Photo credit: https://www.teamviewer.com/en/augmented-reality/

An Implementation Methodology for Augmented Reality Applications

Master Graduation Thesis

Kristen (Thi Ha Giang) Phan | s2616874 | February 6th, 2023

UNIVERSITY OF TWENTE.

LIQUID STUDIO.NL BE DIFFERENT

accenture

COLOPHON

Final Project Report

Title: An Implementation Methodology for Augmented Reality Applications
Date: February 6th, 2023
Classification: Open

Author
Name: Kristen (Thi Ha Giang) Phan
Student number: s2616874
Email: g.t.h.phan@student.utwente.nl
Study program: MSc Business Information Technology
Institute: University of Twente
PO Box 217
7500 AE Enschede
The Netherlands

Supervisors

Name: Prof. Dr. Maria E. Iacob
Email: m.e.iacob@utwente.nl
Organization: University of Twente
Faculty: Behavioral, Management and Social Sciences (BMS)

Name: Dr-Ing. Florian W. Hahn
Email: f.w.hahn@utwente.nl
Organization: University of Twente
Faculty: Electrical Engineering, Mathematics and Computer Science (EEMCS)

Name: Ilia Shakitko
Email: ilia.shakitko@accenture.com
Organization: Accenture Netherlands
Department: Liquid Studio
Job title: Tech Innovation Lead

ABSTRACT

Objective:
To develop an industry-agnostic, tool-agnostic Augmented Reality implementation methodology (ARIM) for guiding the implementation of AR use cases across different industries (i.e., industry-agnostic). The methodology should assist project teams in choosing suitable AR tools for their target use cases instead of prescribing any specific tool (i.e., tool-agnostic).

Methods:
The research consists of phase 1 literature review and phase 2 artifact design. The literature review sets the scientific foundation for the artifact and identified research gaps in existing AR implementation frameworks.

The identified research gaps translate to three key questions to be answered by the ARIM:
1. How to determine if a process can be enhanced with the use of AR?
2. How to determine which AR platform (webAR vs. native AR vs. AR headset) is most suitable for the selected process?
3. How to determine which AR development tools (tracking and rendering engines) are most suitable for the selected process and AR platform?

To answer the first question, 39 academic and industry AR use cases for multiple sectors (industrials, entertainment, healthcare, education, emergency & rescue, military, and miscellaneous sectors) were reviewed to synthesize common characteristics which make a process an ideal candidate for AR.

To answer the remaining two questions, success factors for AR adoptions from existing literature were translated to high-level selection criteria for AR platforms and AR development tools. These high-level selection criteria are further contextualized with example AR use cases and subsequently decomposed into low-level selection criteria. With this approach, the ARIM is firmly built upon AR adoption success factors, enabling AR solutions implemented using the methodology to achieve high user acceptance.

Results:
The proposed ARIM is illustrated in Figure 46. For demonstrating how the proposed ARIM can be used in practice, a webAR application for monitoring the health of houseplants is implemented as guided by the ARIM.

For validating how useful the ARIM is, three semi-structured interviews were conducted with Accenture's industry experts. Experts responded positively to the simplicity, thoroughness, and value-add of the methodology. Additionally, expert feedback was used to further finetune and align the methodology with Accenture's current practice – see the revised ARIM in Figure 52.

Keywords: Augmented reality, augmented reality implementation methodology, WebAR, web-based augmented reality.

Contents

CHAPTER 1. INTRODUCTION

1.1 Introduction to Augmented Reality (AR)

Launched on July 6, 2016, Pokémon Go[1], a location-based Augmented Reality (AR) mobile game, took the world by storm and set five world records with regards to the amount of revenue it generated and the number of times it was downloaded within the first month of release[2].

The premise of the game is simple: As users walk around in the real world, Pokémon characters appear as 3D models on the users' device screen. When standing close enough to the virtual Pokémon's, users can "throw" a ball at the Pokémon's and capture them by swiping the red/white ball in their direction. The objective is to capture as many Pokémon's as possible – see Figure 1.

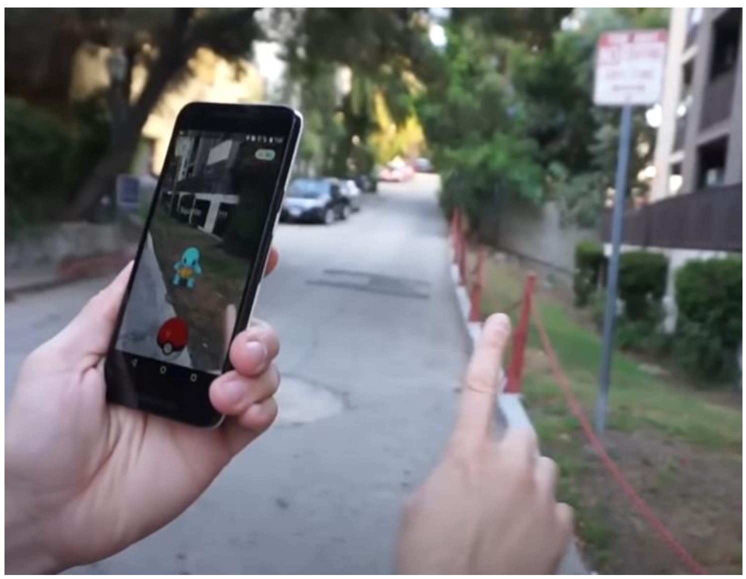


Figure 1. Pokémon Go – a Location-Based AR Mobile Game [3]

The appeal of Pokémon Go and AR at large comes from a novel blend of real-world objects (e.g., the physical space the users are living in) and virtual objects (i.e., the virtual 3D Pokémons). *Defined as "a medium in which digital information is overlaid on the physical world […] and is interactive in real time", AR stands out as an enhanced and simplified way to consume and interact with information* [1].

However, the use of AR does not stop at gaming and entertainment. For example, with the use of the Microsoft HoloLens headset[4], a surgeon can view a patient's medical records projected in

[1] https://pokemongolive.com/en/
[2] https://www.guinnessworldrecords.com/news/2016/8/pokemon-go-catches-five-world-records-439327
[3] https://www.youtube.com/watch?v=4UFwzI6ArVg
[4] https://en.wikipedia.org/wiki/Microsoft_HoloLens

3D from the headset's field-of-view and start a video call to collaborate remotely with another surgeon during the surgery[5].

With the rise of enabling technologies such as fast 5G network connections, powerful AR hardware like Microsoft HoloLens, and maturing AR development tools such as Unity game engine[6], Vuforia SDK[7] (software development kit) and AR.js webAR open-source library[8], developers have more tools at their disposal to build powerful AR applications running on fast internet connection, allowing AR to graduate from a novel technology to an accessible medium for consuming and interacting with information.

1.2 AR Platforms: WebAR vs. Native AR vs. AR Headset

The excitement continues as AR expands its footprint to web technologies, meaning users can now participate in an AR experience right from a web browser.

This class of web-based AR applications (or "*webAR*" for short) is to be distinguished from a class of native mobile AR applications (or "*native AR*" for short) which runs on a specific mobile operation system such as Android or iOS and from another class of AR applications on AR headsets (or "*AR-on-headset*" for short) which run on AR-dedicated hardware such as Microsoft HoloLens [2].

Although native AR and AR headset outperform webAR apps in terms of *tracking* capabilities – the ability to detect a physical object in the physical world to which a virtual object is anchored to (more on tracking in section AR by Tracking Techniques), webAR offers several advantages over native AR and AR-on-headset including but not limited to [3] [4]:

- *Unlike native AR applications*, webAR applications run directly from a browser – no need to download and install any app on user's device. Additionally, webAR has a faster development cycle because it can rely on open-source tools and more convenient release and bug fixing processes since changes can be tested and rolled out when the new HTML/ CSS/ JavaScript files are uploaded to the web server. Finally, webAR has lower development costs as it does not incur publications fee from official app stores like Apple's App Store for iOS apps or Google Play for Android apps.
- *Unlike AR-on-headset applications*, webAR applications does not require expensive AR-dedicated hardware.

While the advantages of webAR discussed above are impactful from the end user's perspective, there is a lack of concrete guidance to help stakeholders and developers choose which AR platform to implement for a specific use case.

[5] https://www.youtube.com/watch?v=x1XnkU31gtk
[6] https://unity.com/
[7] https://www.ptc.com/en/products/vuforia
[8] https://ar-js-org.github.io/AR.js-Docs/

1.3 Research Objective, Research Questions (RQs), and Research Methodology

The objective of this graduation thesis is to create an *industry-agnostic, tool-agnostic Augmented Reality Implementation Methodology (ARIM)* which guides business stakeholders and software developers throughout the implementation of AR applications.

The proposed ARIM should provide high-level steps with concrete guidance for how to carry out each step throughout different phases of the implementation process so that the resulting AR applications can achieve high user acceptance.

The proposed ARIM is designed to be industry-agnostic and tool-agnostic so that the methodology 1) is versatile enough to be used for clients who operate in different industries and would like to implement AR solutions to improve their business processes and 2) helps developers assess and choose appropriate AR development tools to fulfill the requirements of the clients' target use cases rather than biasing developers towards any specific tools.

The thesis consists of two phases – see Figure 2. Research Methodology:

- Phase 1: Systematic literature review
    - The objective of phase 1 is to conduct a systematic literature review, answer RQ#1-8 (i.e., knowledge questions), and lay the scientific foundation for the ARIM to be created in phase 2 the graduation thesis.
    - To answer these knowledge questions, different search queries are composed and targeted for each of the following RQs – see section "Literature Review Protocol" for details on how the systematic literature review is conducted.
- Phase 2: The design, demonstration, evaluation, and revison of an ARIM
    - The objective of phase 2 is to answer RQ#9 (design question) i.e., design an ARIM based on the findings from the literature review.
    - A webAR app is then developed to demonstrate how the proposed ARIM can be used in practice, answering RQ#10 (demonstration question)
    - The proposed ARIM is evaluated for its utility through expert interviews, answering RQ#11.
    - Expert feedbacks are then used to further finetune the proposed ARIM.

Below are the RQs and the motivation behind each RQ.

| RQ#1 (knowledge question): What is AR?
This question aims to provide a high-level understanding of AR.

| RQ#2 (knowledge question): How does AR work?
This question aims to provide technical understanding of the mechanism underlying AR irrespective of the platform through with the AR experience is delivered: via a web browser (webAR), via a native mobile application (native AR), or via an AR headset (AR-on-headset).

| RQ#3 (knowledge question): What factors influence AR adoption?
This question aims to add insights to be considered as part of the ARIM for improved user acceptance.

| RQ#4 (knowledge question): From a technical standpoint, what is the difference between webAR vs. native AR vs. AR-on-headset application development?
This question aims to explore the development of webAR, native AR and AR-on-headset applications and subsequently derive the differences in webAR v. native AR v. AR-on-headset implementation.

| RQ#5 (knowledge question): From a technical standpoint, what is the difference between webAR vs. regular, non-AR web application development?
This question aims to explore the development tools and methods commonly used for the development of regular, non-AR web applications and subsequently derive the differences in webAR development and non-AR web development.

| RQ#6 (knowledge question): What is cloud serverless infrastructure?
There has been a growing acceptance of serverless architecture because of reduced operation costs, flexible scalability, enhanced performance, and increased developer productivity [5].

This question seeks to explore different options for serverless cloud infrastructure which serve as a starting point for the deployment of a webAR app in the second phase of the graduation thesis.

| RQ#7 (knowledge question): What is existing research on AR implementation frameworks?
This question seeks to explore existing AR implementation frameworks in the literature and identify the deficiencies in those frameworks. The ARIM created in the second phase of the graduation thesis will be built up on these frameworks while addressing the identified deficiencies – see Table 9 for a summary of the deficiencies and how the ARIM intends to address them.

| RQ#8 (design question): What does an industry-agnostic, tool-agnostic AR implementation methodology (ARIM) look like?

The answer to this RQ is the main artifact to be delivered in this graduation thesis – a solution template to guide the design, development, and deployment of AR applications for maximized user acceptance.

The proposed ARIM is designed to be industry-agnostic and tool-agnostic, enabling the teams to use this methodology for different client projects in different industries and selecting appropriate AR development tools for the target use cases rather than biasing the teams towards any specific tools.

