailed overview of the best practice: Scaling in the Cloud.

While this best practice is particularly relevant to AdTech, it is not only applicable to this domain. Any software that experiences high traffic surges during peak hours or events such as campaigns should also apply this best practice to reduce cloud costs.

Evolution

Monitoring of AdTech Resources emerged as a best practice in response to the operational risks associated with handling high traffic and the storage of vast amounts of data. When handling large amounts of traffic and storage large amounts of data, an unexpected error could pop up. As one developer explained: "another thing is that if you collect a lot of data and you handle high traffic someone has to maintain it because for instance you can quickly exceed the amount of space on a disk. You can have not enough memory to handle this traffic. Many things can go wrong". To avoid these risks, it is essential for AdTech companies to monitor their system resources on computing power. This ensures potential issues are identified on time before they escalate into more serious problems.

Title	Monitoring of AdTech Resources
Brief Description	Setting automated notifications for important resource measurements such as CPU, memory and disk space to identify potential issues. Next to this, make automated notifications it possible to minimise monitoring.
Additional Details	Automated tools and systems can be used to configure alerts when thresholds for resource usage are exceeded. Notifications can be sent via email, messaging platforms, or system logs. By having these measures in place ensures immediately response to possible issues.
Objective	To maintain seamless operations under high traffic conditions by identifying issues on time and performance degradation.
When to Apply	The best practice should always be applied in systems with high traffic, large data handling, or complex operations that are resource-intensive. Monitoring of resources is also crucial when systems get scaled up or are in peak traffic hours.
Cost Implications	A monitoring dashboard and alerts costs minimal since many could providers provide these for free. On-premise solutions may have minimal associated costs.
Conditions for Successful Application	Reliable monitoring tools with correctly configured thresholds and notification channels. A response plan must also be in place to act on alerts promptly.
Cautions	Overly sensitive alerts can lead to notification that get ignored and where important alerts might be ignored. Proper calibration of thresholds is essential to avoid unnecessary disruptions or false positives.

Table 4.4: Detailed overview of the best practice: Monitoring of AdTech Resources.

Prioritisation of Development was identified when a recurring theme emerged during the interviews: AdTech constantly needs to innovate. As Expert C noted, "If you don't innovate for a year, it can happen quickly. Due to competition, innovation from the rest of the ecosystem." Similarly, Expert A highlighted the market's demand for constant innovation: "The market continuously wants new things; some even have different preferences. Clients will quickly switch to competitors if you don't offer the latest features that your competitors do. As long as there are no competitors, you can hold out for a long time. With standard AdTech, you're out if you don't innovate."

However, innovation alone isn't sufficient. Maintenance is equally critical to make sure functionality works, as Expert B explained: "Without maintaining, I think that this software will not work for a long period..."

With this being said, we could say that AdTech needs to innovate but also be maintained. So, AdTech systems must continually evolve. However, the challenge lies in determining which evolution factor is the most important and which requires immediate attention. To address this, development teams need

to make well considered decisions for the prioritisations.

Title	Prioritisation of Development Tasks
Brief Description	Software has many factors driving evolution, such as technological advancements, market demands and regulations. Prioritisation of development tasks is crucial to make certain development efforts focus on the most high-impact tasks. This process is similar to the process of sprint planning in agile methodologies.
Additional Details	Using frameworks like MoSCoW (Must, Should, Could, Won't) or impact-effort matrices during sprint planning can help teams prioritising tasks that align with the organisations goals.
Objective	To work on tasks which contribute the most to the organisations goals.
When to Apply	Prioritisation should be done during sprint planning, backlog cleaning, or any planning process. These are the moments where team comes together and decisions need to be made on which tasks to prioritise in development cycles.
Cost Implications	Effective prioritisation makes sure that the resources are assigned to the most impactful tasks. This will reducing the risk of wasted time and effort. Poor prioritisation, however, can lead to inefficiencies and missed opportunities.
Conditions for Successful Application	There is clear understanding is which tasks have priority. The prioritisation aligns with the organisations goal and there is collaboration between the stakeholders
Cautions	If the prioritisation is poorly executed, then the team will focus on low-impact tasks. This could lead to resource misallocation. Poor prioritisation could also lead to frequent changes in priorities can disrupt workflows and confuse stakeholders.

Table 4.5: Detailed overview of the best practice: Prioritisation of Development Tasks.

While this best practice is particularly relevant to AdTech, it is not exclusive to this domain. Any software that undergoes frequent evolution should also apply this best practice to address the most urgent development tasks.

Leverage CI/CD in AdTech emerged as a best practice to ensure robust ad serving during software updates and deployments. With AdTech it is top priority that ad serving functions well in all situations. To achieve this, AdTech development teams must adopt a robust strategy to minimise disruptions. One such strategy involves the use of automated tests. Expert A explained this: "In principle, we have automated tests. New features are first rolled out on staging and tested there. If they pass, they are rolled out to production. Later, maybe an acceptance environment is added". By utilising this approach development teams can make sure the latest commits are tested before they roll out further.

With this being said, Expert D explains the role of CI/CD (Continuous Integration and Continuous Deployment) in their organisation: "We do have CI/CD. ... If staging looks good, we merge into production and Vercel updates the production build. We have a similar process for the SDK..." With the SDK, Expert D is referring to the software development kit used for displaying and tracking ads. So, by using automated testing with CI/CD pipelines and using multiple stages, such as staging, acceptance and production, teams can streamline deployments, reduce risks and ensure seamless ad serving.

Title	Leverage CI/CD in AdTech
Brief Description	AdTech development teams should implement CI/CD pipelines to automate the software release process in AdTech. CI/CD pipelines ensures rapid, reliable and low-risk deployments.
Additional Details	CI/CD allows developers deploy commitments with confident, frequently, run automated tests and deploy in multiple environments such as staging, acceptance and production. All this with minimal effort. By applying this approach it ensures that ad severing functions without any disruptions or downtime
Objective	The goal is to reduce deployment risks, improve software stability and enable fast iteration cycles while maintaining seamless ad serving.
When to Apply	Apply CI/CD in AdTech when it is possible. It is especially a good integration when: there are frequent updates, minimising downtime is critical, or teams need a structured release process for ad-serving applications.
Cost Implications	Implementing CI/CD requires development/investment efforts. Developers may need training. However, it reduces long-term costs by preventing downtime, minimising manual errors and improving operational efficiency.
Conditions for Successful Application	CI/CD is successfully integrated when there are well-structured automated tests, there are multiple deployment environment such as staging, test and production and if it is integrated with a version control system.
Cautions	When the CI/CD pipelines are poorly created, it could result in deployment failures. Next to this, can a lack of rollback mechanisms or not enough monitoring lead to undetected issues in production.

Table 4.6: Detailed overview of the best practice: Leverage CI/CD in AdTech.

Replacement

Replacing AdTech in Components was identified as a best practice to manage the complexity of replacing legacy systems. Replacing AdTech system can only be done without disrupting ongoing operations. Replacing AdTech systems is a complex process due of running campaigns with significant reach. This also involves sales houses and ad agencies who prepare new campaigns. According to Expert A, replacing AdTech requires careful consideration: "For example, with an ad server, it's very complicated. You can replace it piece by piece or build something alongside and gradually switch clients over a long period. You can't just stop." Additionally, Expert B explained that extensive planning is needed to replace a system "It depends on how this software is built and what you currently have. It's usually a topic for many meetings to plan the replacement of something with something else." With this being said, we could say it is important to have a phased strategic approach.

Title	Replacing AdTech in Components
Brief Description	The replacement of AdTech software is a complex process because of the many dependencies. To ensure a gradual transition without any disruptions, a phased approach should be applied. This involves replacing software component by component.
Additional Details	To apply this approach the transition team requires a detailed overview of the current AdTech system with its dependencies between components. It also involves building or integrating new components, alongside the existing system. Continuous testing is critical and needs to be done to make sure everything functions.
Objective	To replace existing AdTech systems without causing disruptions to ongoing campaigns or affecting end-users.
When to Apply	When building the system from scratch is to expensive or when a full system has lost its evolvability.
Cost Implications	Incremental replacement can spread the costs over time. However, it may involve higher initial investment. The planning and coordination of the transition efforts could increase operational costs.
Conditions for Successful Application	All software components are replaced without disrupting ad serving or impacting the systems end-users. All the stakeholders know the priorities and there is clear understanding about the replacement.
Cautions	Replacing components one by one requires careful management of dependencies. Poor planning or execution can lead to operational downtime or integration issues. Next to this, should enough resources be allocated for testing and debugging during the transition.

Table 4.7: Detailed overview of the best practice: Replacing AdTech in Components.

4.3 About the Best Practice Catalogue

This best practice catalogue consists of seven best practices, with each stage of the AdTech software life cycle corresponding to a specific best practice: Initial Development (2), Scaling (1), Evolution (3) and Replacement (1). These best practices are derived from exploratory interviews and a case study on branded content. This best practice catalogue serves as starting point for establishing best practices in AdTech development, many more best practices could still be discovered. This will be further discussed in Chapter 6, "Limitations". Further research is needed to expand this catalogue, as discussed in Chapter 7, "Future Work".

Chapter 5

Evaluation

The evaluation of the artifact utilises TAM by Davis [13]. The TAM principles assess its usability, relevance, acceptance and effectiveness. By using the principles we can determine whether the proposed life cycle model and best practices align with the needs of AdTech practitioners. To gather feedback, a mixed-methods approach was employed: quantitative data was collected through a questionnaire, while qualitative insights were obtained via workshops and in-depth interviews. To avoid overloading stakeholders, the best practices were assessed primarily through qualitative in-depth interviews. Meanwhile, the software life cycle in AdTech was evaluated during a workshop using a questionnaire that included both open-ended and multiple-choice questions.

Pilot
Before the full demonstration and the administration of the questionnaire, a questionnaire pilot was conducted to ensure its clarity and comprehensibility. This pilot involved an AdTech campaign manager who provided valuable feedback on the formulation of the questions. The manager pointed out that, in the questions, the software life cycle model appeared to be presented as a tool for development, whereas it should instead be viewed as a model that helps in understanding the different stages of AdTech software. The involving questions were adjusted to incorporate the feedback.

5.1 Questionnaire Results

A total of 8 individuals with expertise in the AdTech field have participated in the workshop and completed the questionnaire. Table 5.1 represents the average rating of each criterion of the TAM of the AdTech software life cycle. Table 5.1 shows an average scores of 4.63 for comprehensibility. Indicating that the software life cycle model of AdTech is easily understandable. One reason for the high score, as noted by Interviewee Expert H, is that "The graph and the description of the life cycle of AdTech software make it very easy to understand". However, some feedback suggested to provide real-life examples to make the model more easy to understand.

The next criterion of the TAM is Completeness, for which the artifact scored 4.38. Feedback from Expert D noted, "The steps are complete. We don't phase out software or keep it running without doing any work to it. The reason for that is a fast-changing environment where rules are constantly changed." This highlights the distinctive traits of the AdTech software life cycle. However, there was also feedback suggesting that more details could be added: "The model itself is more a 'helicopter view' with all stages. Per stage there are quite a few things to keep in mind that are not on there but as a 'helicopter view' it is complete".

The Ease of Use criterion scored slightly lower, with an average score of 4.00. Expert H shared, "I think it covers the software life cycle very well so it can be used as a good reference for work on other AdTech software projects." Expert F provided feedback suggesting that, "I think it is good to understand the phases and know in what phase you are at the moment. But the real insights are in the 'steps' in the phase. For example, what are the reasons for the evolution changes?" This feedback noted the importance of diving deeper into the specifics of each stage. While the workshop did not include all details from each stage, these can be found in Chapter 4, "The AdTech Software Life Cycle".

The criterion Correctness scored the highest with a score of 4.75. While all 7 participants agreed that the model with the context was correct, Expert F had an important note “I think it is correct. One thing to note maybe at the replacement stage. Lot of AdTech will be sold to bigger companies and then integrated in to their own AdTech stack. It maybe will not always be completely replaced but integrated in other AdTech”.

The comment aligns more closely with the concept of evolution rather than a direct replacement. The replacement stage must represent the final phase of an AdTech software's life cycle. In the final stage the system should be fully phased out. However, when AdTech is integrated into a larger AdTech stack, it often does not follow a strict replacement path. Instead, it adapts and evolves to fit within the new system and further evolves in the new system.

Another possible scenario is that the acquired AdTech is fully absorbed into another system, which could mark the end of its software life cycle. To investigate this further and validate how such cases should be represented future research is needed, which is also discussed in Chapter 7. For now, this study addresses Expert F's feedback by incorporating this perspective as an evolution factor within the Evolution stage, as described in Chapter 4.

The artifact scored a 4.50 for Usefulness. The participations found it useful as a reference when explaining AdTech software. As Expert D states, “As I said, the model correctly portrays the life cycle of a bunch of software I have worked on. Every project I work on in AdTech shows the same characteristics as the model, so I would say it is easy to understand”. Similarly, Expert A notes, “Pretty easy, we would probably not just apply this to the product as a whole but also look at individual components - to hopefully not have to deprecate everything at once.” This comment suggests that while the model is applicable, it is particularly useful at the component level, because AdTech platforms do not necessarily become obsolete all at once but rather in stages.

For the final criterion scored the artifact a 4.00. Adoption Likelihood refers to the expected real-world use of the artifact. For the AdTech software life cycle model, we can expect that the model will not be used frequently in day-to-day operations. However, it can be useful for educational or insight driven purposes. As Expert D states, "I'm not sure if I would ever use this, but if we get a new colleague or have to present our software to developers, it makes sense to throw this model in to show the development progress." This suggests that while the model may not have a lot of practical applications, but it can serve as a useful model for onboarding new colleagues. So, for explaining the development process of AdTech software the model could be a useful tool.

The model scored an average score of 4.38 out of 5. With this score, we can conclude that the AdTech software life cycle model is well-received. The model scored the highest for its correctness, comprehensibility and usefulness as a reference. The scores for the ease of use and adoption likelihood suggests that its practical usage may be more situational. The model could serve as an educational or explanatory tool rather than a frequently applied framework in day-to-day operations.

Table 5.1: Average scores for TAM criteria of the AdTech software life cycle.

Criterion	**Average Score (1-5)**
Comprehensibility	4.63
Completeness	4.38
Ease of Use	4.00
Correctness	4.75
Usefulness	4.50
Adoption Likelihood	4.00

Criterion	Average Score (1-5)
Overall Score	**4.38**

5.2 Results of In-depth Interviews

To validate the best practices and the AdTech software life cycle, four in-depth interviews were conducted with AdTech experts. For each best practice, five questions have been asked to assess comprehensibility, correctness, acceptability, effectiveness and efficiency and the likelihood of adaptation. The full interview guide can be found in Appendix D. The most interesting and thought-provoking quotes from the interviewees have been included in this section.