RQ#8 is further broken down into sub-RQs to address the research gaps identified in existing literature – see section Research Gap Summary and Research Output Scope for details.

|| RQ#8.1: What makes a process a good candidate for enhancement with the use of AR?
|| RQ#8.2: How to choose between webAR vs. native AR vs. AR headsets for a selected process?
|| RQ#8.3: How to choose between different AR development tools for a selected process and AR platform?
|| RQ#8.4: What are the requirements for the development of AR apps?
|| RQ#8.5: What are the requirements for the deployment of AR apps?

| RQ#9 (demonstration question): How can the proposed ARIM be used in practice?
To demonstrate how the proposed ARIM can be used in practice, a webAR app will be implemented.

| RQ#10 (validation question): What is the utility of the proposed ARIM?
To assess how useful the proposed ARIM is, semi-structured interviews will be conducted with professionals who have work experience with augmented reality.

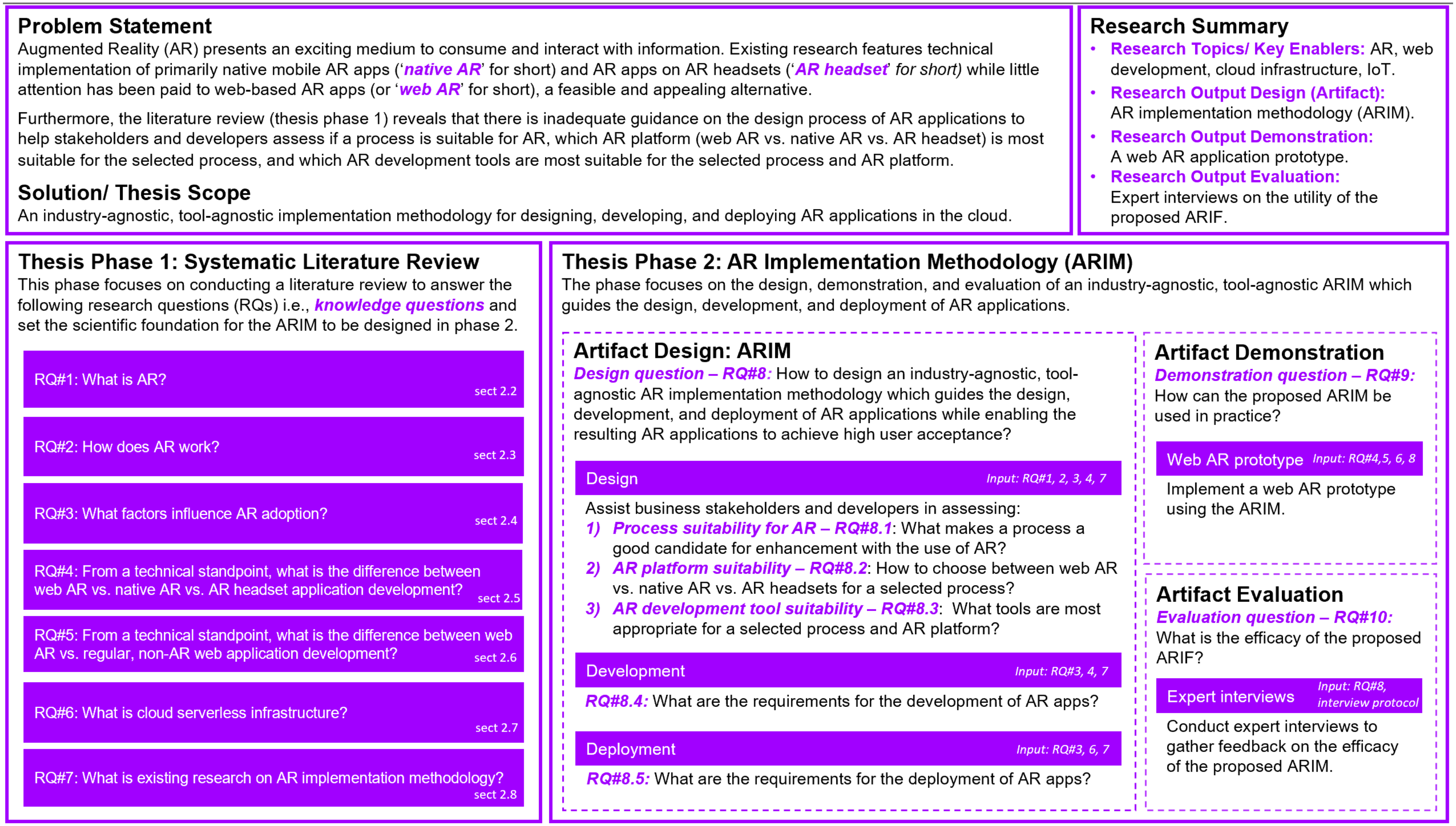


Figure 2. Research Methodology

1.4 Chapter Summary and Thesis Outlook

This chapter sets the stage for the graduation thesis which consists of two phases:

- Phase 1: Conduct a systematic literature review on the implementation of AR applications – the focus on this report.
- Phase 2: Design, demonstrate, and evaluate an industry-agnostic, tool-agnostic AR implementation methodology (ARIM) which guides the design, development, and deployment of AR applications for maximized user acceptance.

Section 1.1 Introduction to Augmented Reality (AR) introduces AR.
Section 1.2 AR Platforms: WebAR vs. Native AR vs. AR Headset explains different AR platforms.
Section 1.3 Research Objective, Research Questions (RQs), and Research Methodology outlines the research questions to be answered in this research by conducting a systematic literature review. The objective of these research questions is to lay the scientific foundation for the ARIM to be designed in phase 2 of the graduation thesis - see Figure 2 for the research methodology.

CHAPTER 2. SYSTEMATIC LITERATURE REVIEW will detail the protocol of the systematic literature review and summarize the findings from the literature review.

CHAPTER 2. SYSTEMATIC LITERATURE REVIEW

2.1 Literature Review Protocol

Table 1 documents the literature review protocol used for this research as adapted from the systematic literature review procedure proposed in book titled “Guidelines for performing Systematic Literature Reviews in Software Engineering” by Kitchenham and Charters.

Table 2 itemizes the search queries and quality checklists for each query.

Table 3 itemizes the data attributes of the data extraction form.

A total of *111 papers* are selected from search query results for detailed examination using a data extraction form, *86* of which (excluding paper [6] regarding how to conduct a systematic literature review) are referenced in this report.

Protocol Component	Protocol Sub-component	Component Description	Component Details
Search strategy	Search terms	Includes synonyms, abbreviations, alternative spellings, terms from reference list, etc. Search terms are finalized through trial-and-error by running test search queries with potential search terms.	• Augmented reality, AR • AR web application, AR web app, web-based AR, webAR, webAR • web application, web app, web development • model generation, model creation, content creation, content generation, content management, data, metadata • interactivity • software requirement, software architecture • SDK, software development kit, IDE, integrated development environment, augmented reality library, AR library, toolkit, tool kit • adoption, user acceptance, acceptance, success, success factor, resistance, motivation, user experience, evaluation, opportunity, opportunities, challenge, threat • cloud serverless infrastructure, serverless infrastructure, serverless architecture, serverless computing • implementation framework, framework, implementation, design methodology, development process, development workflow
	Search queries	Constructed from search terms using ADD, OR, NOT operator.	Each search query is mapped to the relevant research question(s) the query is intended to provide insights for- see Table 2. Not included in Table 2 are additional exploratory research based on the results of search queries.
	Resources	Databases/ resources where the search queries are executed.	IEEExplore: https://ieeexplore-ieee-org.ezproxy2.utwente.nl/search/advanced Google scholar: https://scholar.google.com/ Google search for looking up industry literature: https://www.google.com/
Study selection criteria	Inclusion criteria	Query results which meet the inclusion criteria proceed to the *study quality assessment phase.*	• Open access paper • Papers which involve either native AR or webAR even though this research focuses on webAR for two reasons 1) Unlike AR applications on dedicated hardware, there are similarities in the technical development process of native AR and webAR. 2) There is a limited number of publications specifically for webAR alone if native AR is excluded from search queries.
	Exclusion criteria	Query results which meet the exclusion criteria do <u>not</u> proceed to the *study quality assessment phase.*	• Non-English papers. • Duplicate papers. • Papers which involve the use of dedicated hardware such as HMD (head mounted displays), HoloLens, Google Glass, Kinect, special tracking sensors.
Study quality assessment	Quality checklist (i.e., more detailed inclusion/ exclusion criteria)	Query results which satisfy the quality assessment are examined in detail using the *data extraction form*.	Specific quality checklists for different search queries can be found in Table 2.
Data extraction strategy	Data extraction form	Used to systematically extract key information from the final list of papers filtered from the initial query results.	The data extraction form outlines attributes to be extracted each paper. An example of an attribute is whether the paper discusses the software architecture of the prototype implemented. A full list of attributes is outlined in Table 3.

Table 1. Literature Review Protocol

#	Search Query	Related RQs	Quality Checklist
1	"augmented reality" OR "AR" AND "tracking"	RQ#2: How does AR work?	• Papers should give an overview of AR classifications by tracking methods, not the mathematical concepts behind tracking algorithms.
2	"augmented reality" OR "AR" AND "model" OR "3D model" OR "model generation" OR "model creation" OR "content" OR "content creation" OR "content generation" OR "data" OR "metadata"	RQ#2: How does AR work?	• Papers should indicate the tools for creating and storing AR assets (e.g., creating 3D models using modeling graphic tool and storing the models in a binary storage solution).
3	"augmented reality" OR "AR" AND "render" OR "interactivity" OR "software requirement" OR "software architecture"	RQ#2: How does AR work?	• Papers should discuss how AR contents are rendered and the types of user interactivities implemented for AR applications.
4	"augmented reality" OR "AR" AND "adoption" OR "user acceptance" OR "acceptance" OR "success" OR "success factor" OR "resistance" OR "motivation" OR "user experience" OR "evaluation" OR "opportunity" OR "opportunities" OR "challenge" OR "threat"	RQ#3: What factors influence AR adoption?	• Papers should provide new, AR-specific insights into factors that influence AR adoption which go beyond the scope of traditional technology adoption models such as TAM[9] to measure user acceptance. An example of such factors is that webAR is only appropriate for use cases where a stable internet connection can be guaranteed.
5	"augmented reality" OR "AR" AND "SDK" OR "software development kit" OR "IDE" OR "integrated development environment" OR "augmented reality library" OR "AR library" OR "toolkit" OR "tool kit"	RQ#4: For webAR, what development tools are available? RQ#5: From a technical standpoint, what is the difference between webAR vs. native AR vs. AR-on-headset application development?	• Papers should discuss the implementation of an AR application prototype. • Papers should discuss the technology stack, meaning the names of the tools used to implement the protype should be mentioned.
6	"web application" OR "web app" OR "web development" NOT "web-based augmented reality" OR "web-based AR" OR "webAR" OR "webAR" AND "implementation framework" OR "framework" OR "implementation" OR "design methodology" OR "development process" OR "development workflow" OR "software requirement" OR "software architecture"	RQ#6: From a technical standpoint, what is the difference between webAR vs. regular, non-AR web application development?	• Papers should discuss the tools and methods used for the development of non-AR web applications. The objective is to identify the differences between webAR and non-AR web application development.
7	"cloud serverless infrastructure" or "serverless infrastructure" OR "serverless architecture" OR "serverless computing"	RQ#7: What is cloud serverless infrastructure?	• Papers should discuss key characteristics of serverless cloud services.
8	"augmented reality" OR "AR" AND "implementation framework" OR "framework" OR "implementation" OR "design methodology" OR "development process" OR "development workflow"	RQ#8: What is existing research on AR implementation frameworks?	• Papers should discuss AR implementation frameworks which guide the creation of AR applications.

Table 2. Search Queries and Quality Checklists

[9] https://en.wikipedia.org/wiki/Technology_acceptance_model

#	Paper Attribute	Possible Values
1	Prototype	• Yes: if the paper documents an implementation of an AR app, either native AR app or webAR app • No
2	Software architecture	• Yes: if the paper discusses the software architecture of the prototype implemented • No
3	Prototype evaluation	• Yes: if the prototype is evaluated once implemented e.g., how easy to use the users find the prototype • No: no prototype evaluation
4	Tracking technique	• Marker-based fiducial tracking • Marker-based image tracking • Object tracking • Face tracking • Location tracking • Surface tracking • Spatial tracking
5	Cloud infrastructure	• Yes: if the prototype is deployed to the cloud • No
6	AR Platform	• WebAR: if the prototype is implemented as a web application running from a web browser. • Native AR: if the prototype is implemented for a native mobile application for Android, iOS, or Windows • AR headset: if the prototype is implemented for AR head-mounted displays such as Microsoft HoloLens
7	Technology stack	• Yes: if the technology stack (specific tools, libraries, frameworks) used to implement the prototype is discussed, make notes of the names of the tools part of the technology stack. • No
8	AR content	• Yes: if the paper discusses how the AR content is created and stored. • No
9	Adoption factors	• Yes: if the paper discusses factors which influence AR adoption, make notes of these factors. • No
9	Non-AR web application	• Yes: if the paper discusses the tools and/ or methods commonly used for the development of non-AR web applications • No
10	AR implementation methodology	• Yes: if the paper discusses the general process which guides the design, development, and/ or deployment of AR applications • No
11	Additional notes	• Add any additional details which are relevant to the RQ.

Table 3. Data Extraction Attributes

2.2 RQ#1: What Is AR?

Reality-Virtuality Continuum

The terms Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR) are often discussed in the same context. However, they have a distinct scope on the reality-virtuality continuum.

On the far left of Figure 3 is the real environment where everything that constitutes this environment is real and not computer-generated. AR environment is achieved by adding computer-generated, virtual contents to the real environment to “augment” the real environment. Think Pokémon Go mobile AR game as discussed in section 1.1 Introduction to Augmented Reality (AR).

On the far right of Figure 3 is the virtual environment, also known as VR, where everything that constitutes this environment is not real and computer-generated. Think a VR headset like the Oculus Quest[10] where users see only virtual contents through the headset. Augmented Virtuality (AV) is achieved by augmenting the virtual environment “through the use of real (i.e., unmodelled) imaging data” [7].

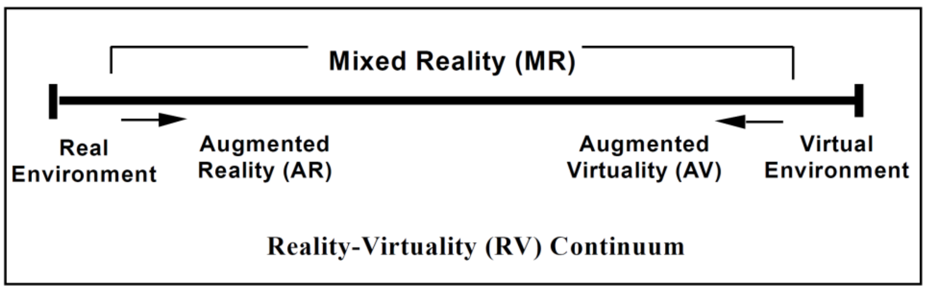


Figure 3. Reality-Virtuality Continuum [7]

Since the introduction of one of the earliest VR systems called the Sensorama[11] in 1962, AR/VR inventions have come a long way to add a new dimension to how humans experience and interact with the world.

AR Definition

Expanding on the definition of AR as a “a medium in which digital information is overlaid on the physical world […] and is interactive in real time” in section 1.1 Introduction to Augmented Reality (AR), Azuma, a prominent AR researcher, defines AR as a system with three key characteristics [8]:

1. AR combines real and virtual environment
2. AR is interactive in real-time
3. AR contents are registered in 3D

[10] https://www.oculus.com/experiences/quest/
[11] https://en.wikipedia.org/wiki/Sensorama

Augmentation of Human Senses: Vision, Hearing, Touch, Taste, Smell

Because AR aims to augment the real world and humans experience the real world through five senses (vision, audio, touch, taste, and smell), AR also aims to deliver virtual contents which human senses can perceive.

Vision

Vision is the most common human sense to be augmented. This can be accomplished by superimposing or rendering a 3D model for a user to view.

In addition to Pokémon Go mobile game, another example of AR use cases for augmenting vision is AR business card created by Oscar Falmer, an iOS developer. By scanning Falmer's business card, the users can view virtual contents including Falmer's headshot, name, hyperlinks to Falmer's website and Twitter, social media web page and others rendered on the users' mobile device - see Figure 4. AR also makes headway in assisting medical professionals by allowing 3D models to be constructed from a patient's MRI and CT scan images and then superimposed onto the patient as a point of reference for surgeons during a surgery [9].

Figure 4. Augmenting Vision - AR Business Card [12]

Hearing

Like a 3D model or a virtual 2D image embedded as shown in Figure 4, a video can also be rendered to include sound that is not emitted by the real environment. The sound can be played on a mobile device's speaker, via a headphone, or via an AR headset.

Touch

Being developed by Northwestern University is a so-called haptic skin. It is a thin layer that is consists of actuators and is in direct physical contact with human skin to deliver patterns of force and augment the sense of touch[13]. Once the technology matures, a potential use of such haptic skin is a full-body suit to aid the physical therapy process for patients such as stroke survivors

[12] https://www.springwise.com/developer-uses-ar-tool-to-create-virtual-business-card/
[13] https://www.youtube.com/watch?v=OYVU9Ha_rHs

who are trying to regain their motor skills by trying to lift up a virtual box superimposed by an AR app.

Taste and Smell

These two senses are likely to be the hardest to augment. There has been some research in building a taste simulator which uses varied electric currents to simulate different taste buds in human tongue. Since different taste buds are responsible for detecting different tastes, simulating targeted taste buds can deliver a sense of taste. Currently, taste simulators can augment primitive tastes like sweet, sour, salty, and bitter by delivering electric currents through the tongue interface as shown in Figure 5 [10].

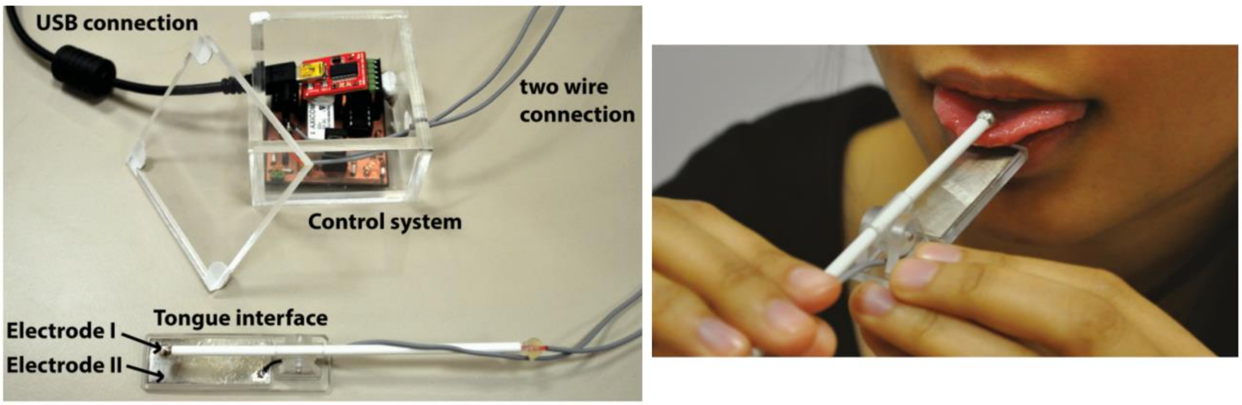


Figure 5. Augmenting Taste - Digital Lollipop [10]

AR by Common Tracking Techniques

A word that is often used when describing an AR experience is *superimposition*. When using a webAR app, users see the real world through the device's camera with the virtual objects such as 3D models superimposed or rendered alongside physical objects, effectively augmenting the real-world objects.

For an AR app to know which physical objects to augment and where to render virtual contents, AR app relies on different tracking methods such as marker-based tracking, image-based tracking, face tracking, location tracking, surface tracking, and spatial tracking.

Marker-based Tracking: Fiducial Marker and Image Marker

A fiducial marker is a 2D, asymmetrical symbol that an AR app detects and tracks in order to understand the orientation of the device running the AR app. Based on that understanding, the AR app superimposes virtual contents onto the marker [1].

An example of *fiducial marker tracking* is shown in Figure 6. When scanning the marker on the left, users will be taken to a WebAR app where a 3D cube with an interior structure is superimposed onto the marker through the device's camera view. This type of marker is called a Hiro marker.

The way AR apps recognize a marker is using computer vision technology. Therefore, depending on the robustness of the computer vision technology implemented, specific AR frameworks/ toolkits might have specific marker requirements such as marker color and size to enhance marker recognition and tracking. For example, fiducial markers with black and white color offer high contrast and, therefore, are recommended [11].

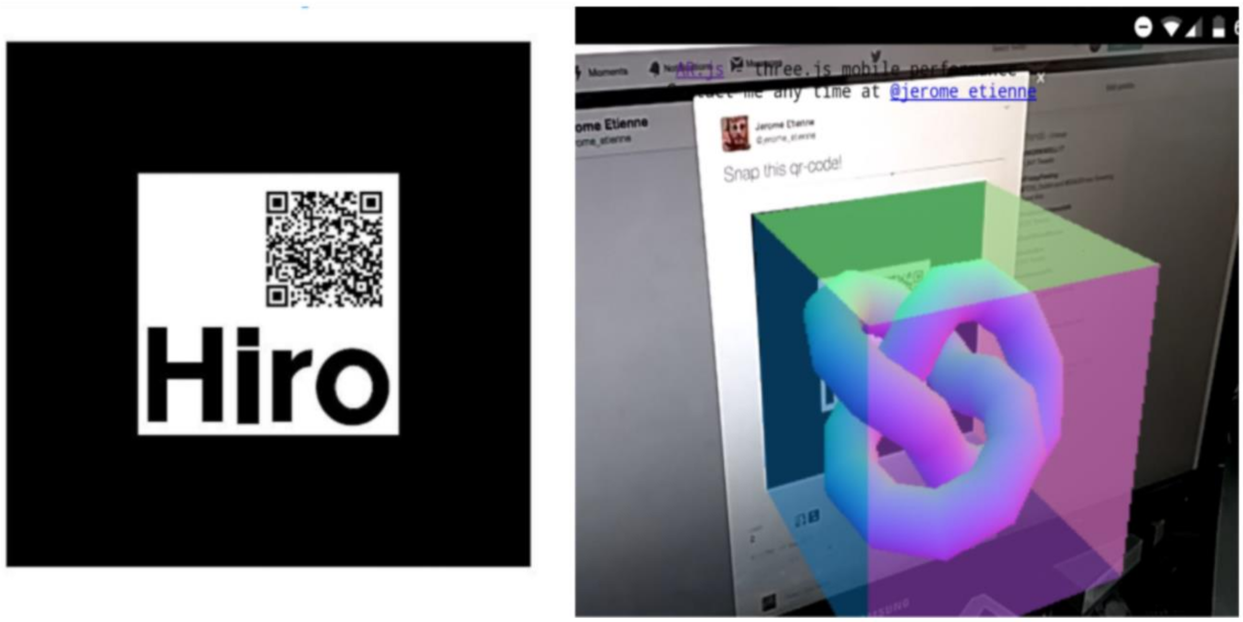


Figure 6. Marker-based Tracking: Fiducial Marker – 2D Hiro Marker [14]

Image-based tracking is a variant of marker-based tracking where an arbitrary, 2D image can be used instead of 2D fiducial markers with geometric figures.

Like markers, images used for AR tracking also need to meet certain requirements. For example, AR.js, a webAR development tool that supports image-based tracking, suggests the use of images with DPI (dots per inch) of 300 or higher. Images with lower DPI can still be recognized but only if the users stand still and close the image[15].

Figure 7 is an example of image-based tracking WebAR app that scans the 2D label of a Heinz ketchup bottle and superimposes a 3D recipe book onto the label. Users can click on the virtual Back/ Next buttons to traverse through the recipe book.

[14] https://medium.com/arjs/ar-code-a-fast-path-to-augmented-reality-60e51be3cbdf
[15] https://ar-js-org.github.io/AR.js-Docs/image-tracking/

Figure 7. Marker-based Tracking: Image Marker – 2D Ketchup Bottle Label [16]

A more robust tracking engine can also detect a 2D image marker as viewed from an occluded angle instead of front-and-center. On the left of Figure 8 is the image marker and on the right is the image marker is viewed from an occluded angle as it is wrapped around a bottle. The tracking engine is robust enough to detect the marker and render a 3D soccer ball.

Figure 8. Marker-based Tracking: Image Tracker - 2D Soda Bottle Label Viewed from Occluded Angle [17]

Object Tracking

For fiducial and image tracking, a 2D image marker needs to be detected in video frames to have the virtual contents rendered. For object tracking, a 3D object as part of the real environment can be detected in video frames.