5.2.1 Develop AdTech in Components

The best practice 'Develop AdTech in Components' was overall well received. When we asked Expert A whether this is a good best practice for AdTech development, he responded with:

"In fact, it's not just a best practice – it's an absolute must. It's a must, yes. Of course, there may be other ways to approach it, but I've seen quite a lot, including some very large tech companies. I know that at some point, they cease to exist. That may be because they don't follow this best practice." - Expert A

This underscores that this best practice should always be applied. If the best practice is not applied can have serious consequences. However, Expert B provided an important side note regarding the best practice:

"Yes, but it depends, of course, because in our case it depends how big the company is and how big the development team is. Because yes if the team is big then it will probably be an effective way of development because they can focus on separate parts of that system. But it depends if your business will succeed. because you can push too much efforts at the beginning. you haven't verified yet what should be exact shape of your application" - Expert B

This is a valid point. Starting with component-based development without first defining the overall structure can make it difficult. To address this, the 'When to Apply' field has been refined to indicate that this best practice should only be applied once the application's structure is clearly defined.

Expert H highlighted another benefit of this approach:

"It is good to focus on making it modular so we can adjust it in the future. Because as with everything concerning with software there are a lot of parts that usually tends to change and if we build it too tightly it will be very hard to update it in the future." - Expert H

This highlights the risks of not adopting a modular approach. Meaning that the best practice particularly fits the fast-evolving AdTech environment as discussed in Chapter 4. Expert J provided a real-world example of the consequences of not following this best practice:

"We were transitioning from old monolith system on AWS where all components were on one singular instance and if any of one of them would need to be scaled up everything would need to be scaled up. . . .and with a modular approach each component is scaled on its own. Yeah, it's incredibly better. We saved a lot of money on that." - Expert J

Additionally, Expert J highlighted another advantage of working with components: "I can see from the development team that it's much easier for them to find mistakes when everything is modular even though they have monolith repository but finding errors is much easier that way." - Expert J

With this being said, we could say that breaking the system into smaller, independent components, debugging becomes significantly more efficient and reducing development time.

5.2.2 Develop AdTech That is Simple to Implement

The best practice 'Develop AdTech That is Simple to Implement' highlights the importance of making AdTech easy to integrate. A simple integration ensuring a smooth onboarding process for publishers. Expert A clearly recognised this benefit:

"The simpler and more step-by-step you can describe what needs to be done for implementation, the easier it is to scale. You can't expect someone handling the implementation to deal with something overly complex, because these are people focused on media and their business, not necessarily hardcore tech or coding." - Expert A

Onboarding publishers who lack technical expertise can be time-consuming. To understand how valuable this best practice is for improving efficiency, we asked Expert B:

"Yes, if they don't bother you because everything is clear for them, it's beneficial for you. You don't consume much more time on helping them." - Expert B

This means that simplifying the integration process also saves time when if publishers can implement the software independently without requiring assistance. Expert H explained that a simple implementation process not only reduces complexity but also saves time:

"Yes, I believe so. Like most things, software is designed to simplify processes. In the context of AdTech, the goal is to streamline complex processes without limiting the range of options available to clients." - Expert H

Next to this, pointed Expert J out that when software is easy to integrate, publishers are more likely to implement it:

"It's correct most of the clients are not very technical so the easier you do it for them the more likely they are to implement your project product" - Expert J

This results eventually in the growth of the number of adopted publisher, making it a key factor in scaling AdTech solutions.

5.2.3 Scale in the Cloud

The best practice 'Scale in the Cloud' was not specifically defined for only the AdTech field. However, Expert B noted that it is relevant in this field:

"I think that it's happen very often. It's a standard in this business." - Expert B

Expert A provided a real-life example of why this best practice is crucial. The utilisation of the best practice is particularly valuable when AdTech platforms experience sudden traffic spikes from external sources:

"For example, if the NU.nl app or another news app sends out the same push notification about a news event, you can sometimes see the total traffic in a country multiply by five—just within a minute. And it doesn't last very long either. These are massive spikes. Yeah. So you pretty much have to have auto-scaling enabled." - Expert A

Expert H notes the importance of cloud-based solutions over on-premise setups by pointing out the challenges of scaling in an on-premise environment:

"So it's very good practice to build in cloud than to try to set up it on premise which can incur a lot of costs up front that won't be utilised at start and it also prevents horizontal scaling at first so it's very easy to buy just a server and set it up somewhere and pay for the storage. But when the demands skyrocket because we have higher campaigns or more traffic to our system then it's very hard to adjust the server. and also if you design the app for the single server it tends to be hard to update it in the future for horizontal scaling like you can do from the start on the cloud. " - Expert H

Expert J further describes the dynamic of scaling in the cloud. Expert J explains how their infrastructure adjusts based on demand:

"In the night is the infrastructure is scaled way down, however not the database, but the kubernetes everything there is scaled way down and during the day and during the ad campaign we can see that it scale 10 times or more." - Expert J

With this being said, we can say that by applying best practice you can ensure efficient resource allocation. It allows systems to dynamically scale in response to fluctuating traffic demands while minimising unnecessary costs during low-traffic periods.

5.2.4 Monitoring of AdTech Resources

This best practice was created to help AdTech developers with preventing their infrastructure faces any downtime. Expert A explained the potential consequences of failing to implement the best practice 'Monitoring of AdTech Resources':

"If your ad server goes down, your clients don't make money and if your clients don't make money, you have a problem. It's also bad for reliability and reputation. So if something happens, you want to be the first to see it and inform your clients." - Expert A

Expert B agreed. The Expert highlighted that monitoring also helps with finding bottle necks in advance:

"Yes, definitely. it's essential to know that everything is up and running and the campaigns are delivered and clients can use your system or it's also important to see where are bottlenecks of a system to take action in advance. " - Expert B

Expert H highlighted that monitoring is not just about system reliability but also plays a crucial role in optimising cloud costs:

"Monitoring and maintaining system goes hand in hand. Especially if we have a system that tends to scale pretty quickly and with all of the software projects especially that are run on cloud we need to monitor it for the costs because it's easy to incur a lot of costs in cloud." - Expert H

Expert J, who actively applies this best practice, explained how monitoring alerts help them take action when needed:

"When the new campaign launches, we can see that queues are rising and the system slowed it can be slow down and and sometimes if the campaign is really big and we didn't set the limits properly then we get alerts and scale it manually enabled it to scale automatically." - Expert J

Based on this, we could say that the best practice ensures stability, proactively addresses issues and optimizes performance. Which ultimately saves costs and prevents any downtime.

5.2.5 Prioritising Development in AdTech

When we introduced the best practice 'Prioritising Development in AdTech', Expert A's initial response was:

"Okay. I think this is super important. I don't know if... I don't know to what extent this is specific to AdTech compared to regular software." - Expert A

This is a valid point. The best practice was originally defined as an AdTech-specific best practice because AdTech deals with various evolving factors that may need to be prioritised based on importance. However, the best practice may not be only applicable to AdTech. Because of the strong point of Expert A has the best practice been adjusted to also apply to software development in general.

To further evaluate its usefulness, we asked Expert B:

"It's very useful best practice., I think that it's happening always. they have a dedicated role roles to ensure that this process is conducted in a right way such as scrum masters." - Expert B

Expert B emphasised the importance of prioritisation tasks in software. By assigning scrum master to execute this best practices is not a strange though since they have similar tasks. Expert A also pointed out that the best practice applies beyond AdTech. Expert H reinforced its strong relevance to AdTech specifically:

"Yes, it's a very good practice since we need to prioritise business-focused tasks. AdTech is a highly dynamic environment that changes rapidly, which means system requirements can shift just as quickly. Because of this, it's crucial to focus on the most important tasks that keep the system running smoothly and ensure it remains competitive with other systems in the market." - Expert H

Expert J agreed on the importance of this best practice but pointed out the challenges in implementing it:

"I think yeah, planning is the foundation, but I think it's one of the hardest one to actually implement because planning is when you have a lot of moving parts and A lot of things can go wrong. . . .when migrating to platform some new obstacles could appear and just mess up your plan if you're not careful about it. " - Expert J

This highlights that while the best practice is essential, executing it can be a challenges when software behaves new obstacles appear.

5.2.6 Leverage CI/CD in AdTech

This best practice was created to ensure developers can confidently and frequently deploy their latest commits to environments such as staging, acceptance and production. Expert B emphasised on the importance of using CI/CD to prevent downtime:

"It's prevent you from breaking production environment. and especially when you have a tests incorporated in your pipeline, you can be quite sure that on production it will be also fine that you will not expect some unpredictable behaviour." - Expert B

As a Head of Tech, Expert A also sees CI/CD as a valuable practice in AdTech:

"It takes the tension off." - Expert A

The comment from Expert A was an unexpected benefit of integrating CI/CD. Integrating CD/CD pipelines not only helps software developers with deployments and debugging but it also gives managers more confidence in releasing new software updates.

Expert H agreed by mentioning that AdTech operates in an rapidly evolving environment:

"Yes, it's good practice because we need a constant pipeline of changes, especially given the rapidly evolving environment mentioned earlier. Its good to have a process that automates system updates, ensuring everything runs smoothly without introducing bugs to clients. Therefore, having a configured CI/CD pipeline is important for AdTech development." - Expert H

Expert J noted that the 'When to Apply' field needs a slightly adjustment. Expert J argued that CI/CD should not only be integrated when frequent updates are required but should instead be implemented as early as possible:

"It now stands that when to apply CI/CD in AdTech when frequent updates are required. I wouldn't agree with that. I think that CI/CD should be implemented whenever you can because manual updates can be tricky and can lead to some issues down the road even if they are in not frequent you do it once half a year if something change." - Expert J

Expert J makes a strong point. Why should you wait until the software requires frequent updates when implementing CI/CD earlier can save development time and simplify debugging? Based on this feedback, we have adjusted the 'When to Apply' field accordingly.

5.2.7 Replacing AdTech in Components

According to Expert A, this best practice is an absolute must. Without replacing AdTech in components, the entire system would need to be replaced all at once:

"Yes, it's almost the only way to replace AdTech. Yeah, you could say it like that. Then it's also the only correct way. Of course, if you don't break it down into smaller pieces, you'll have to do it in one big change. Yeah, it's very difficult, especially because you can't afford those disruptions in your service." - Expert A

Expert B also agreed by emphasising on the importance of ensuring the system is stable when you replace it:

"Yes, as you mentioned here it's very important to not break, to be able to still deliver content and at the same time do some migration of a part of your system. So in AdTech it's essential to not break the system during such operation. So generally we should follow this rule. " - Expert B

Building on the best practice 'Develop AdTech in Components', Expert H highlighted that if an AdTech system is already designed in components it also be replaced in components. He also stressed the importance of understanding software dependencies:

"Initially, we mentioned that having a component-based system is highly beneficial for maintenance and adapting to changes in a dynamic environment like AdTech. During the system's evolution, replacing outdated components becomes crucial, as they can become stale and no longer align with business needs. It's important to identify these components, understand how they integrate with the rest of the system and then design an effective approach to update or replace them with improved versions that offer enhanced functionality and better meet current requirements." - Expert H

Expert J shared a real-life experience of replacing AdTech infrastructure. In this example, he faced an AdTech infrastructure which could only be replaced in all at once:

"I think it can be done gradually, step by step, but in our use case—when we had to migrate to another cloud—it wasn't so easy. It wasn't gradual, However I think the best practice is correct. We didn't stick to it, but I think it's correct." - Expert J

This highlights that while gradual migration is ideal, real-world constraints sometimes make full-system replacements unavoidable. That ultimately reinforced the need for a well-structured component-based approach.

Chapter 6

Discussion

In this chapter, we revisit the predefined hypotheses and challenging them with our findings. The following is a discussion of the relevance and applicability of the key findings. Finally, we explore the broader implications of the study and address its limitations.

6.1 Hypothesises

In Chapter 3.4, we presented the hypotheses for this study, which explore the research question: "*How can the standard software life cycle model be adapted to fit AdTech and which best practices help bridge the gap in existing guidelines throughout this cycle?*" The hypotheses are as follows:

1. $\mathbf{H_1}$: The software life cycle of AdTech is shorter and more iterative compared to the standard software life cycle, due to the fast-paced advertising industry.
2. $\mathbf{H_2}$: AdTech must ensure a reliable demand from advertisers and other consumers before development begins, as its robust infrastructure requires significant upfront costs and resource investments.
3. $\mathbf{H_3}$: The software life cycle of AdTech does not differ significantly from the standard software development life cycle, although it requires specialised best practices for managing privacy regulations.
4. $\mathbf{H_4}$: The software life cycle of AdTech does not differ significantly from the standard software life cycle.

The hypotheses aimed to explore how the standard software life cycle model can be adapted to fit AdTech and which best practices help bridge the gap in existing guidelines. However, based on our research findings we can't accept any of the defined hypothesises.

Regarding $\mathbf{H_1}$, although AdTech operates in a fast-paced environment, the study does not conclusively confirm that its software life cycle is shorter. This study found that the development process is highly iterative to keep up with evolving industry factors, but the findings do not conclusively indicate that its life cycle is shorter.

With $\mathbf{H_2}$, the assumption that AdTech requires a guaranteed demand and funding needs to be secured for the development begins. This was not supported by our findings. While development can cost a lot, the study did not reveal strong evidence that upfront certainty is always necessary.

The results for $\mathbf{H_3}$ indicate that the AdTech software life cycle differs significantly from the standard software life cycle. The AdTech software life cycle contains a new stage, Scaling. Next to this, plays the Evolution stage a much larger role in AdTech development. Additionally, the Servicing and Phaseout stages are not included in our model. All differences are discussed in greater detail in Chapter 4. While best practices were identified, they were not primarily related to managing privacy regulations, as we initially expected. Instead, the study steered us towards developing development best practices specifically tailored to AdTech's software life cycle, because, these kind of best practices did not exist yet, but could be very valuable for AdTech developers.

Finally, $\mathbf{H_4}$ aligns with $\mathbf{H_3}$ and does also not hold, as the research led to the identification of a distinct software life cycle for AdTech.

6.2 Applicability of the Artifacts to AdTech

Chapter 2 introduced the AdTech stack illustrated in Figure 1.1. This technology stack served as a foundational reference throughout this research. The research focus was specifically on AdTech software in this stack that played a role in the ecosystem of serving ads. The full AdTech ecosystem is shown in Figure 2.1. By examining only the technologies within this stack, the research maintained a focused scope. Because of this focused scope the study could pursue more in-depth analyses. This depth was necessary to identify the characteristics of AdTech beyond the already known attributes of the industry.