Figure 9 is an example of object tracking where the AR app can detect a 3D smoke detector in the real environment and render the name, status, and last maintenance date in virtual text next to the smoke detector.

16 https://www.blippar.com/ar-studio
17 https://library.vuforia.com/cylinder-targets/recommendations-designing-cylinder-targets

Figure 9. Object Tracking – 3D Real-life Smoke Detector [12]

Face Tracking

Face tracking is where an AR app tracks a landmark(s) of a human face such as a nose bridge and a forehead and renders, for example, a virtual pair of glasses and a virtual hat on them as shown in Figure 10.

Figure 10. Face Tracking – Snapchat-like Filters[18]

Location Tracking

Location tracking refers to the rendering of AR contents as anchored to an indoors or outdoors location. Figure 11 demonstrates a native AR app that is when pointed at a shop located with a specific latitude and longitude, the app renders a 3D model of the shop's signage [13].

[18] https://github.com/hiukim/mind-ar-js

While marker-based tracking requires the presence of a fiducial marker or image marker to trigger the AR experience, location tracking requires users to be physically near to the location where the AR experience is “attached” to.

Figure 11. Location-Based Tracking – 3D Shop Signage in Advertising [13]

Surface Tracking

“Surface tracking” is sometimes referred to as “marker-less tracking” or “world tracking”.

Figure 12 illustrates how IKEA Place app works using surface tracking:

- The IKEA Place app attempts to detect a planar surface (i.e., the ground in this example) on which it can render virtual contents.
- On the device screen when the user clicks on the ground, the app renders a virtual armchair as attached to the location inside the room marked by the user’s click.
- Now that the virtual armchair is rendered at the user’s defined location, user can walk around to view the armchair from different angles and see if the armchair fits the room before making a purchase.

Figure 12. Surface Tracking – Ikea Place[19]

Spatial Tracking

In spatial tracking, a 3D map of a physical space is created. Users then add virtual contents to different locations inside the physical space by referencing the 3D map instead of having to specify the coordinates of the location [14]. Furthermore, there is no need for any type of physical markers to trigger spatial AR experience.

Figure 13 is an example of spatial tracking. As the user device scans the physical space, a virtual dashboard is displayed on the user device screen to give updates on the maintenance records of different machines. For visual purposes, the dashboard appears to be “floating” in the physical space, while in fact, it can only be viewed from users’ AR device.

[19] https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050

Figure 13. Spatial Tracking: Machine Dashboards Anchored to 3D Mesh of Physical Environment [20]

2.3 RQ#2: How Does AR Work?

In this section, we further discuss key dimensions of the mechanism underlying AR, which is the same for all AR platforms: webAR, native AR, and AR headset:

1. AR content creation
2. Marker detection and tracking
3. AR content rendering
4. User interactivity

AR Content Creation

For augmenting vision, *static AR contents* such as 3D graphics and animations, videos, 2D graphics, and hyperlinks are typically used. 3D graphics and animations can be created using a variety of 3D modeling tools such as Autodesk Maya[21], Blender[22], and SketchUp[23]. An alternative is to create 3D graphics by scanning physical objects using 3D scanners. Another alternative is to purchase pre-made, paid/ free 3D graphics from 3D model marketplaces such as SketchFab[24].

Depending on specific use cases, *dynamic AR contents* might be beneficial. For example, maintenance workers who rely on AR to inspect and perform maintenance tasks would need to view real-time Internet-of-things (IoT) sensor readings for various equipment [15].

[20] https://www.youtube.com/watch?v=lomyM4loAVg
[21] https://www.autodesk.com/products/maya/overview
[22] https://www.blender.org/
[23] https://www.sketchup.com/
[24] https://sketchfab.com/feed

Marker Detection and Tracking

As a mobile device captures live camera feeds in the forms of camera frames, an AR app will attempt to detect marker(s) in those camera frames. Once the marker is detected, the AR app continues the track the marker(s) so that virtual objects can be accurately superimposed onto the marker(s). Even for the case of location tracking and marker-less tracking where there seems to be an absence of markers, the AR app still attempts to detect a planar surface where AR contents can be superimposed upon.

To make the AR contents look believable, AR apps attempt to calculate the *camera pose estimation* in *six degrees-of-freedom (6DOF).*

A related concept is three degrees-of-freedom (3DOF) which refers to the *location* where virtual objects should be superimposed while 6DOF refers to both *location* and *orientation*, collectively referred to as *position* [1]. After a marker has been detected in a video frame, the position of the marker is tracked in subsequent frames. AR apps use the calculated camera pose estimation which are mathematical operations to transform and render virtual objects so that they look as believably as possible [16].

In simple terms, when a shopper uses the IKEA Place app to virtually place an armchair in the corner of their living room, 3DOF only guides the rendering of the virtual chair in the corner of the room while 6DOF guides the rendering of the virtual chair in the corner of the living and as seen from a specific angle. That way, when the shopper walks around the spot where the virtual chair sits, the virtual chair 3D model will be re-rendered for every video frame depending on the angle from which the shopper is looking at.

When implementing an AR app, developers have an option to either:

1. Implements custom tracking algorithms which calculate 6DOF and instruct the rendering engine to render the virtual objects accordingly as implemented in papers [17] [18].
2. Uses an open-source or commercial tracking engines which have built-in tracking capabilities – see section Tracking Engines for more details.
3. Uses a combination of custom tracking algorithms and open-source/ commercial tracking engines with the custom tracking algorithms designed to address the limitations of open-source/ commercial tracking engines as implemented in papers [19] [20].

AR Content Rendering: OpenGL

“OpenGL® is the most widely adopted 2D and 3D graphics API in the industry” [21]. Many other rendering engines are built on top of OpenGL including WebGL[25] for webAR and Sceneform[26] for native AR for Android – see section Rendering Engines for more details.

[25] https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API
[26] https://github.com/SceneView/sceneform-android

User Interactivity: Gesture, Speech, Gaze Controls and Remote Assistance

Using speech, gesture, and gaze to interact with AR contents can provide a richer, more natural, and intuitive user experience. For example, paper [22] implements hand gesture and voice command to scale and manipulate 3D models on a HoloLens. Paper [23] implements gaze interaction where users who wear a head-mounted display can look at a physical object to trigger the display of virtual text about the physical object and hold their gaze to select the “Yes” or “No” virtual button – see Figure 14.

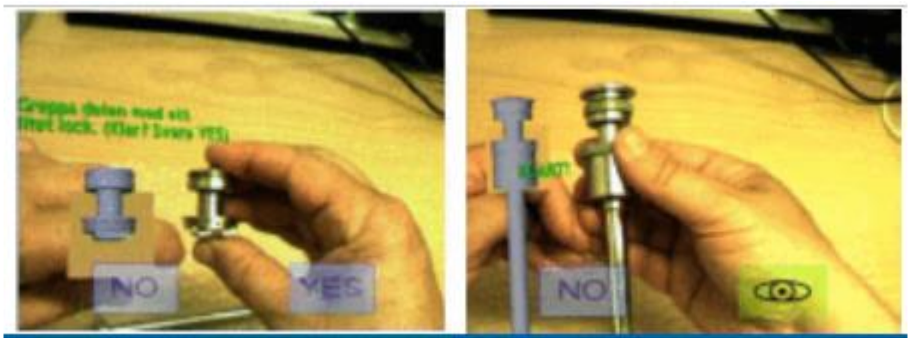


Figure 14. User Interactivity: Gazing [23]

Remote collaboration is another type of user interactivity that involves at least two users while speech, gesture, and gaze-type of interaction only requires one user.

For example, paper [24] illustrates a scenario where an operator (on the left of Figure 15) is pointing his/ her device at a machine that he/ she is about to perform a maintenance service on. The maintenance expert (on the right) views the same live camera feed from the local operator’s device and makes annotations in green and red color on the expert’s computer. The annotations are then rendered on top of the machine as viewed from the local operator’s device. This real-time, remote assistance provides local operators with expert advice and speeds up the maintenance process.

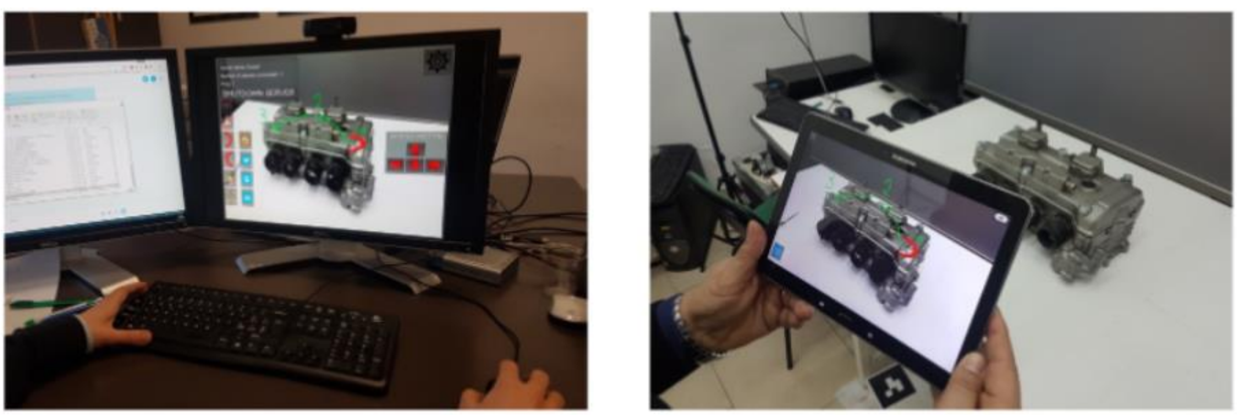


Figure 15. User Interactivity: Remote Assistance [24]

2.4 RQ#3: What Factors Influence AR Adoption?

Because the research aims to produce an industry-agnostic AR implementation methodology, this section synthesizes existing research performed for various industries on the factors which influence AR adoption – see Table 4.

While there is a large number of papers on AR adoption, most of them test for AR adoption using traditional factors (e.g., “perceived ease of use”) from well-documented technology acceptance models such as TAM[27] [25] [26]. For this research, only papers presenting new, AR-specific factors for AR adoption are reviewed.

Seven papers [27] [28] [29] [30] [31] [32] [33] from the search query results were reviewed in detail.

- Paper [27] investigates AR adoption factors for the construction sector through a prototype implementation and evaluation.
- Paper [28] [30] [31] investigates AR adoption factors for manufacturing sector by conducting a literature review.
- Paper [29] investigates general, industry-agnostic AR adoption factors by conducting a literature review.
- Paper [32] investigates AR adoption factors for the maintenance sector by conducting a literature review in conjunction with a prototype implementation and evaluation.
- Paper [33] investigates AR adoption factors for the education section by conducting a survey with teachers have had direct or indirect experience with the use of AR in teaching at their school. No AR prototype was implemented.

In Table 4, the factors extracted from these papers are organized into three categories “organization”, “technology”, and “environment” based on the Technology-Organization-Environment (TOE) framework [34]. The factors are also organized by sub-categories based on the author’s logical reasoning. The factors are mapped to the papers in which the factors were discussed.