The experts who participated in this research had experience with the software from this AdTech stack and related technologies. Their insights contributed to a clear understanding of the ecosystem. However, since they did not have experience with every AdTech solution available on the market. This results in limitations in the reliability of our findings. These constraints, along with their potential impact on the study's conclusions are further discussed in the Limitations section.

6.3 Applicability to Other Domains

To explore whether the AdTech software life cycle could be reused for the MarTech domain, an exploratory interview was conducted with Expert E. According to Expert E, consists the MarTech stack out of six categories: Content and Experience, Social and Relationships, Commerce and Sales, Data and Management. The last two categories Data and Management are primarily focused on measuring ROI [9]. Within this structure, AdTech falls under the Content and Experience category of MarTech explained Expert E. While this insight provides a useful perspective, it should be considered that this perspective is based on a single expert interview. Further research needs to be done to validate this perspective.

That said, it is unlikely that the AdTech software life cycle applies to all MarTech categories. Expert E explained that each category serves a different purpose. While the Content and Experience category focuses on ad serving, other MarTech categories consist of tools designed for functions like data visualisation, Agile & Lean management and other specialised tasks. The purpose of ad serving inherently brings specific software characteristics, such as the AdTech traits described in Chapter 4. These characteristics are not typical of software like data visualisation software or task management software.

Additionally, the end-user requirements in AdTech differ significantly from those in MarTech. AdTech users demand solutions that are easy to integrate, transparent and simple, because the ad environment is already complex. In contrast, Expert E explained that MarTech users typically seek solutions that address their marketing challenges.

"MarTech end-users have fairly low demands and are quick to settle for something that doesn't work... CEOs and CFOs see software and think it will solve everything." - Expert E

A reason for this difference lies in the distance to the marketing budget. Campaign managers are often the users of AdTech software, which makes them directly involved with campaigns budget allocation and optimisation. This leads to a stronger focus on efficiency and ROI, as Expert E explained. On the other hand, marketing managers are the primary users from MarTech solutions. They often operate further from the immediate budget concerns, leading to a preference for tools that they think it will solve their marketing challenges.

Nevertheless, the AdTech software life cycle could apply to software in the Content and Experience category within the MarTech stack. This category consists of various subcategories, including Display & Programmatic Advertising, Mobile Marketing, Native/Content Advertising, PR, Print, Search & Social Advertising and Video Advertising [9]. This is possible if the software that falls under these categories shares similar characteristics with AdTech, as described in Chapter 4.

6.4 Best Practices Principles

In Chapter 2, four principles were selected to define best practices: Replicability, Effectiveness, Sustainability and Innovativeness [49]. For a practice to be considered 'replicable', it must be able to do it exactly the same way as before, or produced again to be exactly the same as before [49]. This aspect is addressed by incorporating the "When to Apply" and "Conditions for Successful Application" fields in

the best practice table, which inform the reader about the necessary steps and appropriate circumstances for implementation.

The next principle is Effectiveness. To meet this criterion, practices must meet the expected goals or outcomes [49]. To assess this principle, we included questions in the validation interviews about the effectiveness and efficiency of the best practices for improving software development. Based on the results of the interviews, we could conclude that most best practices were effective, with some requiring only minor description adjustments to fully reach their potential effectiveness. Chapter 5 Evaluation, discusses how this was achieved.

The third principle is Sustainability. To meet this criterion, best practices has the quality of being able to continue over a period of time [49]. To meet these requirements the best practices have been formulated to be broadly applicable, ensuring they are not tied to specific technologies, platforms or tools. By formulating the best practices with a focus on accessibility, they can be applied in various contexts. Next to this, is there no prior knowledge of a particular technology needed to adapt the best practice, while also not creating any additional dependencies.

The final principle is Innovativeness, which assesses the novelty of the measures and mechanisms developed [49]. To assess the Innovativeness principle, the best practices were validated based on their adaptability in AdTech development. The validation interviews revealed that several practices were already widely adopted and well-known. However, others were recognised as new not because they introduced entirely new ways of working, but because they defined and formalised approaches that practitioners were already applying intuitively.

6.5 How the Artifacts Address Key AdTech Challenges

The solution objective of this research was to develop a framework, which address the AdTech's challenges and provides actionable insights for practitioners of the AdTech field. Such insights could help professionals navigate the life cycle of AdTech more effectively. To achieve this objective, we created an AdTech software life cycle model and best practices. These artifacts address AdTech's challenges and focus on the demands of AdTech such as real-time traffic management, a continuously evolving environment and dynamic scalability.

To meet these requirements, the artifacts were specifically designed to meet these demands. The AdTech software life cycle was adapted to reflect these challenges. AdTech software life cycle introduced a new stage called "Scaling". The Scaling stage focuses on preparing infrastructure to handle significant amounts of real-time traffic and dynamically scale to the demand. Additionally, we included the "Evolution" stage, as introduced by Rajlich and Bennett [37]. The Evolution stage plays a more prominent role in the AdTech software life cycle. In our software life cycle this stage focusses on continuous development. This is necessary due to the changing environment AdTech operates in; this requires software solutions to evolve continuously.

The best practices were also designed to tackle the industry challenges and meet the AdTech demands. For example, the best practice "Scale in the Cloud" suggests leveraging cloud-based solutions. By utilising cloud resources AdTech can dynamically scale up to meet demand and scale down when resources are no longer needed, which results in lower cloud costs.

The best practice "Develop AdTech in Components" recommends designing and developing AdTech systems in modular components. By developing AdTech modularly, AdTech solutions can easily be updated and replaced without having the difficulties of dealing with many dependencies. By having a modular architecture, AdTech can adapt faster to the rapidly evolving environment.

To onboard many publishers, a simple integration process is critical. The best practice "Develop AdTech That is Simple to Implement" addresses this challenge by promoting straightforward implementation where there is no technical expertise needed to install the AdTech. By having a more straightforward implementation the barrier to enter lowers, which ensures broader accessibility.

The best practice "Monitoring of AdTech Resources" emphasises the importance of continuous monitoring of system performance and resource usage. By monitoring AdTech resources or setting notifications about resources performances. By utilising monitor tools stakeholders can identify bottlenecks and more easily identify issues during high traffic.

AdTech operates in a fast paced environment and to stay relevant it must adapt fast. To do this, development efforts must remain aligned with shifting priorities. The "Prioritisation of Development Tasks" best practice supports this by suggesting to prioritise the development tasks. Prioritisation of tasks ensures that the most urgent and impactful tasks are done first. So, by applying this best practice development teams can continuously evolve the system.

To streamline software deployment and minimise downtime, the best practice "Leverage CI/CD in AdTech" is created. The best practice emphasises on using CI/CD processes to deploy more frequently and give developers and managers more confidence with deploying the latest updates. Next to this, include CI/CD pipelines automating tests which help with developing a robust release.

Replacing AdTech is a complex task, since developers must ensure the systems doesn't experience any downtime while replacing it. The best practice "Replacing AdTech in Components" focuses on structuring AdTech solutions so that individual components can be replaced without risking any downtime. By applying a modular replacing approach systems can easily be upgraded without the need of a temporary shutdown during transitions.

6.6 Limitations

While this study has provided valuable insights into AdTech development and its software life cycle, a qualitative study also faces certain limitations. There could potentially be a interviewee selection bias. In addition, had this research a relatively small number of validation interviews and a small group of studied AdTech solutions. Next to this, had the research a limited scope, which may have restricted the range of findings. All of this could have affected the generalisability of our findings.

The biggest limitation of this research was the challenge of recruiting experts in the AdTech field for interviews. A total of 10 AdTech experts participated in the study. While AdTech is a specialised domain, which definitely plays a big role in the global digital economy you could say that the pool of worldwide relevant experts is relatively large. However, this study relied on a convenience sample. This means that the researcher only involved interviewee's which are located in the Netherlands or within the researchers reach. As a result, the findings may reflect regional practices and perspectives. With this being said, we can say that the findings not fully capture the global diversity or explored all AdTech solutions. This limitation potentially affects the generalisability and comprehensiveness of the research results and introduces a bias toward Netherlands-based professionals.

The exploratory phase of the research included five exploratory interviews and a case study conducted over a period of 2 months. The resulted findings are a AdTech software life cycle and seven best practices. In the exploratory interviews, saturation was observed after speaking with three experts who had direct experience developing AdTech. However, the interviews with a MarTech expert and a Sales Director introduced new perspectives.

The scope of this phase could have influenced the outcomes. A different software life cycle might have emerged if the interview group had been larger or included of different participants. By selecting interviewee's from the Netherlands and within our reach we introduced the possibility of selection bias. All the interviewees in this study had experience working with Dutch AdTech and their perspectives may have been shaped by the context of the Dutch market. So, our regional focus have impacted the applicability of our findings in other regions. This limits the generalisability of our findings. Additionally, if we had interviewed a larger group or more diverse in terms of expertise and geographical background, we may had different findings. So with a larger and more divers interviewee's, it is likely that more best practices would have been uncovered or a different software life cycle could have been modelled.

The validation of the artifacts was also limited by the scope of the study. The AdTech software life cycle model could only be validated by the participants from the workshop. The validation could only be done in a workshop, as the model required extensive details and a proper explanation of the conducted research. Had this not been necessary, a larger audience could have assessed the model. However, even the potential for a larger audience was then again limited by the reach of AdTech developers. For validating the best practices, an in-depth interview approach was chosen to meet the qualitative validation requirements. A quantitative validation approach was not pursued because of the limitations in finding a sufficient number of AdTech developers. This restricted us the ability to gather the data needed for such an approach. This decision was made to ensure that the validation process was still valuable despite the constraints in participant availability.

Additionally, all workshop participants had a personal connection with the researcher. This could pose a potential threat to validity. Next to this, could the relationship have influenced the TAM scores, introducing another bias. However, to reduce the risk of this bias, participants had to justify each score they provided.

Another limitation lies in the group of AdTech software studied. In the case study, AdTech from Adswag was examined. Their software is a combination of a DSP and SSP, which means that it facilitates both the selling and buying of ads, as well as ad serving. However, the interviewees had experience developing a variety of AdTech solutions. Their experience ranged from Adservers to SSP-only or DSP-only systems, as well as combinations of these. Despite this broad experience, it cannot be excluded that the AdTech software life cycle and the best practices may not apply universally to all types of AdTech software. It could also only apply partially to specific components of AdTech systems.

Chapter 7

Conclusions

7.1 Answers to the Research Questions

In Chapter 1, we defined three research questions that collectively address the main research question: "*How can the standard software life cycle model be adapted to fit AdTech, and which best practices help bridge the gap in existing guidelines throughout this cycle?*"

These research questions were explored and answered in previous chapters. In this section, we provide a summary of the answers to the research questions and highlight the key findings.

7.1.1 RQ.1 How can the AdTech software life cycle be modelled accurately?

Based on the outcomes from the interviews, we concluded that standard software life cycle models do not meet the requirements of AdTech. Therefore, the aim of this research was to create the AdTech software life cycle from scratch. By conducting interviews with AdTech experts, coding the interview transcriptions and grouping the codes, seven groups were identified. Of these seven groups, four groups could be identified as software life cycle stages. These software life cycle stages address the challenges of AdTech, which is discussed in detail in Chapter 6. With insights from the case study and interviews, we modelled the AdTech software life cycle. We can conclude that the AdTech software life cycle consists of four stages: Initial Development, Scaling, Evolution and Replacement.

Initial Development

In the initial development stage for AdTech software, a foundational version is either built from scratch or using existing infrastructure. This stage includes integration with publisher systems and making the software easy to roll out and install, regardless of technical expertise. The software is developed in components to reuse the components for other use cases and simpler future updates. The developers frequently update the software and improve the software iteratively until a stable version 1.0 is achieved.

Scaling

The Scaling stage starts when a functional version of the AdTech software is developed. The development focus shifts to ensuring the system can handle large-scale traffic and data demands, because one single client could have significant impact on the infrastructure. To prepare the infrastructure for fluctuating traffic, the cloud should be leveraged. By utilising cloud resources and the auto scaling function it often offers, is it possible to scale automatic with the demand.

Evolution

In the Evolution stage, the AdTech software undergoes continuous development to remain relevant in a fast changing market. The software gets frequent updates address shifting user needs, emerging industry trends and regulatory developments. The combination of these forces make the Evolution stage a never-ending process of development. To keep up, it requires significant investment of time and resources.

Replacement

In the Replacement stage, AdTech software is outdated because of changing regulations, force competition, or lack innovation. The Legacy systems are often deeply integrated in publishers operations which pose challenges because replacing them risks disrupting ongoing campaigns. Consequently, a carefully transition transition plan is necessary. Replacing AdTech could be done in two ways: incremental component replacements or gradual migration of clients.

7.1.2 RQ.2 Does the AdTech software life cycle differ from the standard software life cycle?

By comparing the standard software life cycles with the AdTech software life cycle, we can conclude that the AdTech software life cycle differs from the standard software life cycle, such as the software life cycles from Rajlich and Bennett [37] and IEEE [23], because of its unique characteristics. While the Initial Development stage follows a traditional structure, AdTech software must integrate within publishers' websites, to create a functional version. This integration makes AdTech more complex in the initial development. Additionally, we introduced a new stage: the Scaling stage. AdTech requires a dynamic and robust infrastructure capable of handling the immense demand, to become fully operational.

Furthermore, the AdTech software life cycle does not include traditional Servicing, Maintenance, or Phaseout stages, as found in Rajlich and Bennett [37] and IEEE [23]. These stages are not applicable for AdTech software, which undergoes continuous development driven by evolution factors. If these factors are not enough tackled, AdTech software quickly becomes outdated. Consequently, there is no place for a phase dedicated to small updates or extending monetisation of a legacy system. The final stage, called "Replacement", aligns with the traditional Closedown or Retirement stages but emphasises transitioning to a new AdTech solution without revenue loss for publishers and the AdTech company. As a result, the life cycle does not accommodate a phase focused merely on minor updates or extending the monetisation of outdated software.

7.1.3 RQ.3 What best practices are effective for the development and maintenance of AdTech, and how can they be mapped to the software life cycle?

Based on the outcomes from the case study and interviews, we identified seven best practices that are tailored for AdTech development. These best practices are evaluated and found to be valuable and effective for AdTech development. Based on the feedback we received during the in-depth interviews, we concluded that not all best practices are only applicable to AdTech. Some of the best practices could be applicable to other domains such as MarTech. This is discussed in detail in Chapter 5 and 6. The best practices are categorised in a stage of the AdTech software life cycle. The following best practices have been defined:

Initial Development

Develop AdTech in Components
AdTech software should be developed in modular components to increase flexibility for updating the system. While the initial development cost may be be higher, this approach significantly reduces maintenance and adaptation expenses in the long run. Experts noted that modularity simplifies debugging and scaling independently. Implementation of this best practice should be done once the application's structure is clearly defined.

Develop AdTech That is Simple to Implement
Simplifying AdTech implementation is crucial for ensuring simple onboarding and reducing publisher dependency on technical support. A difficult implementation can lead to misconfigurations. Experts pointed out that when a setup is simple to execute through automation or clear documentation it will also improve adoption rates. When applying the best practice developers should caution for oversimplification. Customisation should remain available for advanced users and default configurations should be simple. Investing in a user-friendly implementation process reduces long-term support costs and accelerates adoption.

Scaling

Scale in the Cloud
The best practice Scaling in the Cloud was created for the AdTech domain, but is also applicable to other domains. AdTech companies must handle massive real-time data demands. By leveraging cloud-based resources with automatic scaling functions, AdTech infrastructure can be allocated the right amount of resources on based on the demand. These automated capabilities ensure that infrastructure scales up during peak traffic periods and scales down during low activity which lowers cloud costs.

Evolution

Monitoring of AdTech Resources
AdTech resources need to be continuous monitored to ensure maintain systems stability and prevent unexpected downtimes. Some Experts highlighted that handling large amounts of data without any proper oversight can lead to critical issues. By monitoring AdTech resources issues such as storage limitations, memory shortages and other bottlenecks can be identified on time.

Prioritising of Development Tasks
The best practice Prioritising Development Tasks is another best practice which is not exclusively applicable to AdTech. Experts mentioned that while innovation is necessary to stay relevant, maintenance also still needs to be done. Based on this, AdTech development teams must make strategic decisions when prioritising the development tasks. By utilising prioritisation frameworks such as MoSCoW or impact-effort matrices teams can allocate the available resources.

Leverage CI/CD in AdTech
This best practices suggests to leverage CI/CD in AdTech. Utilising CI/CD pipelines is essential for ensuring stable ad serving. Experts highlighted the importance of automated testing and multi-stage deployments. By integrating CI/CD, new features will be thoroughly tested before reaching production. So, by integrating CI/CD pipelines AdTech companies can streamline their releases while at the same time minimise downtime. Implementing the CI/CD pipelines requires investment in infrastructure, but it will ultimately enhances software stability and accelerates development cycles.

Replacement

Replacing AdTech in Components
When AdTech needs to be replaced, this best practices suggest to do this by replacing the system component by component. Experts mentioned that replacing AdTech software can be complex task because it involves live ad campaigns. A phased replacement approach allows new components to be integrated gradually. By applying this approach the development team reduces risks and ensures service continuity. This method spread costs over time, but it still requires careful planning, dependency management and thorough testing to prevent any downtime.

7.1.4 Answer to the Research Question

To answer the main research question: "*How can the standard software life cycle model be adapted to fit AdTech, and which best practices help bridge the gap in existing guidelines throughout this cycle?*" - we draw on the findings from the sub-research questions.

To adapt standard software life cycle models such as those proposed by Rajlich and Bennett [37] and IEEE [23] the AdTech requires extensive modifications. By, building on 7.1.2, which identified the key differences we can answer our main question. In this section we briefly describe how these standard models can be adjusted to meet the requirements of AdTech and its development.

The Initial Development stage generally follows a similar structure as standard software. However AdTech software must be built with early integration into publishers' systems in mind. This installation is a must-have requirement for any functional version.

Moreover, a new stage the Scaling stage must be introduced. AdTech requires a dynamic and robust infrastructure capable of handling the immense demand, to become fully operational. This is not required for standard software.

Additionally, the AdTech software life cycle does not include Servicing, Maintenance, or Phaseout stages. These stages are not applicable for AdTech software, which undergoes continuous development driven by evolution factors. If these factors are not enough tackled, AdTech software quickly becomes outdated. This continuos development means that AdTech software is in a state of constant evolution. Because of this trait plays the Evolution stage a much more important role in AdTech's software life cycle.

The final Replacement stage aligns with the Closedown or Retirement stages from Rajlich and Bennett [37] and IEEE [23]. However, the Replacement stage involves transitioning to a new AdTech solution without revenue loss for publishers and the AdTech company.

We can conclude that by transforming the standard life cycle with our suggested structural differences, it becomes better suited for AdTech. While the standard model can be adapted to address AdTech's requirements, we propose treating the AdTech software life cycle as a distinct model in its own right. The full model is explained in detail in Chapter 4.

To guide practitioners in the AdTech development field, we have created seven tailored best practices for each stage of the AdTech software life cycle. These practices offer practitioners actionable insights during AdTech development. A detailed discussion of these practices can be found in Section 7.1.3. By providing these best practices, is the existing gap in the AdTech guidelines filled.

7.2 Contributions

This study has contributed by providing a software life cycle model and a best practice catalogue tailored to AdTech to the existing literature on AdTech. By providing these artifacts we have addressed the gap in the current literature. The best practice catalogue offers a systematic collection of practices that can guide AdTech development practitioners in refining their development processes throughout the AdTech software life cycle. By mapping these best practices into the life cycle model, this research delivers a new understanding of AdTech's evolution. Our contributions should serve as a framework for both researchers and practitioners in the AdTech field.

7.3 Future Work

To address the limitations outlined in Chapter 6, we discuss in this section several areas for further research and refinement.

First, in this research five interviews were conducted in the exploratory phase. However, expanding this number in a future research could lead to the identification of additional best practices. Next to this, a more extensive set of exploratory interviews could capture a wider variety of industry perspectives and experiences. This could also help with improving the AdTech software life cycle and help refine the best practices. Additionally, this research included only one interview with a MarTech expert. Future research should include more interviews with MarTech experts to validate whether the developed artifacts are also applicable to the MarTech domain.

Second, increasing the number of validation interviews could improve the validation of the artifacts from a quantitative perspective. A larger sample size could not only strengthen the reliability of the findings but also provide a broader range of perspectives. This could improve the assessment of the effectiveness and applicability of the proposed artifacts.

Next to this, while this document provides a overview of the software life cycle of AdTech, a dedicated publication in a journal with the attached best practices could add even more value to industry professionals.

Another area for further investigation is how acquisitions affect the AdTech software life cycle. This study briefly discussed how scenarios of AdTech acquisitions could affect the software life cycle. These cases raise questions about whether such integrations represent a continuation of the original software life cycle, a form of evolution, or a complete replacement. Currently, this study addresses these secnarios by including them as evolution factors. However, more research is needed to decide if a dedicated stage or sub-model is needed to more accurately represent the role of acquisitions in the AdTech software life cycle model.

This study found that AdTech's software life cycle does not align with the standard software life cycles such as the proposed by Rajlich and Bennett [37] or IEEE [23]. This finding suggests that other domains may also have software that does not fit standard software life cycle models. So, future researches could investigate whether other domains require tailored software life cycles as well.

In addition to this direction, future research could also explore whether other software domains share characteristics with AdTech. Identifying such domains would allow researchers to examine whether their software life align with AdTech's life cycle. Such kind of analysis could improve the understanding of software in these specialised domains.

Another possible research direction is to examine whether the software life cycle proposed by Rajlich and Bennett [37] remains applicable to today's software. This software life cycle model was introduced in the year 2000. It could be possible that shifts in software development practices and software requirements over time have influenced its relevance and applicability.

Bibliography

[1] Adswag B.V. Advanced advertising solutions, made easy., 2025. URL https://adswag.nl/. Accessed: January 29, 2025.

[2] Meshari Alwazae, Erik Perjons, and Paul Johannesson. Applying a template for best practice documentation. *Procedia Computer Science*, 72:252–260, 2015. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2015.12.138. URL https://www.sciencedirect.com/science/article/pii/S1877050915035991. The Third Information Systems International Conference 2015.

[3] Hala Assal and Sonia Chiasson. Security in the software development lifecycle. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 281–296, Baltimore, MD, August 2018. USENIX Association. ISBN 978-1-939133-10-6. URL https://www.usenix.org/conference/soups2018/presentation/assal.

[4] Muhammad Ahmad Bashir and Christo Wilson. Diffusion of user tracking data in the online advertising ecosystem. *Proceedings on Privacy Enhancing Technologies*, 2018(4):85–103, 2018. doi: 10.1515/popets-2018-0033.

[5] Muhammad Ahmad Bashir, Sajjad Arshad, William Robertson, and Christo Wilson. Tracing information flows between ad exchanges using retargeted ads. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 481–496, Austin, TX, August 2016. USENIX Association. ISBN 978-1-931971-32-4. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/bashir.

[6] Melanie Birks and Jane Mills. *Grounded Theory: A Practical Guide*. SAGE Publications, London, 2011. ISBN 978-1848609921.

[7] Lars Brehm and M Lynne Markus. The divided software life cycle of erp packages. In *Proceedings of 1st Global Information Technology Management (GITM) World Conference*, pages 43–46. Citeseer, 2000.

[8] Stuart Bretschneider, Jr. Marc-Aurele, Frederick J., and Jiannan Wu. "Best practices" research: A methodological guide for the perplexed. *Journal of Public Administration Research and Theory*, 15 (2):307–323, 12 2004. ISSN 1053-1858. doi: 10.1093/jopart/mui017. URL https://doi.org/10.1093/jopart/mui017.

[9] Scott Brinker and Frans Riemersma. Martech supergraphic, 2025. URL https://martechmap.com/int_supergraphic. Accessed: 2025-02-25.

[10] Jan vom Brocke, Alan Hevner, and Alexander Maedche. *Introduction to Design Science Research*, pages 1–13. SpringerNature, 09 2020. ISBN 978-3-030-46780-7. doi: 10.1007/978-3-030-46781-4_1.

[11] Badrish Chandramouli, Jonathan Goldstein, and Songyun Duan. Temporal analytics on big data for web advertising. In *2012 IEEE 28th International Conference on Data Engineering*, pages 90–101, 2012. doi: 10.1109/ICDE.2012.55.

[12] She-I Chang, David Yen, Shi Huang, and Pei-Qei Hung. An erp system life cycle-wide management and support framework for small- and medium-sized companies. *Communications of the Association for Information Systems*, 22:275–294, 01 2008. doi: 10.17705/1CAIS.02215.

[13] Fred Davis. User acceptance of information systems: the technology acceptance model (TAM). *Int.J.Man-Machine Studies*, page 14, 01 1987.

[14] Steve M. Easterbrook, Janice Singer, Margaret-Anne D. Storey, and Daniela E. Damian. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*, 2008. URL https://api.semanticscholar.org/CorpusID:1105134.

[15] Naomi Elliott and Anne Lazenbatt. How to recognise a 'quality' grounded theory research study. *Nursing Times*, March 2004. Accepted for publication March 2004.

[16] Jose M. Esteves and Joan A. Pastor. An erp life-cycle-based research agenda. *First International Workshop in Enterprise Management and Resource Planning: Methods, Tools and Architectures (EMRPS'99)*, 1999.

[17] Alexander Gilmanov. What is madtech and how it came to be, February 2023. URL https://tms-outsource.com/blog/posts/madtech/. Accessed: 2024-12-08.

[18] Barney G. Glaser. *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory.* Sociology Press, Mill Valley, CA, 1978. ISBN 978-1884156014.

[19] Barney G. Glaser and Anselm L. Strauss. *Discovery of Grounded Theory: Strategies for Qualitative Research.* Routledge, 1999. doi: https://doi.org/10.4324/9780203793206. URL https://www.taylorfrancis.com/books/mono/10.4324/9780203793206/discovery-grounded-theory-barney-glaser-anselm-strauss.

[20] Lynda Gratton and Sumantra Ghoshal. Beyond best practice. *MIT Sloan Management Review*, 46 (3):49–57, 2005. URL https://sloanreview.mit.edu/article/beyond-best-practice/.

[21] Grounded Theory Online. What is grounded theory? https://www.groundedtheoryonline.com/what-is-grounded-theory/, 2025.

[22] Nerma Gusic and Valerio Stallone. The digital advertising ecosystem – status quo, challenges and trends. In *Proceedings of the Conference on Digital Advertising*, pages 36–42. Zurich University of Applied Sciences, 04 2020. ISBN 978-989-8704-14-6. doi: 10.33965/es2020_202005L005. © 2020.

[23] IEEE. Ieee standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, pages 1–84, 1990. doi: 10.1109/IEEESTD.1990.101064.

[24] Ritu Jain and Ugrasen Suman. A systematic literature review on global software development life cycle. *ACM SIGSOFT Software Engineering Notes*, 40:1–14, 04 2015. doi: 10.1145/2735399.2735408.

[25] Marcus A. Rothenberger Ken Peffers, Tuure Tuunanen and Samir Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007. doi: 10.2753/MIS0742-1222240302. URL https://doi.org/10.2753/MIS0742-1222240302.

[26] Sujit Kumar and Pushkar Dubey. Software development life cycle (sdlc) analytical comparison and survey on traditional and agile methodology. *ABHINAV NATIONAL MONTHLY REFEREED JOURNAL OF RESEARCH IN SCIENCE & TECHNOLOGY*, 2:22–30, 08 2013.

[27] F. Lehner. Software life cycle management based on a phase distinction method. *Microprocessing and Microprogramming*, 32(1-5):603–608, 1991. doi: 10.1016/0165-6074(91)90409-m.

[28] Juanjuan Li, Xiaochun Ni, Yong Yuan, and Fei-Yue Wang. A hierarchical framework for ad inventory allocation in programmatic advertising markets. *Electronic Commerce Research and Applications*, 31:40–51, 2018. ISSN 1567-4223. doi: https://doi.org/10.1016/j.elerap.2018.09.001. URL https://www.sciencedirect.com/science/article/pii/S1567422318300711.

[29] M. Mahalakshmi and M. Sundararajan. Traditional sdlc vs scrum methodology - a comparative study. In *Traditional SDLC Vs Scrum Methodology - A Comparative Study*, 2013. URL https://api.semanticscholar.org/CorpusID:110054821.

[30] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*, pages 413–427, 2012. doi: 10.1109/SP.2012.47.

[31] Peter Meso and Radhika Jain. Agile software development: Adaptive systems principles and best practices. *IS Management*, 23:19–30, 06 2006. doi: 10.1201/1078.10580530/46108.23.3.20060601/93704.3. URL https://www.tandfonline.com/doi/abs/10.1201/1078.10580530/46108.23.3.20060601/93704.3.

[32] Renad Mokhtar and Mashael Khayyat. A comparative case study of waterfall and agile management. *SAR Journal - Science and Research*, 5(1):52–62, 2022. doi: 10.18421/SAR51-07.

[33] N. Offierowski. *Programmatic Advertising. Voraussetzungen und Herausforderungen für kleine Mittelständische Unternehmen.* Grin Verlag, München, 2017.