[27] https://en.wikipedia.org/wiki/Technology_acceptance_model

Category	Sub-category	Factor (F)	Description	[27]	[28]	[29]	[30]	[31]	[32]	[33]
Organization	Company-level	F1. Task identification	• Determine how AR can help enhance *worker performance* for a specific task e.g., use AR to view traditionally paper-based inspection manuals to speed up inspection processes. • Additionally, the *working conditions* where a task takes place should also be considered to assess *AR suitability*. Examples of working conditions include stable/ unable internet connection, lighting, noise, weather conditions.	x	x	x	x		x	
		F2. Health and safety concerns	• The use of AR should not cause *distractions*, resulting in users not paying attention to real dangers in the physical environment and possibly getting injured. The use of AR might also lead to *visual fatigue*.	x	x	x	x	x		
		F3. Implementation cost	• *Cost* of designing, developing, deploying, and maintaining AR apps.				x	x		
		F4. Change management	• Provide *training and additional support* to enable users to effectively use AR apps for their work.				x	x	x	
	User-level	F5. User experience during AR usage	• The use of AR should not complete replace *face-to-face interactions.* • The *user interface* should render *information,* which is accurate, timely, legible, and beneficial to assisting users in performing their tasks and not overload their mental capacity. • User *interactivity* should be natural and appropriate for the working conditions. E.g., users wearing gloves might not be able to perform a lot of clicks on AR display screen. *Hand gestures* and *voice commands* might improve interactivity. • The user interface should be *designed in close collaboration with users.*	x	x	x	x		x	
		F6. Privacy and data protection	• Privacy and data protection for users should be considered. For example, users might be concerned that AR devices are used to *surveil* and track their movements through a shop floor in industrial AR use cases. • AR systems which handle *sensitive information like medical records* should be handled with care. • The use of *human face* as a marker is controversial and gives a surveillance sentiment • *Data policies* regarding 1) what data is being logged and for what purposes 2) how data should be stored and for how long 3) who should have access to what data and for what purpose should be defined.		x	x	x	x	x	
		F7. User's technology quotient	• The more *technology savvy* users are, the more likely they are to accept AR. Emphasize the needs for providing the workforce with technology professional development.							x
		R8. Psychological risks	• *Change management* should address user's psychological risks such as fear of job loss, fear of over-reliance on technology and making wrong decisions, fear of perceived loss competence because workers are guided through their tasks with the help with AR.	x	x					
Technology	Hardware	F9. Essential hardware requirements	• Battery lifetime, camera and display resolution, storage capacity, compliance with safety standards, possibility to implement interfaces to enterprise system and operating systems, processing speed, sensor accuracy			x			x	
		F10. Ergonomics	• Refers to whether AR hardware imposes restrictions on movement, provides limited/ wide field of view, weight, portability, suitability for outdoor use and specific working conditions.	x		x	x		x	
	Software	F11. Essential software requirements	• The AR engine should be robust enough to delivery an acceptable level of latency while ensuring accurate *tracking* of markers and *rendering* on AR contents on detected markers •			x	x	x		
		F12. Integration with enterprise systems for data exchange purposes	• AR software should be able to *integrate with enterprise systems* for data retrieval (Data augmented in 3D in user's field of view) and/ or data manipulation (User can modify data while using AR systems. Modified data will be persisted to a database). • Measures should be designed to prevent *accidental or unauthorized data updates* via the use of AR systems.		x		x		x	
		F13. Easy authoring system	• Establish a *content creation pipeline* where contents can be created by individuals with limited to no technical expertise and later be integrated into AR solutions.					x		
Environment	Regulations	F14. Compliance with employee protection regulations	• Comply with formal health and safety concerns (see *F2*), privacy and data protection (see *F6*) *as mandated by the laws*.					x		
	Third party	F15. Integration with third party systems	• In addition to *F12* (integration with internal systems), AR systems might need to integrate with *third-party systems.*					x		
		F16. External support	• Rely on *external technology partners* instead of in-house talents to guide the implementation of AR.					x		
		F17. Industry-wide standardization of AR solutions	• Keep a close pulse on *tried-and-tested AR use cases in your industry* such as the common use of AR in order picking for warehouse operations.					x		

Table 4. Factors Influencing AR Adoption

2.5 RQ#4: From a Technical Standpoint, What Is the Difference Between WebAR v. Native AR v. AR-on-Headset Application Development?

In this section, existing literature on webAR, native AR and AR-on-headset use cases was reviewed to compare how technical implementations differ for webAR, native AR, and AR-on-headset applications.

WebAR

In essence, there are two primary technologies that support WebAR:

1. *Tracking engine*:
    a. This engine allows a webAR app to track the location and orientation of a physical object or environment and then renders virtual objects (i.e., AR contents) as anchored the physical object or environment
2. *Rendering engine*:
    a. This engine allows for the rendering of AR contents such as 3D models in a browser.

Note that the tracking engine and rendering engine which are used to implement an AR application must interface with each other.

In the following section, we will go into more details of different 3D rendering engines and AR tracking engines for webAR.

Rendering Engines

WebGL (Open-source)
WebGL[28] is a low-level API built on top of OpenGL and allows browsers to render 2D and 3D graphics. The word “low-level” here refers to the fact that WebGL only draws points, lines, and triangles in a browser[29].

Therefore, writing code using WebGL API to render full-fledged AR contents of complex geometries is highly time-consuming. As a result, there have been emerging open-source libraries such as *three.js* and *A-Frame* built on top of WebGL to simplify the task of rendering AR contents for webAR apps.

Three.js (Open-source)
Three.js is a JavaScript library and an abstraction layer on top of WebGL. In other words, Three.js uses WebGL to draw 2D and 3D models. Using Three.js, developers can construct 2D/ 3D graphics with fewer lines of code compared if they were to do it with WebGL.
A-Frame (Open-source)

[28] https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL
[29] https://threejs.org/manual/#en/fundamentals

A-Frame is a JavaScript library that functions as a web component wrapper for Three.js, allowing developers to create AR scenes by using custom HTML elements created by A-Frame such as <a-scene>, <a-box>, and so on in an HTML document.

Figure 16 illustrates an example .html which uses <a-scene>, <a-box>, <a-sphere>, <a-cylinder>, <a-plane>, <a-sky> to create an AR scene with a box, a sphere, a cylinder, a plane, and a sky. When a web browser renders this .html, the result is a 3D scene as shown in Figure 17.

```html
<html>
  <head>
    <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
  </head>
  <body>
    <!-- Creating an a-scene to encapsulate 3D models created using a-frame components -->
    <a-scene>
      <!-- Rendering 3D shapes for a box, a sphere, a cylinder, a plane, and a sky -->
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```

Figure 16. A-Frame Example HTML[30]

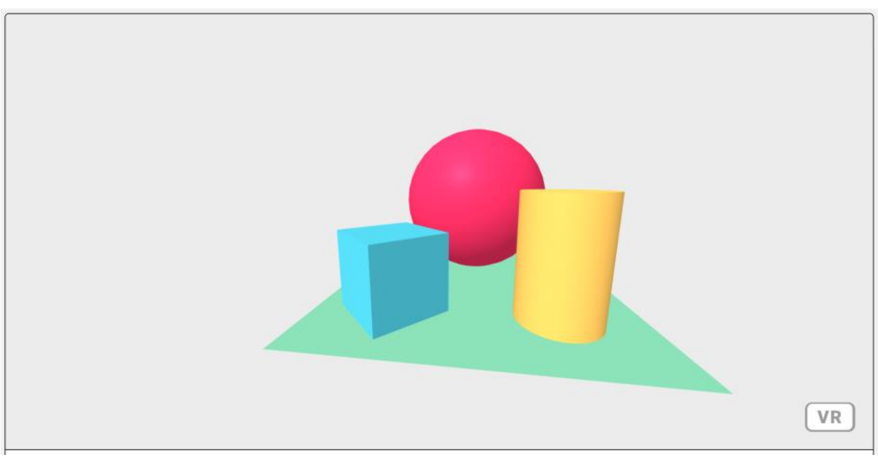


Figure 17. A-Frame Example 3D Rendering[31]

Tracking Engines

AR.js (Open-source): Fiducial Marker, Image Marker and Location Tracking

[30] https://aframe.io/docs/1.3.0/introduction/
[31] https://aframe.io/docs/1.3.0/introduction/

AR.js[32] is an open-source JavaScript library that supports marker-based, image-based, and location tracking. In the case of location tracking, AR.js attempts to detect a planar surface on which AR content can be superimposed.

Figure 18 is the source code of an example fiducial marker-based webAR application built using *A-frame* as the 3D rendering engine and *AR.js* as the AR tracking engine. As shown in Figure 19, a yellow 3D cube is rendered when the AR tracking engine detects a marker printed on a piece of paper with the word 'Hiro' on it.

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Loading A-Frame library for modeling AR content -->
        <script src="https://aframe.io/releases/1.0.0/aframe.min.js"></script>
        <!-- Loading AR.js library for marker detection and tracking -->
        <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>
    </head>
    <body style="margin : 0px; overflow: hidden;">
        <a-scene embedded arjs>
            <!-- Defining marker type: Hiro marker -->
            <a-marker preset="hiro">
                <!-- Superimposing a yellow 3D box on the detected marker -->
                <a-box
                    position="0 0.8 0"
                    material="color: yellow;"
                    depth="0.5"
                    height="0.5"
                    width="0.5">
                </a-box>
            </a-marker>
            <a-entity camera></a-entity>
        </a-scene>
    </body>
</html>
```

Figure 18. AR.js Example WebAR Application Source Code (Code created by author)

32 https://github.com/AR-js-org/AR.js

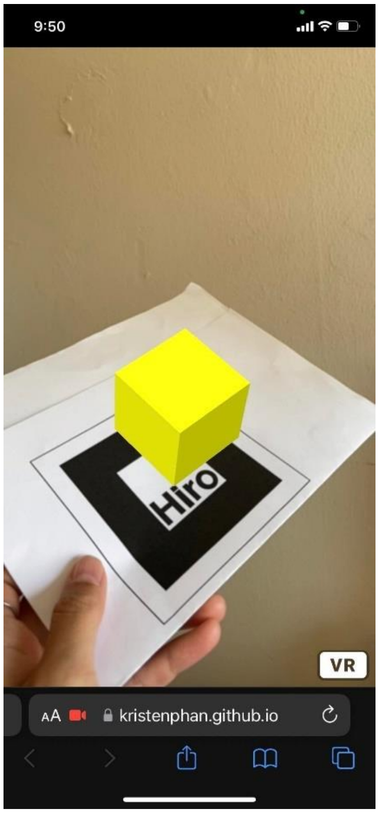


Figure 19. AR.js Example WebAR Application Screenshot (Screenshot taken on author's mobile device)

MindAR.js (Open-source): Image Marker and Face Tracking

MindAR.js[33] is an open-source JavaScript library that supports image-based and face tracking. Although MindAR.js and AR.js both support image tracking, MindAR.js offers more robust image tracking with the ability to track multiple images simultaneously.

Figure 20 is the source code of an example image marker-based webAR application built using *three.js* as the 3D rendering engine and *MindAR.js* as the AR tracking engine. While Figure 19

33 https://hiukim.github.io/mind-ar-js-doc/

shows a 3D cube rendered on top of a fiducial marker, Figure 21 shows a 3D cube rendered on top of an image marker.

```html
<!-- index.html -->
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Import MindAR tracking engine -->
    <script src='./libs/mindar-image-three.prod.js'></script>
    <script src="./main.js"></script>
    <style>
      html, body {
        position: relative;
        margin: 0;
        width: 100%;
        height: 100%;
        overflow: hidden;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

```javascript
// main.js

//  Import three.js 3D rendering engine
const THREE = window.MINDAR.IMAGE.THREE;

document.addEventListener('DOMContentLoaded', () => {
  const start = async() => {
    // Instantiate a MindARThree object using MindAR engine
    const mindarThree = new window.MINDAR.IMAGE.MindARThree({
      container: document.body,
      imageTargetSrc: './assets/target.mind'
    });
    const {renderer, scene, camera} = mindarThree;

    // Create a 3D yellow cube
    const geometry = new THREE.BoxGeometry(0.5, 0.5, 0.5);
    const material = new THREE.MeshBasicMaterial({color: 0xFFFF00});
    const cube = new THREE.Mesh(geometry, material);
    cube.position.set(0, 0, 0.2);

    // Create a MindArThree image anchor
    const anchor = mindarThree.addAnchor(0);
    // Attach the cube to the image anchor
    anchor.group.add(cube);

    // Start the MindARThree engine
    await mindarThree.start();
    renderer.setAnimationLoop(() => {
      renderer.render(scene, camera);
    });
  };
  start();
});
```

Figure 20. MindAR Example WebAR Application Source Code (Code created by author)

Figure 21. MindAR Example WebAR Application Screenshot (Screenshot taken on author's mobile device)

WebXR (Open-source): World Tracking

WebXR[34] is an emerging web API standard which standardizes the way web browsers access *VR/ AR hardware* (e.g., mobile phones, AR dedicated headsets such as HoloLens 1) including the *sensors* (e.g., mobile phone camera, HoloLens 1 hand controllers). Thanks to this access, web browsers can understand the *physical environment* and render *VR/ AR contents* correctly in the *web VR/AR applications*.

Figure 22 is the source code of an example webAR application built using *three.js* as the 3D rendering engine and *WebXR* as the AR tracking engine. As shown in Figure 23, when a user taps on the device screen, at the click point position the webAR application renders a 3D yellow cube, like Figure 19, but without the need for a fiducial marker.