[34] M. Steven Palmquist, Mary Ann Lapham, Suzanne Miller, Timothy Chick, and Ipek Ozkaya. Parallel worlds: Agile and waterfall differences and similarities. Technical Note CMU/SEI-2013-TN-021, Software Engineering Institute, Carnegie Mellon University, 2013. URL http://www.sei.cmu.edu.

[35] Paulina Jo Pesch. Drivers and obstacles for the adoption of consent management solutions by ad-tech providers. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 269–277, 2021. doi: 10.1109/EuroSPW54576.2021.00034.

[36] Mary Peters and Timothy Heron. When the best is not good enough: An examination of best practice. *Journal of Special Education - J SPEC EDUC*, 26:371–385, 01 1993. doi: 10.1177/002246699302600403.

[37] V.T. Rajlich and K.H. Bennett. A staged model for the software life cycle. *Computer*, 33(7):66–71, 2000. doi: 10.1109/2.869374. URL https://ieeexplore.ieee.org/document/869374.

[38] T. Saravanan, Sumit Jha, Gautam Sabharwal, and Shubham Narayan. Comparative analysis of software life cycle models. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 906–909, 2020. doi: 10.1109/ICACCCN51052.2020.9362931.

[39] Clive Seale. *The Quality of Qualitative Research.* SAGE Publications Ltd, 1999. doi: 10.4135/9780857020093. URL https://doi.org/10.4135/9780857020093.

[40] Sarang Shaikh and Sindhu Abro. Comparison of traditional & agile software development methodology: A short survey. *International Journal of Software Engineering and Computer Systems*, 5:1–14, 08 2019. doi: 10.15282/ijsecs.5.2.2019.1.0057.

[41] Robert Thornberg, Lisa M. Perhamus, and Kathy Charmaz. *Handbook of Research Methods in Early Childhood Education: Research Methodologies*, volume I of *Contemporary Perspectives in Early Childhood Education.* Information Age Publishing, Charlotte, NC, 2014. ISBN 978-1-62396-610-2, 978-1-62396-611-9, 978-1-62396-612-6.

[42] Michael Veale and Frederik Zuiderveen Borgesius. Adtech and real-time bidding under european data protection law. *German Law Journal*, 23(2):226–256, 2022. doi: 10.1017/glj.2022.18.

[43] Yash Vekaria, Rishab Nithyanand, and Zubair Shafiq. Turning the tide on dark pools? towards multi-stakeholder vulnerability notifications in the ad-tech supply chain. *ArXiv*, page 16, 06 2024. doi: 10.48550/arXiv.2406.06958.

[44] Deepak Venkatachalam, Pradeep Manivannan, and Rajalakshmi Soundarapandiyan. Case study on the integration of customer data platforms with martech and adtech in pharmaceutical marketing for enhanced efficiency and compliance. *Journal of Artificial Intelligence Research and Applications*, 2(1):39, 2022. URL https://www.journalofartificialintelligence.com. Accessed: 2024-11-29.

[45] Benjamin Weder, Johanna Barzen, Frank Leymann, Marie Salm, and Daniel Vietz. The quantum software lifecycle. In *Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software*, APEQS 2020, page 2–9, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450381000. doi: 10.1145/3412451.3428497. URL https://doi.org/10.1145/3412451.3428497.

[46] Gary Whited and Awad Hanna. Best practices from wisdot mega and arra projects – best practice catalog. Technical report, Wisconsin Department of Transportation, March 2012. Final Report, WisDOT Policy Research Program Project, Project ID: 0092-10-20.

[47] Bella Williams. Example of field notes for observational research, January 2025. Insight company. Accessed: May 1, 2025.

[48] M. Worledge, S. Head, R. Manager, and M. Bamford. Adtech market research report. https://www.harrisinteractive.co.uk, 2019. Accessed: 2024-11-29.

[49] Jiannan Wu, Yao Liu, and Stuart Bretschneider. Best practice is not just "best": An empirical study based on judges' perceptions. *Urban Governance*, 3(2):130–137, 2023. ISSN 2664-3286. doi: https://doi.org/10.1016/j.ugj.2023.03.005. URL https://www.sciencedirect.com/science/article/pii/S2664328623000347.

[50] Robert K. Yin. *Case Study Research: Design and Methods*. SAGE Publications, Thousand Oaks, CA, 3th edition, 2003.

[51] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, ADKDD '13, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450323239. doi: 10.1145/2501040.2501980. URL https://doi.org/10.1145/2501040.2501980.

[52] Christos Ziakis and Maro Vlachopoulou. Artificial intelligence in digital marketing: Insights from a comprehensive review. *Information*, 14:664, 12 2023. doi: 10.3390/info14120664. URL https://api.semanticscholar.org/CorpusID:267061851.

Appendix A

Exploratory Interview Guide

Interview no.	
Interviewee name	
Interviewee role	
Company	
Date	

Table A.1: General interview information.

A.1 Introduction

Thank you for taking the time to participate in this interview. This interview is part of my Master's research project. In this research, we focus on understanding the software life cycle of AdTech. The objective of the research is to define the software life cycle specific to AdTech. Your insights will help shape the outcomes of this research. In this study AdTech refers to "Advertising Technology" and to all technologies, software and services used to deliver and target online advertisements. As said in the interview invitation, the interview will be recorded. The recording will be exclusively for analytical use. All data will be anonymised and managed with confidentiality. Before we begin, do you have any questions?

A.2 General Questions

1. What is your role within your organisation?
2. What is your experience with software development?
3. What type of AdTech does your organisation develop? (e.g., DSP, SSP)
4. Which components of your software need to be scalable from the start? Why?

A.3 Initial Development

(Engineers develop the system's first functioning version. [37])

1. What are the most critical technical considerations when launching new AdTech software?
2. Who are your end-users?
3. What requirements do your end-users have?
4. Which development methodologies do you specifically use for AdTech development? (e.g., Scrum, Agile, Waterfall, etc.)

- Do you continue to use these methodologies as the software matures?

5. What challenges do you face when developing AdTech?
6. Are there challenges in AdTech development that you do not encounter in standard software development? (e.g., collaboration or implementation issues)

A.4 Evolution

(Engineers extend the capabilities and functionality of the system to meet user needs, possibly in major ways [37])

1. What are the biggest challenges in maintaining AdTech software compared to starting it?
2. Are software requirements influenced by market changes or customer needs?
3. Do you use testing methods before deploying the software?
4. Does your software reach many users during testing?
5. Does your software reach many users when it is first deployed?
6. If yes, do you take precautions to ensure that the rollout of a new feature proceed smoothly?
7. How do you account for regulations during your software development?
8. How do you handle frequent API changes from dependent parties?
9. What strategies do you use to minimise disruptions during software replacement or updates?

A.5 Servicing

(Engineers make minor defect repairs and simple functional changes. [37])

1. How would you estimate the maintenance costs of AdTech compared to standard software? (e.g., an ERP system)
2. How much maintenance is required for AdTech once the all the features are implemented?

A.6 Phaseout

(he company decides not to undertake any more servicing, seeking to generate revenue from the system as long as possible. [37])

1. How quickly does AdTech become outdated? What factors drive this?

A.7 Closedown

(he company withdraws the system from the market and directs users to a replace- ment system, if one exists. [37])

1. Can AdTech be easily replaced? Why?
2. What happens to the software when it is no longer used?
3. Is it more difficult to replace AdTech systems than standard software? (e.g., an ERP system)

Thank you for participating in this interview. Your insights are invaluable to our research and understanding of AdTech development. If you are willing, I may have follow-up questions to clarify certain points. Would you be open to answering those in the future?

Thank you again for your valuable time and expertise.

Appendix B

Exploratory MarTech Interview Guide

Interview no.	
Interviewee name	
Interviewee role	
Company	
Date	

Table B.1: General interview information.

B.1 Introduction

Thank you for taking the time to participate in this interview. This interview is part of my Master's research project. In this research, we focus on understanding the software life cycle of AdTech. The objective of the research is to define the software life cycle specific to AdTech. Your insights will help shape the outcomes of this research. In this study AdTech refers to "Advertising Technology" and to all technologies, software and services used to deliver and target online advertisements. As said in the interview invitation, the interview will be recorded. The recording will be exclusively for analytical use. All data will be anonymised and managed with confidentiality. Before we begin, do you have any questions?

B.2 General Questions

1. What is your role within your organisation?
2. What is your experience with software development?
3. What type of MarTech do you have experience with?

B.3 Questions about MarTech in General

1. Which technologies are included in MarTech?
2. What characteristics do all MarTech technologies have?
3. Is AdTech a part of MarTech, vice versa, or are AdTech and MarTech two distinct domains?
4. What challenges do MarTech technologies face in the industry?

B.4 Questions about the Technical Aspects of MarTech

1. Who are MarTech's end-users?
2. What requirements do MarTech end-users have?
3. Are software requirements influenced by market changes or customer needs? If so, which factors drive these changes?
4. What challenges do you face when developing MarTech solutions?
5. Does MarTech continually evolve to stay relevant?
6. How quickly does MarTech become outdated? What factors drive this?
7. Can MarTech solutions be easily replaced? Why or why not?
8. What happens to MarTech software when it is no longer used?
9. Is it more difficult to replace MarTech systems than standard software (e.g., ERP systems)? Why?

B.5 Comparing AdTech and MarTech

1. What are the key differences between AdTech and MarTech in terms of technology and functionality?
2. Do AdTech and MarTech share a common software architectures?
3. Are AdTech and MarTech solutions interchangeable in any way?
4. How does data usage differ between AdTech and MarTech?
5. Are the regulatory challenges the same for both AdTech and MarTech?
6. Does AdTech evolve at a different pace compared to MarTech? If so, why?

Thank you for participating in this interview. Your insights are invaluable to our research and understanding of AdTech development. If you are willing, I may have follow-up questions to clarify certain points. Would you be open to answering those in the future?

Thank you again for your valuable time and expertise.

Appendix C

Questionnaire

Q1. On a scale of 1-to-5, how easy is the software life cycle to understand? (1 being very difficult and 5 being very easy.)

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q2. Could you motivate your answer to the previous question?

Q3. On a scale of 1-to-5, how complete do you think the AdTech software life cycle model is? (1 being very incomplete and 5 being very complete.)

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q4. Could you motivate your answer to the previous question?

Q5.On a scale of 1-to-5, how easy do you think is it is to use the software life cycle for insights in AdTech development? (1 being very difficult and 5 being very easy.) very easy

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q6. Could you motivate your answer to the previous question?

Q7. On a scale of 1-to-5, how correct do you think the software life cycle is? (1 being very incorrect and 5 being very correct.)

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q8. Could you motivate your answer to the previous question?

Q9. How useful do you think the AdTech software life cycle model is for understanding AdTech software and its stages? (1 being very useless and 5 being very useful).

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q10. Could you motivate your answer to the previous question?

Q11, On a scale of 1 to 5, how likely are you to reference the AdTech software life cycle to explain the stages of AdTech software? (1 being very likely and 5 being very unlikely).

- ❍ 1
- ❍ 2
- ❍ 3
- ❍ 4
- ❍ 5

Q12. Could you motivate your answer to the previous question?

Appendix D

Validation Interview Guide

Interview no.	
Interviewee name	
Interviewee role	
Company	
Date	

Table D.1: General interview information.

D.1 Introduction

Thank you for taking the time to participate in this interview. This interview is part of my Master's research project. In this research, we focus on understanding the software life cycle of AdTech. The objective of the research is to define the software life cycle specific to AdTech. Your insights will help shape the outcomes of this research. In this study AdTech refers to "Advertising Technology" and to all technologies, software and services used to deliver and target online advertisements. As said in the interview invitation, the interview will be recorded. The recording will be exclusively for analytical use. All data will be anonymised and managed with confidentiality. Before we begin, do you have any questions?

D.2 General Questions

1. What is your role within your organisation?
2. What is your experience with software development?
3. What type of AdTech have you build? (e.g., DSP, SSP)

D.3 Best Practice Questions

1. Do you understand the best practice?
2. Do you think the best practice is correct?
3. Do you think the best practice is a good best practice for AdTech development?
4. How useful is the best practice for improving efficiency and effectiveness?
5. How likely do you think AdTech development teams are to adopt the best practice?

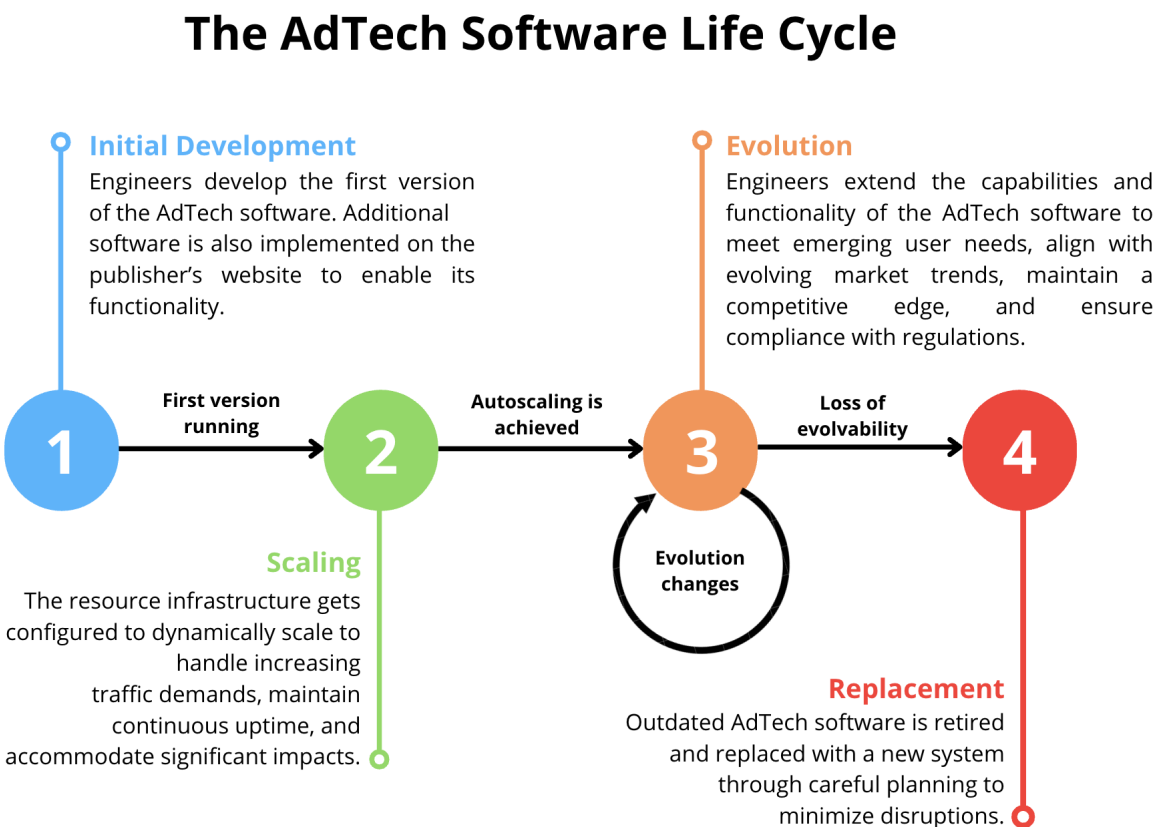
Thank you for participating in this interview. Your insights are invaluable to our research and understanding of AdTech development. If you are willing, I may have follow-up questions to clarify certain points. Would you be open to answering those in the future?