```html
<!-- index.html -->
<html lang="en">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script type="module" src="./main.js"></script>
  <style>
    body {margin: 0}
  </style>
</head>
<body>
</body>
</html>
```

```javascript
// main.js

// Import three.js AR rendering engine
import * as THREE from '../../libs/three.js-r132/build/three.module.js';
// Import three.js ARButton which is not part of three.js core api
import {ARButton} from '../../libs/three.js-r132/examples/jsm/webxr/ARButton.js';

// Load .js code after html doc has loaded
document.addEventListener('DOMContentLoaded', () => {
  const initialize = async() => {
    // Instantiate three.js scene, camera, renderer
    const scene = new THREE.Scene();
    const camera = new THREE.PerspectiveCamera();
    const renderer = new THREE.WebGLRenderer({antialias: true, alpha: true});
    // Set renderer as full-screen and attach renderer to document body
    renderer.setSize(window.innerWidth, window.innerHeight);
    renderer.setPixelRatio(window.devicePixelRatio);
    document.body.appendChild(renderer.domElement);

    // Create a 3d cube
    const geometry = new THREE.BoxGeometry(0.06, 0.06, 0.06);
    const material = new THREE.MeshBasicMaterial({color: 0xFFFF00});
    const cube = new THREE.Mesh(geometry, material);
    // Position the cube and add it to scene
    cube.position.set(0, 0, -0.3);
    scene.add(cube);
    // Add light to scene so we can see the color on the cube
    const light = new THREE.HemisphereLight( 0xffffff, 0xbbbbff, 1 );
    scene.add(light);

    // Enable WebXR
    renderer.xr.enabled = true;
    // Start rendering loop
    renderer.setAnimationLoop(() => {
      renderer.render(scene, camera);
    });

    // Create a variable for WebXR current session
    let currentSession = null;