Thank you again for your valuable time and expertise.

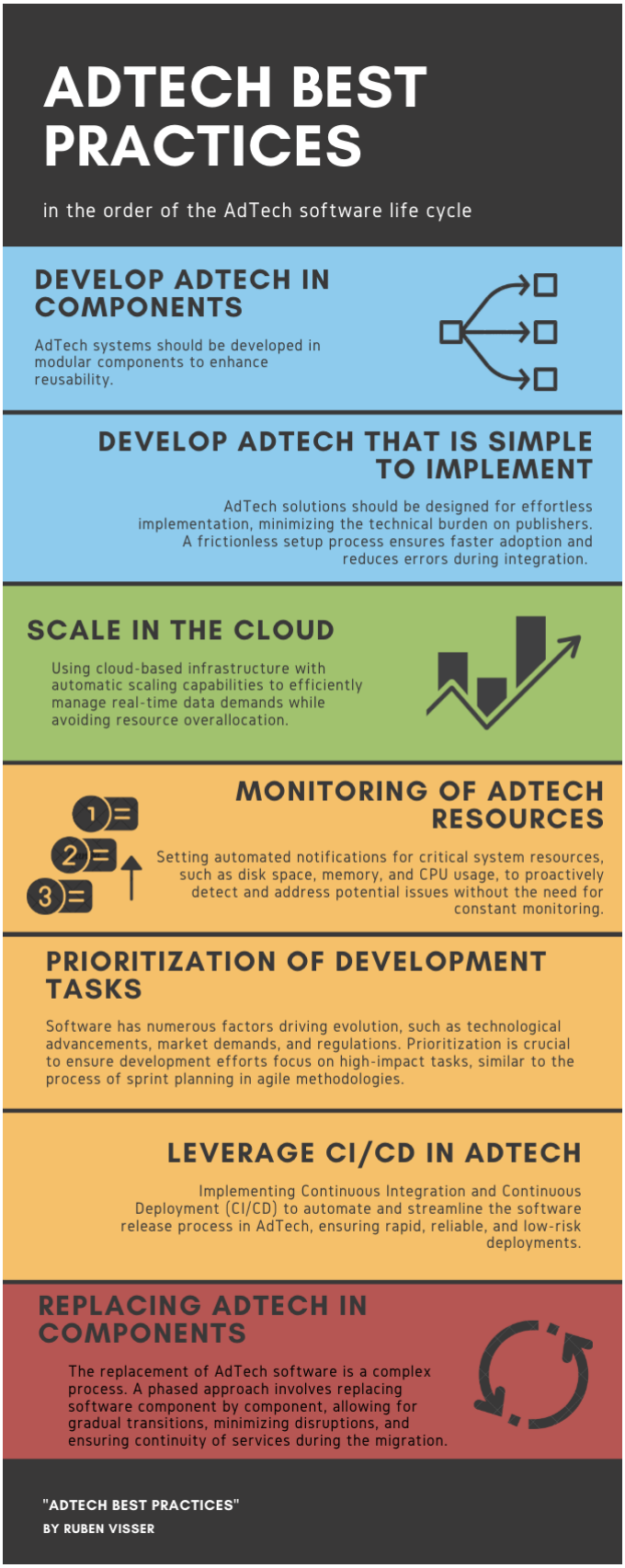
Appendix E

Infographics

E.1 Infographic of AdTech Software Life Cycle Model



E.2 Infographic of AdTech Best Practices



Appendix F

Log Book

This appendix presents the complete log book from the exploratory phase of the branded content case study, conducted over two months (January–February). It documents key activities, observations and insights related to the development of Adswag's branded content software. The Date column is formatted as DD-MM-YYYY.

Table F.1: Overview of logged entries from the branded content case study.

Date	**What**	**Who**	**How**	**Why**
03/01/2025	Problems with configuring publishers. Branded content gets displayed on multiple places on the publishers website.	Publishers, Head of Product	Publishers implemented code to display branded content.	Configuration of the code went wrong on the publishers side.
06/01/2025	There is a need for a preview version of the branded content software to demo the software to client.	Sales Director	Develop dummy branded content software with example data on a separate domain.	To sell the software, demo software is needed.
08/01/2025	Server stores cache too long, because of this the latest advertisement changes are not displayed.	Software Developer	Server doesn't empty the cache.	The server's provider is blocking the cache emptying on a command basis.
08/01/2025	Data storage of advertisement logs in the cloud is too expensive.	Data Engineer	Switch the databases to a different region in the cloud.	By switching the databases to a different region, we can expect to reduce the cost of data transfers.

20/01/2025	The maintenance of data types in software development is a hurdle.	Software Developer	Researching data types and reusing them better.	It is now time-consuming to maintain all the different types.
20/01/2025	UI Designs from consultancy are taking long to finish.	Software Developer	Waiting on the designs slows down progress.	The functionalities are complete, but the next step is to code the designs.
20/01/2025	Native ad configuration is a time consuming manual task.	Head of Product	Create a prompt to automatically configure publisher domains.	To onboard publishers' domains for native ads, the configurations need to be tailored to each domain.
27/01/2025	The development team have a hard time with choosing the right software version merge with main and make public.	Software Developer, Full Stack Developer, Head of Product	Merge one version with the main branch.	The sales team need a working version to demonstrate.
29/01/2025	Versions of the software need to be defined to celebrate success.	Sales Director	Defining the scope and requirements of each software version.	If software versions are not clearly defined, it becomes difficult to prepare users for updates.
29/01/2025	The framework where the AdTech application is build on needs to be replaced by a cheaper framework.	Software Developer	Replacing NextJS with ReactJS will lower the costs to host the app.	The traffic generated by the AdTech platform has increased so much that the current hosting provider has become too expensive.
03/02/2025	Unexpected error occurred while campaign was running.	Software Developer	One old cluster had an error.	Cubejs was configured to update, which was causing an executing error.
05/02/2025	Unexpected error occurred in the cloud functions for data collection.	Data Engineer	The cloud function could not find a python library.	An update in Google cloud functions caused the error.
05/02/2025	Cloud function needs to be fixed to run a campaign on the same day.	Data Engineer	Updating the cloud function and test it.	There is a campaign planned on the same day.
11/02/2025	Infinite scroll pages cause errors when serving branded content articles.	Publishers	Publishers should only display branded content articles on regular, non-infinity scroll pages.	Branded content articles are not meant to be displayed on infinite scroll pages.

12/02/2025	New publishers network need to be onboarded for the branded content.	Head of Product	Plan a call with the stakeholders to make a plan for their onboarding.	By onboarding new publishers, agencies gain greater flexibility in targeting their audience based on various domains.
12/02/2025	A decision must be made regarding whether the advertising performance dashboard should display real-time data or near-real-time data (every 5 minutes).	Data Analyst, Head of Tech, Data Engineer	By implementing data pipelines using Kafka, it is possible to obtain real-time data, but this comes at a higher cost.	Real-time data improves customer satisfaction.
17/02/2025	Decide which bucket storage provider will replace the old storage system.	Software Developer	By comparing multiple providers, you can compare on factors such as speed, costs, dependencies, and integrating effort. This will help you make a wise decision.	The current storage system is inadequate because the benefits it once provided are no longer relevant in the new advertising infrastructure.
18/02/2025	The Junior Full Stack Developer needs to meet with the lead developer to understand how the new framework work.	Full Stack Developer	By explaining the new framework to the Junior Developer, he can migrate his code into the new framework.	If the entire team comprehends the new framework, the development of the AdTech app accelerates.
25/02/2025	The AdTech system exceeds the edge request limits in the cloud.	Head of Tech, Software Developer	Monitor the cloud resources to see if the limits are exceeded by an expected reason.	Exceeding the limits of cloud resources can lead to increased costs.

Appendix G

Code Book

This appendix provides the code book which contains all the codes used in this study. The codes have been used for data analysis. Each code group is organised into multiple levels of detail to provide greater clarity and depth. These levels are denoted by an underscore ('_') following each code, indicating that subsequent codes are nested subsets of the preceding ones within the group. In total, 7 code groups (or themes) were identified and defined.

- **AdTech:** Contains codes that refers to AdTech.
- **Data Collection (DC):** Contains codes related to the data collection practices of AdTech companies.
- **Evolution of AdTech(EA):** Contains codes related to the Evolution of AdTech.
- **Initial Development (ID):** Contains codes related to the initial development of AdTech..
- **Replacement of AdTech (RA):** Contains codes related to the replacement AdTech.
- **Regulations:** Contains codes related to regulations.
- **Scale AdTech (SA):** Contains code related to the scaling of AdTech.

Table G.1: Codes extracted from interviews with their definition and source.

Code	Frequent	Definition	Expert	Example
#AdTech_accurate_measuring	1	Describes the importance of accurately measuring the viewer of an advertisement.	D	"You have to make sure you don't double count requests, because that can become costly if you do it wrong. Those are very expensive mistakes." - Interviewee Expert D
#AdTech_acquisition	1	Describes the acquisition of AdTech software.	C	"So what comapany x does is acquire startups or companies that are essential in the chain, ensuring that revenue is immediately offset against those costs" - Interviewee Expert C
#AdTech_adserver_as_infrastructure	1	Describes the use of an adserver as an infrastructure.	B	"it's an adserver we mostly use their software to share ad across the web." - Interviewee Expert B

Code	Frequent	Definition	Expert	Example
#AdTech_big_orders_in_AdTech	1	Describes the customer orders in AdTech	D	"...In AdTech, you can be talking tens of thousands of euros a day, scaling higher as you grow." - Interviewee Expert D
#AdTech_branded_content_software	2	Describes branded content software.	B	"yeah, it's a software whichs displays branded content on a publishers site. Yeah. so our clients can buy an article on a different websites" - Interviewee Expert B
#AdTech_buy_DSP	1	Describes the demand side platform.	A	"What is very similar is that we have a sell side, the SSP side" - Interviewee Expert A
#AdTech_buyers_and_sellers_have_the_same_amount_of_requirements	1	Describes that buyers and sellers have the same amount requirements.	A	"Generally, the buying and selling parties the same amount of requirements" - Interviewee Expert A
#AdTech_both_DSP_SSP	2	Describes AdTech which provide both DSP and SSP.	A	"Our platform connects sellers (websites/publishers) and buyers (brands), in that sense it encompasses both sides" - Interviewee Expert A
#AdTech_business_relations	1	Describes the importance of business relations.	B	"I mean business relations are really important. Google DV 360 platform. So let's say you have ad server and someone would like to be your competitor and he will create ad server and this ad server will be even better because it will be quicker. It will have more features deliver campaigns in a better way." - Interviewee Expert B
#AdTech_closer_to_end_users_focus_on_regulations	1	Describes the responsible parties in the AdTech value chain.	B	"I think that it's a mostly maybe for instance investor or someone who is closer to the real end users really keep track on what are regular regulations are and what we can't do. I mean for instance we are aware that we can't collect sensitive data to identify user directly. we can't share this data with some other companies." - Interviewee Expert B

Code	Frequent	Definition	Expert	Example
#AdTech_delivers_what_it_promises	1	Describes the importance of promises in AdTech.	A	”On the advertiser side, it needs to be reliable, the product promise is to simplify the work. It’s easy, what they book also happens” - Interviewee Expert A
#AdTech_GDPR_changes_affects_sales,	1	Describes how GDPR effects sales.	A	”Legislation, GDPR can make certain things prohibited. It impacts what can be sold” - Interviewee Expert A
#AdTech_high_entree_barrier	2	Describes the high entree barrier of AdTech.	C	”Not easily, because there is a high barrier to entry” - Interviewee Expert C
#AdTech_higher_ROI_than_competitors	1	Describes the importance of offering a higher return on investment than your competitors.	D	“They want the platform to function properly and, ideally, earn more money here than with competing solutions such as Google.” - Interviewee Expert D
#AdTech_interwoven_ecosystem	1	Describes the ecosystem of AdTech.	C	”The ecosystem is intertwined through contracts with advertisers, media agencies, sales houses, and publishers” - Interviewee Expert C
#AdTech_ISSO_must_for_collaboration_with_big_parties	1	Describes the importance of the ISSO for collaboration.	A	”Additionally, we comply with certain ISSO measures. To collaborate with large parties, you must meet ISSO standards. You can’t avoid it if you’re large. Everything must be correct, so it’s auditable” - Interviewee Expert A
#AdTech_market_changes_rapidly	1	Describes the pace in the AdTech market.	A	”If you design it smartly, you can indeed reuse it. If it’s one big monolith and a use case is something different, you have a problem. The market changes very quickly” - Interviewee Expert A
#AdTech_needs_to_be_easily_understand	1	Describes the importance of an easy understandable solution.	B	”As I understand the assumption for our system is that it has to be easy to start to configure campaign.” - Interviewee Expert B
#AdTech_needs_to_be_simple	1	Describes the importance of an simple solution.	A	”For the most part, yes. On the advertiser side, it needs to be reliable, the product promise is to simplify the work.” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#AdTech_paymentprocess_needs_to_work_and_transparant	1	Describes the importance of a transparant paymentprocess.	A	”This doesn’t always happen in the advertising world. Additionally, there is a billing model. The payment process must be transparent and accurate, so watertight” - Interviewee Expert A
#AdTech_rapid_world	3	Describes the fast paced industry.	C	”Maintaining is one thing, but today’s AdTech is already outdated” - Interviewee Expert C
#AdTech_ad_serving_exceptions	1	Describes that not always the ad serving works because of exceptional devices.	D	“We installed Sentry to monitor errors, and we’re at about 0.001% of requests failing—often because someone’s using an old Nokia phone that doesn’t load everything. It’s just not a huge concern for us because the API is robust.”- Interviewee Expert D
#AdTech_large_stakeholders	1	Describes that AdTech works with large stakeholders.	D	“. . . there are larger stakeholders behind them. The consequences of errors are bigger.” - Interviewee Expert D
#AdTech_sell_SSP	1	Describes the supply side platform.	A	”What is very similar is that we have a sell side, the SSP side and a buy side, the DSP side” - Interviewee Expert A
#AdTech_software_can_be_used_in_different_upcoming_domains	1	Describes AdTech software’s adaptability to emerging domains.	A	”The market operates in cycles, for example: 20 years ago were the first ads, it’s becoming more automated, then other sectors like retail (e.g., AH) emerge. These sectors need to be rebuilt from scratch. So, AdTech can be reused Therefore, it’s wise to build in components” - Interviewee Expert A
#AdTech_using_api	1	Describes the importance of API usage in AdTech.	B	”Usually the process looked like in the app user can create the creative and later through this platform’s API the ad is shared it’s published to the platforms and later served” - Interviewee Expert B

Code	Frequent	Definition	Expert	Example
#AdTech_lack_of_SEO	1	Describes that ads struggle with SEO (Search Engine Optimisation)	D	"One possible issue is how we load articles by adding a parameter in the URL. That means there's no SEO for that page, and we have to replace content to actually show the ad or article." - Interviewee Expert D
#AdTech_leverage_third_party_software	1	Describes the importance of third party software.	C	"In the first two or three years, allow it on the network, learn from it, and then decide if the technology is essential for operations. If so, then decide whether to develop it in-house or acquire that partner " - Interviewee Expert C
#AdTech_is_SaaS	1	Describes that AdTechreally is a SaaS (Software as a Service).	A	"Think for any software, in principle, we are just a SaaS solution, so it doesn't need to be different" - Interviewee Expert A
#AdTech_UI_less_critical	1	Describes UI less critical in the context of AdTech downtime.	A	"Maybe also in ERP. If you're live, you're live and can't be offline for an hour for maintenance #AdTech_cant_have_downtime. Maybe some ERP systems too. UI can be offline for an hour" - Interviewee Expert A
#AdTech_technical_standpoint_maintainance_is_harder_than_starting	1	Describes technical standpoint maintainance is harder than starting in the context of AdTech.	B	"let's assume that the maintaining part is a challenge maybe even more that starting from a technical point of view" - Interviewee Expert B
#AdTech_write_own_code_for_overview	1	Describes the importance of writing code by the AdTech company itself.	D	"We keep things as simple as possible, writing as much code ourselves as we can. That way we know exactly what's going on—no weird magic that might break when traffic spikes. Sticking to the basics was an important choice we made." - Interviewee Expert D

Code	Frequent	Definition	Expert	Example
#DC_AdTech_dont_collect_sensitive_data	2	Describes that their AdTech doesn't collect sensitive data.	B	"I think that it's a mostly maybe for instance investor or someone who is closer to the real end users really keep track on what are regular regulations are and what we can't do. I mean for instance we are aware that we can't collect sensitive data to identify user directly. we can't share this data with some other companies." - Interviewee Expert B
#DC_AdTech_store_only_data_to_deliver_business_value	1	Describes AdTech store only data to deliver business value.	B	" So during designing system we are trying to collect only needed informations to be able to deliver business value without collecting any user specific details." - Interviewee Expert B
#DC_ask_permission_to_use_data	1	Describes AdTech has to ask permission to use data.	A	"I think the advertising world is responsible for privacy violations. So yes, it's a major part of the software. We listen to what is allowed and what isn't. Users on a site give their preferences" - Interviewee Expert A
#DC_data_collection	1	Describes data collection in AdTech.	B	"another thing is that if you collect a lot of data and you handle high traffic someone has to maintain it because for instance you can quickly exceed the amount of space on a disk. you can have not enough memory to handle this traffic. many things can go wrong." - Interviewee Expert B
#DC_GDPR_affect_data_collection_which_and_how	1	Describes how gdpr affect data collection which and how.	A	"We take GDPR into account, so how we store data and which data we store" - Interviewee Expert A
#EA_AdTech_needs_maintenance	1	Describes the importance of maintenance in AdTech.	A	How quickly does the advertising system become outdated without maintenance? "Very quickly. " - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#EA_API_are_effected_by_regulations	1	Describes that API are effected by regulations.	B	”so if you serve ad through these platforms maybe it’s not strictly on your side to follow these rules but indirectly it also affect your software because they introduce some changes because they need follow the rules the restrictions And in the end they have to make some changes in API to cover these requirements and so it’s also affect your software.” - Interviewee Expert B
#EA_API_update_method	1	Describes how and when an API should be updated with AdTech	D	“We might only do a big API update once a year.” - Interviewee Expert D
#EA_check_ad_serving_at_every_deployment	1	Describes the importance of the checking of ad serving at every deployment.	B	”for BCSD DK we have these tests after each deployment if the adserving still works.” - Interviewee Expert B
#EA_CI_CD	1	Describes the use of CI/CD with releasing software updates	D	“Which strategies do you use to minimise disruptions during software updates or replacements? We do have CI/CD. . . . If staging looks good, we merge into production and Vercel updates the production build. We have a similar process for the SDK. . . ” - Interviewee Expert D
#EA_factor_adbocker	1	Describes the evolution factor adblocker	D	“One major factor is ad blockers. If they put us on a blacklist, then every Chrome extension that blocks ads will start blocking our requests.” - Interviewee Expert D
#EA_factor_competitors	1	Describes the evolution factor competitors.	C	”If you don’t innovate for a year, it can happen quickly. Due to competition, innovation from the rest of the ecosystem” - Interviewee Expert C

Code	Frequent	Definition	Expert	Example
#EA_factor_customer_needs	2	Describes the evolution factor of customer needs.	D	"Yes, for example, initially nobody asked for it, but now advertisers want click-through links in the article. Like if Brand X wants to sell its beer, they add a link, a user clicks it, we record an event, and Brand X can track it using UTM parameters." - Interviewee Expert D
#EA_factor_customer_needs_not_addressed	1	Describes what happends when the customer needs are not addressed.	D	"What if you don't address these needs? Then customers might be less satisfied. Would they leave? Possibly not over just that, but it's definitely a downside." - Interviewee Expert D
#EA_factor_API_changes_frequently	1	Describes the evolution factor api changes frequently.	B	"Adtech is that usually you integrate with many third-party companies these platforms that you mentioned before. So this API evolve they introduce changes quite frequent in this. So you have to maintain it because they always give some migration period but in the end they can just disable some functionality that you used before. so you have to keep track on it." - Interviewee Expert B
#EA_factor_different_client_profiles	1	Describes the evolution factor different client profiles.	C	"From their own usage, initially, we were focused on one client, but eventually, as we implemented it with publishers, we realized they could also be clients, and the client definition became much broader" - Interviewee Expert C
#EA_factor_fast_outdated	2	Describes the evolution factor fast outdated software.	C	"More than you think. It's never finished. I think maintenance is always more, a lot. Maintaining is one thing, but today's AdTech is already outdated" - Interviewee Expert C

Code	Frequent	Definition	Expert	Example
#EA_factor_high_traffic_is_maintainency	2	Describes the evolution factor high traffic creates maintainency.	B	”another thing is that if you collect a lot of data and you handle high traffic someone has to maintain it because for instance you can quickly exceed the amount of space on a disk. you can have not enough memory to handle this traffic. many things can go wrong.” - Interviewee Expert B
#EA_factor_maintance	1	Describes the evolution factor maintance.	C	”Because there are new technologies, you also have to connect to them, so maintenance is part of development.” - Interviewee Expert C
#EA_factor_unexpected_errors	1	Describes the evolution factor unexpected errors.	B	”another thing is that if you collect a lot of data and you handle high traffic someone has to maintain it because for instance you can quickly exceed the amount of space on a disk. you can have not enough memory to handle this traffic. many things can go wrong.” - Interviewee Expert B
#EA_factor_new_technologies	1	Describes the evolution factor new technologies.	C	”Because there are new technologies, you also have to connect to them” - Interviewee Expert C
#EA_factor_platform_innovation	1	Describes the evolution factor of the innovation of the AdTechs platform	D	“That’s where the innovation is—simplifying the branded content workflow. The product itself is already innovative in that it cuts out an entire flow by bundling everything into one platform.” - Interviewee Expert D
#EA_factor_regulations	2	Describes the evolution factor regulaitons.	A	”If regulations change, technical solutions also change accordingly” - Interviewee Expert A
#EA_factor_trends	1	Describes the evolution factor trends.	A	”People follow trends in this market. Everyone wants the latest technology” - Interviewee Expert A
#EA_factor_technical_changes_affect_ad_market	1	Describes the evolution factor technical changes affect ad market.	A	”Technical changes, for example, browsers changing cookies” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#EA_factor_younger_market	1	Describes the evolution factor younger market.	D	"The market is getting younger, meaning younger people use our platform. We have Koen and Kiki, for example; Kiki is quite young. We want the platform to be user-friendly for that demographic. Hence the UI/UX designer. So the design of our app might keep evolving, but on the backend side, not much changes based on the market or client needs." - Interviewee Expert D
#EA_in_year_outdated	1	Describes AdTech is in a year outdated.	C	"Very quickly, you can be outdated in a year" - Interviewee Expert C
#EA_innovation_nessicity	1	Describes innovation nessicity in AdTech.	C	"I think. If you don't innovate/develop, it dies" - Interviewee Expert C
#EA_never_finished	1	Describes that AdTech is never finished.	C	"It must continue to be developed; it's never finished" - Interviewee Expert C
#EA_no_innovation_means_leaving_clients	1	Describes no innovation means leaving clients.	A	"Speaking about regular AdTech. The market continuously wants new things; some even have different preferences. Clients will quickly switch to competitors if you don't offer the latest features that your competitors do. As long as there are no competitors, you can hold out for a long time. With standard AdTech, you're out if you don't innovate." - Interviewee Expert A
#EA_ongoing_API_changes_demand_frequent_updates	1	Describes ongoing api changes demand frequent updates.	B	"I think that there many IPIs can change quickly where you have to react on a traffic on a load on your application." - Interviewee Expert B
#AdTech_maintenance_with_care.	1	Describes the importance of carefully maintaining AdTech	D	"Unless it's a significant change that could cost us thousands of euros a day if we don't implement it, it's often best to leave it alone. Especially the logic for articles and placeholders." - Interviewee Expert D

Code	Frequent	Definition	Expert	Example
#EA_maintenance_for_regulations	1	Describes maintenance for regulations.	A	”With regulations, a lot sometimes needs to change, and the entire AdTech system must quickly adapt. The question is whether this is maintenance or development.” - Interviewee Expert A
#EA_test_method	1	Describes the use of lists to test on specific domains	D	“Do you test with a small portion of the audience first, or does everyone get updates at once? Because of how our API is built, all publishers are immediately able to access new functionality. . . . We might add domain-based logic later on (like domain whitelists or blacklists).” - Interviewee Expert D
#EA_third_party_software_for_monitoring	1	Describes the use of third party software for monitoring of AdTech	D	“We installed Sentry to monitor errors, and we’re at about 0.001% of requests failing—often because someone’s using an old Nokia phone that doesn’t load everything. It’s just not a huge concern for us because the API is robust.”- Interviewee Expert D
#EA_software_update_rules	1	Describes software update rules.	C	”Yes, rule number one is that we don’t deploy anything live on Fridays. That’s the basic rule. Essentially, it is tested in beta; then you have to deploy the event live, and things can still go wrong” - Interviewee Expert C
#EA_standarisation_importance	1	Describes standarisation importance.	C	”Initially, we were very broad, wanting everything. But then we started focusing on what we actually wanted and what it was intended for, and how we could possibly standardize” - Interviewee Expert C
#EA_without_maintance_fast_outdated	1	Describes evolution factor without maintance fast outdated.	B	”without maintaining I think that this software will not work for a long period. . . ” - Interviewee Expert B
#ID_accessibility	1	Describes accessibility.	A	”We also consider accessibility, such as colors, so people with color vision deficiencies can read” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#ID_API_versions	1	Describes what happends when third party APIs change	D	“If that API changes often, I assume they’ll use versioning. If we’re on version 1 and they release version 2, we’ll keep using version 1 until we see a reason to switch. We have direct contact with them, so we can coordinate and avoid disruptions.” - Interviewee Expert D
#ID_components_help_reducing_complexity	1	Describes components help reducing complexity.	A	”By thinking in components, you try to prevent this, to allow reuse” - Interviewee Expert A
#ID_create_software_in_components_for_reusability	1	Describes the importance to create software in components for reusability.	A	”Media agencies sometimes want exceptions, only to build a component that can be reused”
#ID_cost_strategy	1	Describes cost strategy in AdTech industry.	C	”Therefore, we chose to collaborate with Adswag so they would develop it and bear the costs ” - Interviewee Expert C
#ID_development_method_depends_on_team_size	2	Describes development method depends on team size.	A	”Yes, I think so. Other AdTech companies still do it this way. As the team grows, you will need to define tasks more specifically” - Interviewee Expert A
#ID_design_feedback	1	Describes design feedback in initial development.	B	”But yeah when we take into account the adswag it’s happening right now for instance user the company which working on designs as a point of their work is sending survey to the adswag clients about what they think about one version of an app comparing to they showed them different versions of an app and they expected some feedback.” - Interviewee Expert B
#ID_expectation_management	1	Describes expectation management.	C	”What requirements do your end users have? ”Certainly, it’s about expectation management.The higher you set expectations, the more they will demand.” - Interviewee Expert C
#ID_first_impressions_matters	1	Describes how first impressions matters.	C	”A delayed response results in a poor experience ” - Interviewee Expert C

Code	Frequent	Definition	Expert	Example
#ID_improve_by_feedback	1	Describes the importance of feedback in initial development.	B	” But I would say that in some of these points it’s the common situation that you have to listen your customers to make product better. it’s not even for a web application. It’s for every product that you have to react on a users’s client’s needs.” - Interviewee Expert B
#ID_information_instant	1	Describes the importance of instant information.	C	”They always want instant information, so instant responses are necessary ” - Interviewee Expert C
#ID_integration_centralize_controlle	1	Describes the importance of integration controlle.	C	”Ideally, you should centralize control so you can set the pace and not depend on sixty different parties, as we experienced with dynamic branded content” - Interviewee Expert C
#ID_integration_lack_effort	1	Describes integration lack effort of publishers.	C	”Publishers often don’t read all the documentation for implementation” - Interviewee Expert C
#ID_integration_pace_controlle	1	Describes the importance of pace controlle in development.	C	”Ideally, you should centralize control so you can set the pace and not depend on sixty different parties, as we experienced with dynamic branded content ” - Interviewee Expert C
#ID_integration_simplicity_urgency	1	Describes the integration simplicity urgency.	C	”Because they have relatively little knowledge of custom things, to roll out quickly you either have to build it yourself or simplify it” - Interviewee Expert C
#ID_keep_low_expactations	1	Describes the importance of setting low expactations.	C	”The higher you set expectations, the more they will demand. It’s important to keep expectations low ” - Interviewee Expert C