    // Create a WebXR button from three.js to start/ end a WebXR session when arButton is clicked
    const arButton = ARButton.createButton(renderer, {optionFeatures: ['dom-overlay'], domOverlay: {root: document.body}});
    // Attach renderer and ARButton to document body
    document.body.appendChild(arButton);
    document.body.appendChild(renderer.domElement);
  }
  initialize();
});
```

Figure 22. WebXR Example WebAR Application Source Code (Code created by author)

---

[34] https://www.w3.org/TR/webxr/

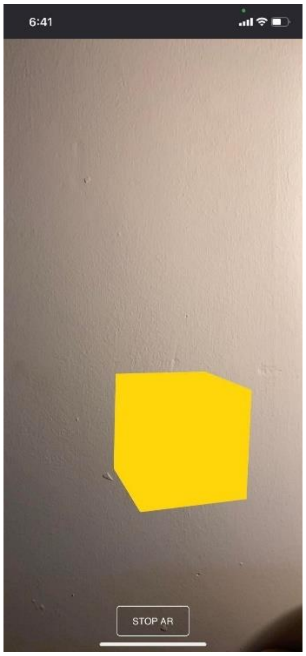


Figure 23. WebXR Example WebAR Application Screenshot (Screenshot taken on author's mobile device)

Commercial Tracking Engines

In addition to the open-source AR tracking engines discussed above, there are several commercial tracking engines such as *8th Wall*[35], *ZapWorks*[36], and *Blippar*[37]. They implement their own computer vision for tracking capabilities and often integrate with major 3D rendering engines such as *three.js* and *A-frame*.

Native AR

For *rendering engine*, Unity[38] appears to be the most popular choice as implemented in 17 out of 24 papers which were examined as part of the literature review and feature a native AR mobile application [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51]. The second most-popular rendering engine is OpenGL[39] as implemented in 3 out of 24 papers examined [52] [53] [54].

While Unity and OpenGL support multiple mobile operating systems (Android, iOS, Windows), rendering engine SceneKit[40] only supports iOS as implemented in paper [55] [56] and rendering engine Sceneform[41] (built on top of OpenGL) only supports Android as implemented in paper [57].

35 https://www.8thwall.com/
36 https://zap.works/
37 https://www.blippar.com/webar-sdk
38 https://unity.com/
39 https://www.opengl.org/
40 https://developer.apple.com/documentation/scenekit/
41 https://github.com/SceneView/sceneform-android

For *tracking engine*, Vuforia[42] appears to the most popular choice as implemented in 12 out of the 24 papers which were examined as part of the literature review and feature a native AR mobile application [35] [36] [38] [39] [40] [41] [42] [43] [44] [45] [47] [46].

AR-on-Headset

For *rendering engine*, 4 out of 12 papers which were examined as part of the literature review and feature an AR application on AR headset use Unity rendering engine [58] [59] [45] [60], 1 paper uses OpenGL [61], and the remaining 7 papers do not specify the implemented rendering engine.

For *tracking engine*, 6 out of those 12 paper use ARToolkit[43] (no longer available) tracking engine [61] [62] [63] [64] [65] [66], 2 papers use Vuforia tracking engine [45] [60], 2 papers use Mixed Reality Toolkit tracking engine [58] [67], 1 paper uses Lumin SDK [59], and 1 paper implements its own tracking algorithm [17].

While Vuforia supports different AR headsets (HoloLens, Magic Leap, Vuzix, Realwear)[44], Mixed Reality Toolkit supports only HoloLens[45] and Lumin SDK supports only Magic Leap[46]. And because Vuforia integrates with Unity, Unity also supports all AR headsets which Vuforia supports.

WebAR vs. Native AR vs. AR Headset

Based on the findings of this section, it becomes clear that the mechanism underlying AR is the same for webAR, native AR, and AR applications on AR headsets: AR requires the use of 1) a tracking engine to track a physical object/ environment and 2) a rendering engine to anchor a virtual object to that physical object/ environment.

Vuforia rendering engine stand out for their cross-platform support (Android, iOS, HoloLens, Magic Leap, Vuzix, Realwear). However, Vuforia still do not support webAR as of Vuforia engine version v10.10.2[47].

Worth noting is that each paper in existing literature primarily focuses on a specific use case and implements a prototype using a specific rendering and tracking engine. To the author's best knowledge, there is, however, no overarching guidance that facilitates the design phase for how to choose between different AR platforms: webAR, native AR, and AR headset. AR platforms is a means to an end, not the end objective, and should be chosen based on its suitability for a specific use case.

---

[42] https://developer.vuforia.com/downloads/SDK
[43] https://en.wikipedia.org/wiki/ARToolKit
[44] https://library.vuforia.com/platform-support/recommended-devices
[45] https://condescending-wilson-d1a22a.netlify.app/documentation.ja/releasenotes
[46] https://ml1-developer.magicleap.com/en-us/learn/guides/lumin-sdk-0-24-1-release-notes
[47] https://library.vuforia.com/release-notes/vuforia-engine-release-notes

Additionally, since the use cases in existing literature are primarily small-scale proof-of-concept, they do not provide technical guidance for the development and deployment of AR applications so that the AR applications can be developed efficiently and deployed on scale, securely, and cost-effectively.

2.7 RQ#5: From a Technical Standpoint, What Is the Difference Between WebAR vs. Regular, Non-AR Web Application Development?

Regular, Non-AR Web Application

To identify the differences between webAR and non-AR web application development, a search query is conducted to explore different tools and methods used for the non-AR web application development.

Figure 24 is a reference architecture of a full-stack web application. Worth paying attention to are components labeled 1 (front-end), 2 (cloud storage), 3 (back-end), and 4 (database).

The front-end comprises of a web server serving HTML, CSS, and JavaScript files to a user browser. The JavaScript code is executed by the web browser and can pull additional data from a cloud storage such as an image to be included in a web page. The browser renders the HTML, CSS, and JavaScript files into a complete web page [68].

The back-end might comprise of multiple backend services running on different servers with each service performing a specific function. This is called a service-oriented architecture (SOA) [68, 69].

For example, on https://www.bol.com/ the e-commerce, as a shopping is completing a purchase, the front-end communicates with different back-end services. For example there might be a back-end service to process payments and another back-end service to send a confirmation email to a user after the payment has been processed and the purchase has been finalized. The back-end services then return output to the front-end so that users can view the payment status in the web browser.

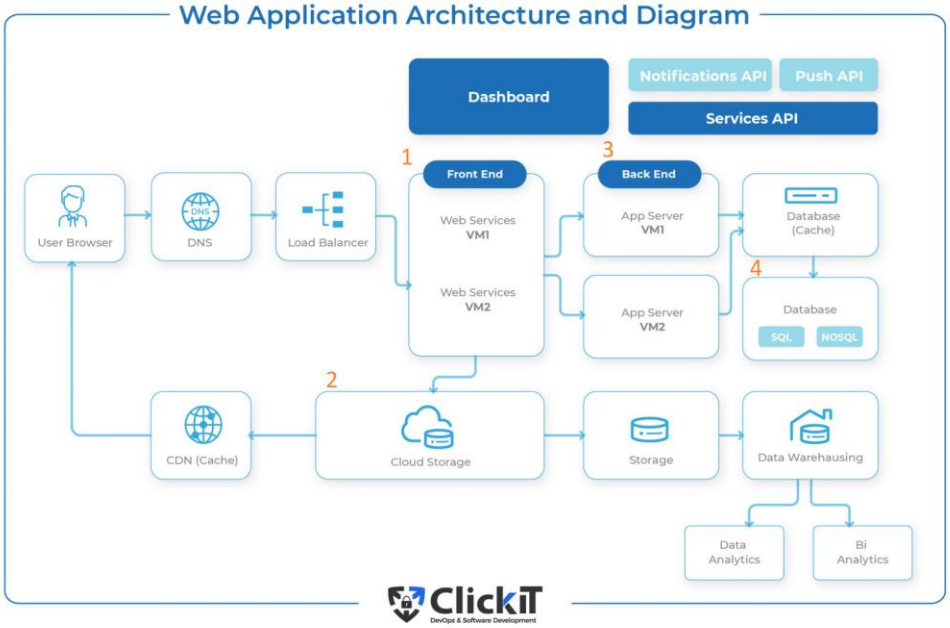


Figure 24. Non-AR Web Application Architecture [68]

WebAR vs. Regular, Non-AR Web Application

This SOA approach applies for webAR as well. The example webAR applications discussed in section WebAR only consist of a front-end since all both the marker detection/ tracking and 3D rendering operations can be executed by the browser. However, developers can also decide to offload more computation-intensive tasks to a backend service(s) rather than executing those tasks in a browser which has access to limited computation resources provided by the user's mobile phone/ tablet.

The means if developers want to implement, say, gesture control as a form of user interactivity for a webAR application, developers can choose to implement a back-end service to execute the gesture control feature, a potentially computation-intensive task which involves the use of computer vision machine learning algorithms to recognize hand gestures and execute pre-defined commands.

In summary, SOA as a design principle is applicable for both webAR and non-AR web applications. The design decisions on how to split up a webAR/ non-AR web application into smaller services depend on the application's specific functionalities.

2.8 RQ#6: What Is Cloud Serverless Infrastructure?

Cloud Service Types

Before discussing the definition of serverless infrastructure, it is helpful to differentiate different types of cloud services by examining the technology stack a software application runs on – see Figure 25:

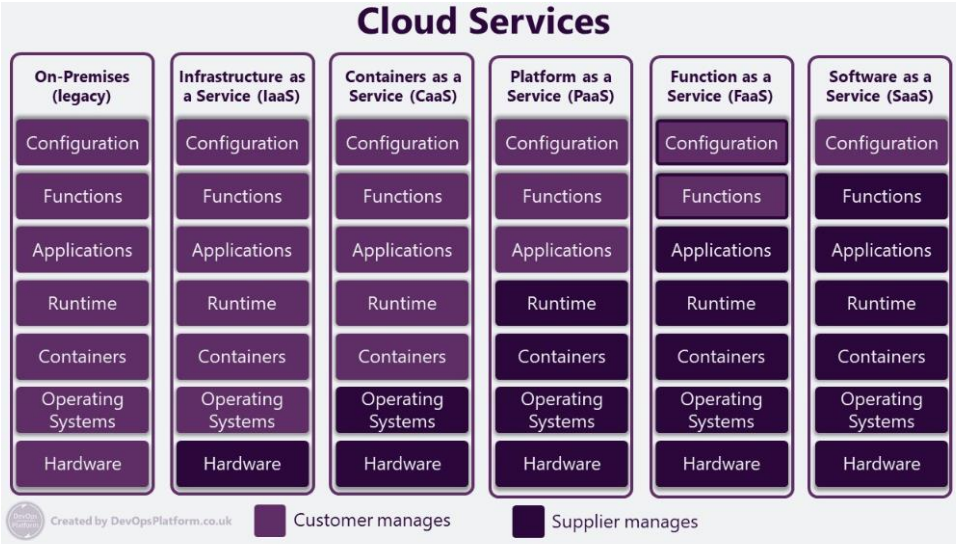


Figure 25. Cloud Service Types [70]

On-premises

Customers manage the full technology stack in-house. Not including in Figure 25 is networking, which enables the connectivity between different cloud resources in a cloud network [71].

Infrastructure-as-a-Service (IaaS)

Customers can rent virtual machines (hardware level) provided by public cloud providers. On a single physical server, a cloud provider can install multiple virtual machines (i.e., virtual servers), each with its own operating system, and rent them out. This allows a cloud provider to host multiple tenants on their physical servers. On these virtual machines, customers can install an operating system of choice and run any application on them [72].

For examples, customers can rent a virtual machine and install MySQL, a relational database management system, if they want to host their own MySQL database on the virtual machine.

Container-as-a-Service (CaaS)

“Containers encapsulate an application as a single executable package of software that bundles application code together with all of the related configuration files, libraries, and dependencies required for it to run” [73].

Because of the way the code and dependencies are packaged, containers can run uniformly and consistently on any virtual machine or physical server if the machine provides the appropriate container engine such as Docker[48]. This level of abstraction from the operating system makes CaaS a more flexible and secure way to deploy an application than deploying the application directly on a virtual machine as in IaaS [73].

When instantiating a container, developers need to define a container image which contains the application source code along with all dependencies and which instructs the container engine on how to spin up a new container [74].

When the number of running containers goes up, it gets harder to monitor and manage the containers. Examples of container management tasks include monitoring container health, restarting a crashed container on a different virtual machine, and provisioning new containers as traffic grows. Enter container orchestration services such as Kubernetes[49], Dock Swarm[50], Meso[51], and NoMad[52]. Container orchestration services automate the management of container lifecycle including provisioning, deployment, scaling (up and down), networking, load balancing and more [75].

Platform-as-a-Service (PaaS)

Instead of managing the containers and container runtime yourself as in CaaS, for PaaS customers only need to manage the application source code including functions and the configurations of the application [70]. Note that an application can contain multiple functions – more on "functions" in section Function-as-a-Service (FaaS).

"Runtime" as known as "runtime environment" is a software that is designed to run other software [76]. Each programming language requires its own runtime. While CaaS cloud services support any runtime, PaaS cloud services typically support a more limited selection of runtimes. For example, Google Cloud's Cloud Run[53], a CaaS, supports any runtime while Google Cloud's App Engine[54], a PaaS, supports only Node.js, Java, Ruby, C#, Go, Python, or PHP.

Function-as-a-Service (FaaS)

A functions is a snippet of code which performs a specific action [77]. Think a math function that takes in an input and produces an output. FaaS allows developers to execute snippets of code on demand without having to manage the underlying infrastructure. In the case of AWS Lambda, a FaaS service, Lambda spins up a container every time a user-defined function is triggered, for example, via HTTP, runs the function inside the container, and shuts down the container once the container has finished executing the function [78].

For example, a web shop (e.g., https://www.bol.com/) allows customers to make a payment for their purchases. Using a microservice architecture, the web shop application can be designed

[48] https://www.docker.com/
[49] https://kubernetes.io/
[50] https://docs.docker.com/engine/swarm/
[51] https://mesos.apache.org/documentation/latest/containerizers/
[52] https://www.nomadproject.io/
[53] https://cloud.google.com/run
[54] https://cloud.google.com/appengine

as a composite of multiple services, one of which is the “payment” service. This payment service receives the payment detail including card number (input), sends the payment detail via an API call to an external payment API (e.g., https://www.adyen.com/), receives a confirmation of a successful/ failed payment from the external payment API provider, and returns the confirmation to the web shop (output) so that shoppers can view the status of their payment.

This “payment” service can be implemented with a FaaS cloud service. When there are multiple, simultaneous requests for processing payments, FaaS cloud service will dynamically spin up multiple containers with each container running the payment function and processing one payment request [79].

Software-as-a-Service (SaaS)

SaaS is the highest level of abstraction where customers are only responsible for configuring the application they want to use such as Gmail and Outlook [70].

A special type of SaaS is Database-as-a-Service (DbaaS) where users can use and access a cloud database without having to manage the physical server that database runs on along with other database management tasks such as performing backups [80].

Serverless Cloud Services

Since the term “serverless computing” is coined by the industry, it lacks a consistent, formal definition [81]. Therefore, this section goes straight to the source and discusses the definition of serverless computing as described by the three largest cloud providers (AWS, Google Cloud, and Microsoft Azure) along with their serverless cloud services.

Definition of serverless computing

AWS defines “serverless computing” as “[…] technologies for running code, managing data, and integrating applications, all without managing servers. Serverless technologies feature automatic scaling, built-in high availability, and a pay-for-use billing model to increase agility and optimize costs. These technologies also eliminate infrastructure management tasks like capacity provisioning and patching, so you can focus on writing code that serves your customers” [82].

Google Cloud defines “serverless computing” as “[…] a fully managed […] platform. Our serverless computing automatically scales your services up or down, even to zero, depending on traffic, and you pay only for what you use” [83].

Microsoft Azure defines “serverless computing” as “[…] enables developers to build applications faster by eliminating the need for them to manage infrastructure. With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code” [84].

Based on these three definition, this research defines “serverless computing” as the technologies which allow developers to deploy applications or functions without managing the underlying infrastructure. Thanks to this flexibility, developers can shift their attention from heavy-lifting infrastructure management tasks (e.g., virtual machine provisioning and patching) to what they do best – developing new applications.

The word “serverless” refers to the fact that cloud providers, not developers, are responsible for managing the servers, automatically scaling up/ down the deployed applications and functions depending on traffic for built-in high availability, and offering pay-for-use billing services for cost optimization.

“Serverless computing” is also referred to as “cloud computing” which is different from so-called “private or on-premises computing” where companies must manage their own data centers. It appears that in recent years, the industry prefers the term “serverless computing” over “cloud computing” as cloud computing services mature to the highest level of abstraction where developers do not need to manage any hardware or software server, hence the term “serverless”.

For example, without cloud/ serverless computing services, to build a web application, developers need to:

1. Create the source code of the application
2. Buy a physical server and manage it in their private data center
3. Install and manage a web server (software) on the physical server (hardware) to serve the application content to web browsers

With the earliest form of cloud/ serverless computing services such as virtual machines, developers only need to:

1. Create the source code of the application
2. Rent a virtual machine from a cloud provider. They no longer need to manage the physical server (hardware)
3. Install and manage a web server (software) on the virtual machine

With the latest form of cloud/ serverless computing services such as web hosting software-as-a-service, developers only need to:

1. Create the source code of the application
2. Upload their code to a web hosting service. They no longer need to manage the physical server (hardware) nor web server (software)

Serverless cloud services

To give examples of serverless cloud services as categorized by service types, Table 5 catalogs serverless cloud services provided by Google Cloud. Each cloud service is mapped to:

- Category: Compute (resources which provide processing capabilities), Storage (resources which provide storage capabilities), or Service Integration (resources which integrates different application microservices).
- Service type: IaaS, CaaS, PaaS, FaaS, or SaaS.
- Description: Describes key functionalities of the cloud service.

Not included in this section are serverless cloud services offered by other cloud providers such as AWS and Microsoft Azure. However, the fundamentals behind serverless cloud services are the same. For example, Google Cloud SQL[55] and AWS RDS[56] (Relational Database Service), although marketed under different names, both offer a SaaS relational database service.