Code	Frequent	Definition	Expert	Example
#ID_knowledge_gap_agencies_and_publishers	1	Describes the knowledge gap agencies and publishers.	C	”Often, the website owners aren’t knowledgeable enough to implement simple things and get redirected to the next agency, which are usually small firms or agencies that aren’t very quick or smart. This leads to going through many layers due to the lack of knowledge, which is very challenging ” - Interviewee Expert C
#ID_long_integration_process	1	Describes the long integration process with publishers.	C	”It’s a long process and can take a long time ” - Interviewee Expert C
#ID_no_customization	1	Describes the importance of no customization.	A	”But exceptions are problematic” - Interviewee Expert A
#ID_no_strictly_development_method	1	Describes that there is no strictly development method in developing AdTech.	B	”It’s I have a feeling that we don’t stick strictly to some developing method. we just think what will work for us in a short term to deliver some MVP” - Interviewee Expert B
#ID_publisher_dependency	1	Describes the publishers dependency.	C	”We often need publishers to deliver the ads” - Interviewee Expert C
#ID_retain_mvp	1	Describes the importance of retaining the mvp.	C	”You need to reduce the expectation pattern and not share too much” - Interviewee Expert C
#ID_ROI	1	Describes the importance of ROI.	C	”A critical consideration for launching is to ensure that we have the funds to cover costs or that there is a direct return on investment ” - Interviewee Expert C
#ID_scrum	3	Describes the use of Scrum as a development method	D	“We use Scrum. We have a sprint demo every two weeks and a stand-up two or three times a week, depending on what’s needed ID_Scrum.” - Interviewee Expert D
#ID_servicing_needed	1	Describes how servicing during development is needed.	C	”If something is wrong, they want it fixed immediately a delayed response results in a poor experience. Therefore, maintaining a high service level is essential” - Interviewee Expert C

Code	Frequent	Definition	Expert	Example
#ID_servicing_ software_updates	1	Describes servicing when software updates.	C	”I do maintain customer contact, for example, telling them to be careful and to call me if something goes wrong ” - Interviewee Expert C
#ID_stability	1	Describes the importance of stable AdTech	D	“So our main challenge is consolidating everything on a single platform and ensuring stability. ” - Interviewee Expert D
#ID_start_with_ focus_on_product_platform	1	Describes the importance of starting with focus on product platform.	B	”When you see that you have more and more users then you think more about or maybe scalability will never be a problem for your software and you will Spend a lot of time on making some optimizations at the beginning. so it’s better to focus at the beginning more on a product than on scalability.” - Interviewee Expert B
#ID_standardize	1	Describes the importance of standardizing making no exceptions for customization	C	”Don’t share too much, as it leads to more questions. Avoid too many exceptions” - Interviewee Expert C
#ID_test_could_ be_done_for_all_ users	1	Describes the testing method that affects all users.	A	”We’re still small, so we don’t do that yet. Currently, it’s rolled out to all users. This might change as usage grows.” - Interviewee Expert A
#ID_test_could_ be_done_to_ small_group_first	1	Describes that tests could be done to small group first.	A	”You can do A/B testing or roll out to a subset of users first” - Interviewee Expert A
#ID_testmethod_ staging_to_test_ or_acceptance_ stage	1	Describes the testmethod that uses a staging area to test or acceptance area.	A	”In principle, we have automated tests. New features are first rolled out on staging and tested there. If they pass, they are rolled out to production. Later, maybe an acceptance environment is added ” - Interviewee Expert A
#ID_urgent_error_fix	1	Describes the importance of urgent error fix.	C	”If something is wrong, they want it fixed immediately” - Interviewee Expert C
#ID_development_time_increases_when_ software_matures	1	Describes that development time increases when software matures.	A	”The first version took a week, now a new feature also requires a whole week of work. ” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#ID_dependencies_increase_complexity	1	Describes that dependencies increase complexity.	A	"As the software develops, all dependencies on each other increase. If you want to do something, you have to consider everything, which is always the case " - Interviewee Expert A
#ID_AdTech_reusability	1	Describes the importance of software reusability in AdTech.	A	"The market operates in cycles, for example: 20 years ago were the first ads, it's becoming more automated, then other sectors like retail (e.g., AH) emerge. These sectors need to be rebuilt from scratch. So, AdTech can be reused " - Interviewee Expert A
#ID_work_in_components_to_reuse	1	Describes how working in components helps reusability.	A	"It depends on whom you want to serve. If something works, it needs to be simple and applicable to many of your end-users. Media agencies sometimes want exceptions, only to build a component that can be reused" - Interviewee Expert A
#ID_continuous_development	1	Describes the continuous development in AdTech.	A	"With AdTech software, yes, there is a maintenance phase on the component level. But across the entire package, you're always working on a new component, so there's always something new." - Interviewee Expert A
#ID_development_is_expensive	1	Describes how development of AdTech is expensive.	C	"They try to acquire a lot of technology because developing technology is very expensive. Innovation costs a lot of money" - Interviewee Expert C
#ID_implementation_difficulty	1	Describes the difficulty of implementing AdTech..	C	"Not easily, because there is a high barrier to entry. That's due to the difficulty of implementation " - Interviewee Expert C
#ID_components_help_reducing_complexity	1	Describes how components help reducing complexity.	A	"Media agencies sometimes want exceptions, only to build a component that can be reused" - Interviewee Expert A
#RA_AdTech_cant_be_unplugged_quickly_due_to_contracts	1	Describes that AdTech cannot be unplugged quickly due to contracts.	A	"If you're connected everywhere, many scripts run on publishers that can't find a server anymore. If other companies are dependent, you can't just unplug it." - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#RA_ad_performance_cost	1	Describes that replacing AdTech affects ad performance.	C	”There are also performance costs because you have to provide new inputs, and the new system needs to optimize” - Interviewee Expert C
#RA_adserver_is_difficult_to_replace_piece_by_piece_or_slowly_switching_clients	1	Describes that AdTech is difficult to replace piece by piece or slowly switching clients to different AdTech.	A	”For example, with a adserver, it’s very complicated. Replacing it or replacing it piece by piece or building something alongside and gradually switching clients over a long period. You can’t stop.” - Interviewee Expert A
#RA_consequences_of_errors	1	Describes the possible consquences of erros when you try to replace AdTech.	D	“Are AdTech systems harder to replace than standard software? Yes, because typically a lot more money is involved, and there are larger stakeholders behind them. The consequences of errors are bigger. In a typical system, if one payment fails, that might be a few hundred euros lost. . . ” - Interviewee Expert D
#RA_contract_dependencies	1	Describes contract dependencies in replacing AdTech.	C	”So switching is relatively easy in terms of relationships, through a one-year contract or something” - Interviewee Expert C
#RA_compex	1	Describes the complexity of replacing AdTech.	D	“Can large AdTech systems be replaced easily once they’re established? No, it’s very difficult. There’s a ton of complexity behind them, and you have to be extremely accurate with request counting and tracking. . . . a massive undertaking—lots of testing, careful planning.” - Interviewee Expert D
#RA_data_loss	1	Describes data loss when switching of AdTech.	C	”You can switch easily; it just takes time and money. You have certain datasets in a particular system and your learnings, etc” - Interviewee Expert C
#RA_hardcoded_scripts_at_publishers	1	Describes the use of hardcoded scripts at website of publishers.	A	”Some things are hardcoded since they need to be on the publisher’s site.” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#RA_education_cost	1	Describes the cost of educating employee when switching to different AdTech	C	”It’s all integrated, and replacing it takes time, and time is money.Then you first lose money because you have to retrain everyone, but it’s not impossible.” - Interviewee Expert C
#RA_knowledge_cost	1	Describes the knowledge cost when switching to different AdTech.	C	”All your staff have a lot of knowledge in one system, which is lost” - Interviewee Expert C
#RA_migrating_data	1	Describes the migration of data during replacement	D	“Building a new API from scratch and migrating data is a massive undertaking—lots of testing, careful planning” - Interviewee Expert D
#RA_money_involvement	1	Describes the involvement of money when replacing AdTech	D	“...larger stakeholders behind them. The consequences of errors are bigger. In AdTech, you can be talking tens of thousands of euros a day...” - Interviewee Expert D
#RA_providers_dependency	3	Describes that providers of cloud make it difficult to change.	A	”Providers lock you in; that’s their revenue model, like with Vercel” - Interviewee Expert A
#RA_replacement_needs_thorough_planning	2	Describes that AdTech replacement needs thorough planning.	B	”it depends how this software is built, what you have currently. it’s a topic for many meetings usually to plan replacement of something with something.” - Interviewee Expert B
#RA_replacement_requires_extensive_planning_to_minimize_disruptions_for_clients	2	Describes that AdTech replacement requires extensive planning to minimize disruptions for clients.	A	”If you’re connected everywhere, many scripts run on publishers that can’t find a server anymore. If other companies are dependent, you can’t just unplug it.” - Interviewee Expert A
#RA_shutdown	1	Describes that if the software is not used anymore you should shut it down.	D	“What if your own branded content software isn’t used anymore? We’d probably scrap it. We might reuse some of the API logic if it’s still valuable...then shut it down. Unless we find another purpose for it.” - Interviewee Expert D

Code	Frequent	Definition	Expert	Example
#RA_tracking_pixel_replacement	1	Describes tracking pixel replacement.	C	”If they switch to another system, all those people need to be retrained for the new system, and all the pixels need to be replaced and everything remeasured, which can certainly take half a year” - Interviewee Expert C
#RA_lots_of_testing	1	Describes the amount of testing before a replacement can happen.	D	“. . . a massive undertaking—lots of testing, careful planning.” - Interviewee Expert D
#RA_think_ahead_about_replacing_to_avoid_trouble	1	Describes the importance of thinking ahead about replacing to avoid trouble.	A	”You can’t just migrate away from the cloud, for example. It’s also difficult because it always needs to be running, like the adserver. If you don’t think it through initially, you’ll get into trouble. ” - Interviewee Expert A
#RA_unplug_if_not_used_and_possible	1	Describes that AdTech should be unpluged if not used and possible.	A	”What happens if your software is no longer used?You unplug it ” - Interviewee Expert A
#Regulations_best_effort	1	Describes the importance of regulation’s best effort.	C	”Ultimately, you have to make a best effort; that’s also in the KSA regulations, which also apply to betting—you must make a best effort.Regulations can conflict with each other, so some elements clash. For example, you can store certain data under KSA but not under GDPR, putting you in a dilemma. Therefore, you have to make a best effort to comply with both KSA and GDPR as best as possible; it’s a human judgment.” - Interviewee Expert C
#SA_AdTech_cant_have_downtime	1	Describes that AdTech cannot have downtime.	A	”If you’re live, you’re live and can’t be offline for an hour for maintenance” - Interviewee Expert A
#SA_api_needs_scalable	2	Describes that API of AdTech needs to be scalable.	A	”Additionally, we have an API that ensures content is displayed. If it fails, the product collapses. This must always work” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#SA_scale_on_user_amount	1	Describes that AdTech should scale on the amount of users.	B	”Actually it depend on how many users your system had has so yeah I would say that it’s always a balance and it’s always depend on which stage the project is.” - Interviewee Expert B
#SA_budget_limits_in_cloud	2	Describes budget limits in cloud.	A	”Most of it runs in the cloud and needs to scale automatically. We have budget limits” - Interviewee Expert A
#SA_cloud_auto_scaling	2	Describes the use of the cloud with auto scaling.	A	”Most of it runs in the cloud and needs to scale automatically. We have budget limits” - Interviewee Expert A
#SA_cloud_based	1	Describes the use of the cloud.	B	”I mean that for instance several years ago people tried to set up database on their own server but right now they prefer to use hosted solution that database it’s something which is hosted by which is maintained by external company who is really take care of any parameters of this piece of infrastructure.” - Interviewee Expert B
#SA_cloud_cost	1	Describes that the cloud scales as long as you pay	A	”Everything runs in the cloud, so you can scale as long as you pay” - Interviewee Expert A
#SA_design_important_to_grow_AdTech	2	Describes why design is important to grow AdTech.	A	”If not designed correctly, it becomes very expensive quickly” - Interviewee Expert A
#SA_later_focus_on_scalability_platform	1	Describes the late prioritiy of scaling	B	”When you see that you have more and more users then you think more about or maybe scalability will never be a problem for your software and you will Spend a lot of time on making some optimizations at the beginning. so it’s better to focus at the beginning more on a product than on a scale scal scalability.” - Interviewee Expert B
#SA_logging_needs_scalable	1	Describes the importance of scaling for logging.	A	Which components of AdTech need to be scalable from the beginning?”Everyting related to logging” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#SA_no_downtime	1	Describesthe importance of no downtime.	B	”But I can imagine that if you have a real working ad server for instance the main goal is to not have any downtime. So probably you have to design the deployment and some migrations in a way to not break the app.” - Interviewee Expert B
#SA_one_client_could_have_big_impact	1	Describes the impact that one client could have on the cloud resources.	A	”If a new client comes tomorrow, it could have a significant impact” - Interviewee Expert A
#SA_outsourcing_infrastructure	1	Describes the choice of outsourcing an infrastructure.	B	”So you have to incorporate also analytic on the condition of your infrastructure to keep control of your system. yeah, but I think that it’s also changed a bit in the past years because the trend is that companies would like to outsource some parts of their infrastructure to companies which are specialized. in this” - Interviewee Expert B
#SA_scale_cost	1	Describes the cost of scaling in the cloud.	A	”Additionally, it must be scalable, automatically scalable, and cost-controlled” - Interviewee Expert A
#SA_some_parts_require_scalability_at_start	1	Describes that some parts require scalability at start.	B	”Of course, some parts of a system has to be scalable from the beginning in case of our software where we track activity of the users on a pages where is a article displayed because from the beginning you will have quite high load quite high amount of requests. But still, probably in the later stage of the system when we will onboard more customers and we will have more users and more campaigns.” - Interviewee Expert B
#SA_logging_needs_scalable	1	Describes that resources need to scale for the logging.	A	Which components of AdTech need to be scalable from the beginning?”Everyting related to logging” - Interviewee Expert A
#SA_budget_limits_in_cloud	1	Describes the budget limits in cloud.	A	”Most of it runs in the cloud and needs to scale automatically. We have budget limits” - Interviewee Expert A

Code	Frequent	Definition	Expert	Example
#SA_fast_growth	1	Describes that AdTech products can grow very fast.	A	”With advertising, you grow very quickly because you operate on various websites rapidly. So it quickly involves millions of requests per day and billions per month” - Interviewee Expert A