[55] https://cloud.google.com/sql
[56] https://aws.amazon.com/rds/

Google Cloud Service	Category	Service Type	Description
App Engine[57]	Compute	PaaS	Allows developers to deploy applications for a limited number of supported runtimes including Node.js, Java, Ruby, C#, Go, Python, or PHP.
Cloud Run[58]	Compute	CaaS	Unlike App Engine, Cloud Run supports any runtime and can scale down to zero container when there is no traffic.
Cloud Functions[59]	Compute	FaaS	While App Engine and Cloud Run can be used to deploy a full-fledged application, Cloud Functions can be used to deploy only stateless functions which can be triggered via HTTP or via events within Google Cloud resources.
Cloud Storage[60]	Storage	SaaS	Provides object storage for objects such as images, videos, static HTML/ CSS/ JavaScript files.
Filestore[61]	Storage	SaaS	Provides hierarchical, file system storage.
Cloud SQL[62]	Storage	SaaS	Provides relational database service for MySQL, PostgreSQL and MySQL Server.
Cloud Spanner[63]	Storage	SaaS	Like Cloud SQL but with global and horizontal scalability.
Cloud Firestore[64]	Storage	SaaS	Provides document NoSQL database service.
Cloud Bigtable[65]	Storage	SaaS	Provides key-value and wide-column NoSQL database service.
Cloud Memorystore[66]	Storage	SaaS	Provides in-memory, key-value storage service for Redis and Memcached.
Eventarc[67]	Service integration	SaaS	Allows developers to build event-driven architecture by connecting different Google Cloud resources. For example, a social media platform might allow users to upload an image to their profile page if the image does not violate some requirements such as no nudity. When an image is uploaded and stored in a cloud storage, Eventarc can read the log of the cloud storage resource and recognizes that a new image has been loaded. Eventarc then delivers an event to Cloud Functions, triggering Cloud Functions to run an image recognition machine learning model and check if the image complies with the requirements.
Cloud Scheduler[68]	Service integration	SaaS	Allows developers to schedule tasks as known as cronjobs to be executed at a pre-defined time or regular intervals. An example of such work units is updating cached data every 10 minutes. Each cronjob created by Cloud Scheduler is sent to a “target” as scheduled, where the work of the task is executed.
Workflows[69]	Service integration	SaaS	Allows developers to define and execute a workflow which orchestrates different services. Use cases include app integration and microservice orchestration, business process automation, data and machine learning pipelines, and IT process automation. For example, the developers of a web shop can design a Workflow that 1) queries a database for records of abandoned virtual shopping carts 2) triggers a Cloud Function to draft a reminder email reminding shoppers that they have items in their shopping cart 3) triggers another Cloud Function to call an external email API and send out the reminder email.
Cloud Tasks[70]	Service integration	SaaS	Allows developer to separate out an asynchronous task that can be performed independently from the application flow and send the task to a “target” for where the task is completed. If the target fails to the perform the task, the task is retried.

Table 5. Google Cloud Serverless Cloud Services

57 https://cloud.google.com/appengine
58 https://cloud.google.com/run#section-12
59 https://cloud.google.com/functions#section-14
60 https://cloud.google.com/storage
61 https://cloud.google.com/filestore
62 https://cloud.google.com/sql
63 https://cloud.google.com/spanner
64 https://firebase.google.com/docs/firestore
65 https://cloud.google.com/bigtable
66 https://cloud.google.com/memorystore
67 https://cloud.google.com/eventarc/docs/overview
68 https://cloud.google.com/scheduler/docs/overview
69 https://cloud.google.com/workflows
70 https://cloud.google.com/tasks/docs/dual-overview

2.9 RQ#7: What Is Existing Research on AR Implementation Methodologies?

Existing AR Implementation Methodologies

To confirm the *originality* of this research, a search query is conducted to explore existing research on AR implementation frameworks. Three papers were found.

Paper [85] presents a six-step implementation methodology for implementing AR in the marketing industry as shown in Figure 26 and summarized in Table 6.

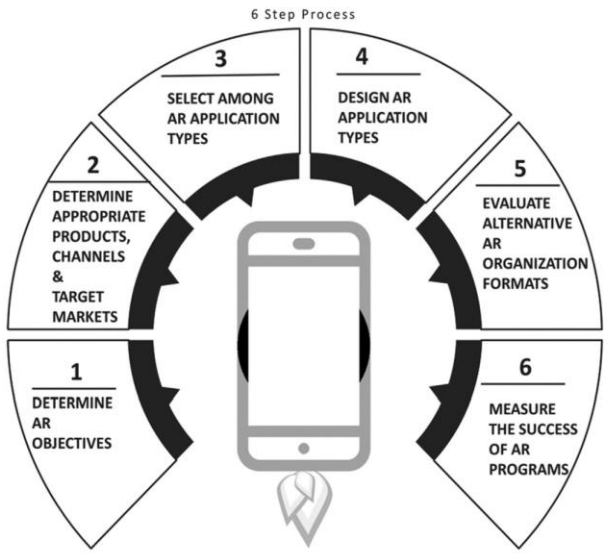


Figure 26. AR Implementation Methodology for Marketing Industry [85]

Process Step	Description
Step 1: Determine AR objectives	AR objectives can fall into the following categories: • Increase profits through higher sales • Increase profits through lower expenses • Generate excitement for retailers or a product • Facilitate consumer purchase process thru increased product info • Implement and upgrade a firm's omnichannel strategy
Step 2: Determine products, channels, and target markets	Companies need to determine which products in their portfolio are best marketed to which target markets using AR technologies. For example, companies can choose to build AR experiences only for their 10 best-selling products or its best retail locations as a proof-of-concept before wide implementation.
Step 3: Select among AR application types	The AR application types discussed in the paper are: • Product demonstration via interactive media • Try-on applications: e.g., trying out virtual sunglasses before purchase • Try-out applications: e.g., Ikea Place app in Figure 12 • Remote technical support
Step 4: Design an AR application types	The paper discusses the technical and consumer-based issues relevant for the development of an AR application: • Technical issues ○ Realism of AR images ○ Ease in developing, maintaining, and upgrading an AR app ○ Cross-platform/ cross-device ○ Ability to use existing CAD models as the basis for AR content • Consumer-based issues ○ Quality of user experience ○ Download speed ○ Ease of use ○ Availability of essential functions like calling, chat, GPS, social media ○ Hedonistic benefits to users: aesthetic, experiential, enjoyment, realism ○ Provide interactivity with AR content ○ Vividness: realism or richness of AR content
Step 5: Evaluate alternative AR organization formats	Companies can choose to develop an AR application using in-house talents or outsource the development work to a third party.
Step 6: Measure the success of AR programs	Use ROI calculation, development cost, and user journey analysis to determine the financial impact of an AR application.

Table 6. Summary of AR Implementation Methodology Presented in Paper [85]

This paper [85] has several deficiencies:

- Steps 1, 2, and 3 are specific to the marketing industry and does not offer concrete insights into the design thought process for a general AR use case.
- Step 4 provides more of a list of success factors which dictate user acceptance of AR solutions. It does not, however, provide guidance on 1) which AR platforms are most

appropriate for a selected use case nor 2) which AR development tools are best suited to develop the AR solutions.

- Step 5 does not provide technical guidance on the development and deployment of a AR application – how to design and deploy AR solutions efficiently, at scale, securely, and cost-effectively.

Another paper [86] describes a six-process step for the implementation of an AR application as shown in Figure 27 where it is recommended to analyze each process step four dimensions of impact: social, technology, process, organization. The methodology is summarized in Table 7.

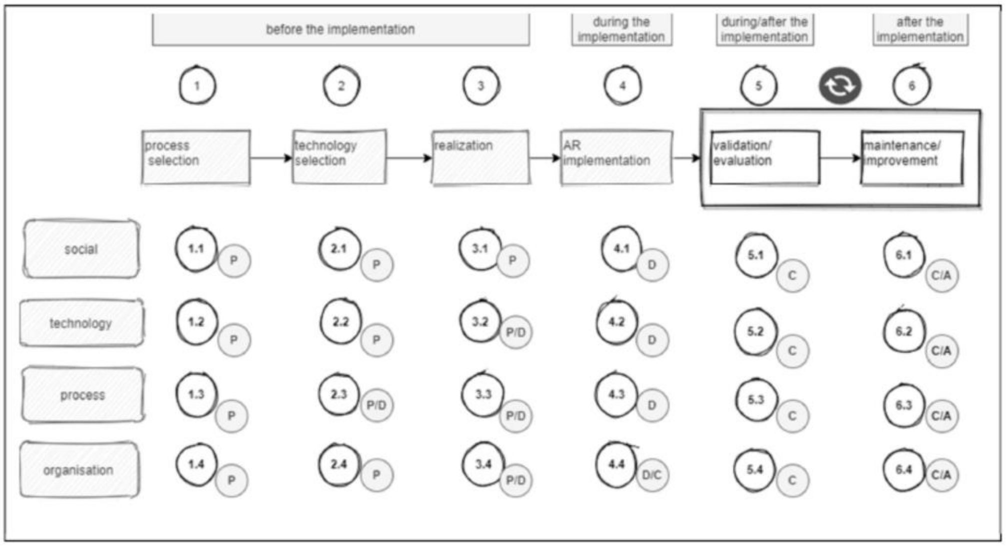


Figure 27. AR Implementation Methodology [86]

Process Step	Description
Step 1: Process selection	Select appropriate processes which can be suitably for enrichment with the use of AR as some process might be too complex and variable for AR. AR development requires human and financial resources. User acceptance is important. Poorly AR experience process might lead to employee frustration and aversion to AR.
Step 2: Technology selection	Conduct an analysis of technologies and technology providers to determine which technologies/ technology providers are best suited to fulfill the necessary requirements.
Step 3: Realization	Create a project plan to outline the activities to be carried out during the implementation e.g., facilitate training sessions to teach employees how to use AR tools. Collect requirements of the AR application. Hire developers to develop the AR application based on the requirements.
Step 4: AR implementation	The implementation should be carried out in 3 phases: 1. Under laboratory environment 2. Under real environment without interfering with the running operation 3. In the running operation. Process KPIs (cost, time, error rate) and other KPIs should be monitored (user acceptance, employee wellbeing, technology, process design, and others).
Step 5: Validation and evaluation	Measured KPIs are analyzed for identifying opportunities for improvement.
Step 6: Maintenance and improvement	Measured KPIs are analyzed for identifying opportunities for improvement.

Table 7. Summary of AR Implementation Methodology Presented in Paper [86]

This paper [86] offers a high-level guidance for the implementation of AR applications. It is a particularly useful for business leaders who want to understand the general process for AR implementation, the impact of AR implementation, and the types of human and technical resources are required.

However, the paper [86] lacks in-depth guidance.

- Step 1 stresses the importance of selecting a suitable process which can be improved with the use of AR and that a poorly selected process might result in employee frustration and aversion to AR. However, there is no concrete strategies for facilitating the design thought process so that the most suitable process can be identified.
- Step 2 suggests that an analysis of technologies and technology providers should be conducted but leaves no guidance for how to conduct such an analysis or on what dimensions should be the technologies and technology providers should be measured. For example, how to choose between different AR platforms (webAR vs. native AR vs. AR headset) and why? How to choose between different AR development tools to realize the selected use case for a specific AR platform?

- Step 3 provides a good project management practice of using a project plan to track the project progress. However, it does not provide technical guidance for the development of AR solutions.
- Step 4 provides a good practice of deploying a new AR solution in phases (under laboratory environment, under real environment without interfering with the running operation, in the running operation). However, it does not provide technical guidance for the deployment of AR solutions on on-premises or cloud infrastructure so that the AR solutions can be deployed at scale, securely, and cost-effectively.

Another paper [87] presents an interesting AR design thought process as summarized in Table 8:

Process Step	Description
Step 1: identify *performance measures* to be improved	E.g., the time it takes for a stranded military combatant to locate and reunite with his or her troop.
Step 2: Identify *cognitive tasks* which help improve the target performance measures	E.g., the ability of the combatant to orient and navigate in an unfamiliar area
Step 3: Design *AR interfaces* which have features assisting with *perceptual tasks* required for performing the cognitive tasks.	Perceptual tasks involve visual, auditory, tactile, smell and taste. For example, design an “x-ray vision” AR system that allows combatants to view information about a specific area (viewing is a visual *perceptual task*) which help the stranded combatant orient his or herself and find his/ her way back to the location of his/ her troop (*cognitive task*) in the shortest amount of time possible (*performance measure*).
Step 4: Test the AR interfaces by measuring the target *performance measure*.	Measure the amount of time a stranded combatant takes to locate and reunite with the troop when using the “x-ray vision” AR system.

Table 8. Summary of AR Implementation Methodology Presented in Paper [87]

This paper [87] fills in the gap present in paper [85] [86] regarding the design thought process for a general AR use case. However, paper [87] does not provide guidance on how to assess whether a process is a good candidate for AR. Rather, the paper assumes that the target process can be and should be enhanced with the use of AR and proceeds to decomposing the process into perceptual tasks to be assisted with the use of AR. This approach could lead to a mistake of forcing a technological solution, AR in this case, on an ill-suited problem.

Step 3 of paper [87] does not provide sufficient technical and non-technical guidance for how to design, develop and deploy the AR interfaces. For example, what AR platform (webAR, native AR, AR applications on AR headset) should be chosen and why? What AR development tools are suitable for the use case requirements? How to deploy the newly developed AR application?

Step 4 of paper [87] appears to rely on performance measures to assess the utility of an AR solution but neglects to consider other factors which influence user acceptance. For example, a solution might be efficient but still be rejected by the users if the costs of using the solution outweigh the benefits.

Research Gap Summary and Research Output Scope

Table 9 summarizes the research gaps identified as part of the literature review (column “Research Gap”) and the papers in which the research gaps are present (column “Research Gap Source”).

Additionally, column "Scope of the AR Implementation Methodology” includes the questions to be answered as organized by the Design, Development, and Deployment phase of the AR implementation methodology (the output of this research) to close the identified research gaps.

#	Research Gap	Research Gap Source	Scope of the AR Implementation Methodology
1	No guidance on how to choose an appropriate process to be enhanced with the use of AR. i.e., *Process suitability for AR.*	See paper [85] [86] [87] in section 2.9 RQ#8: What Is Existing Research on AR Implementation Methodologies?	An example of a process is where a maintenance worker walks around a building and inspect different smoke detectors to ensure they still function properly. An AR application can allow the maintenance worker to point a mobile device at a smoke detector and view the information about the smoke detector rendered in 3D, speeding up the inspection process – see Figure 9. This research gap is partially answered in the design thought process presented in paper [87]. However, the approach presented in paper [87] has several limitations as explained above. The overall objective of the proposed ARIM to be designed in phase 2 of the graduation thesis is to guide the design, development, and deployment of AR applications – see Figure 2 for more details. To close this research gap, the proposed ARIM (design phase) will address: RQ#9.1: What makes a process ideal to be enhanced with the use of AR? To answer RQ#9.1, the proposed ARIM will: 1) present a list of characteristics which make a process a good candidate for AR 2) discuss example use cases to explain the proposed characteristics.
2	No guidance on how to choose between different AR platforms: webAR vs. native AR vs. AR headset. i.e., *AR platform suitability.* No guidance on how to choose between different AR development tools. i.e., *AR development tool suitability*.	See section WebAR vs. Native AR vs. AR Headset See paper [85] [86] [87] in section Existing AR Implementation Methodologies.	Each AR platform has its capabilities, and limitations. Therefore, the design questions of which AR platform and AR development tools are applicable should be driven by 1) the requirements of the selected process and 2) the capabilities and limitations of different AR platforms. To close this research gap, the proposed ARIM (design phase) will address: RQ#9.2: How to choose between webAR vs. native AR vs. AR headsets for a selected process? and RQ#9.3 How to choose between different AR development tools for a selected process and AR platform? To answer RQ#9.2, the proposed ARIM will: 1) catalogs the requirements which are used to assess the capabilities and limitations of different AR platforms, 2) benchmarks different AR platforms with regards to the requirements as documented in a non-exhaustive artifact called "AR Platform Decision Matrix", 3) discusses example use cases to demonstrate how an AR platform might be suitable for one use case but not suitable for another use case. To answer RQ#9.3, the proposed ARIM will: 1) catalogs the requirements which are used to assess the capabilities and limitations of different AR development tools, 2) benchmarks different AR development tools with regards to the identified requirements as documented in a non-exhaustive artifact called "AR Development Tool Decision Matrix" 3) discusses example use cases to demonstrate how an AR development tool might be suitable for a use case and AR platform but not suitable for another use case and AR platform.
3	No technical guidance on the *development + deployment* of AR applications.	See paper [85] [86] [87] in section Existing AR Implementation Methodologies; See section WebAR vs. Native AR vs. AR Headset.	Neither the existing AR implementation frameworks nor use cases in literature provide technical guidance for the development and deployment of AR applications. The existing use cases are primarily small-scale proof-of-concepts and, therefore, do not provide guidance on how to develop and deploy AR solutions efficiently, at scale, securely, and cost-effectively. To close this research gap, the proposed ARIM (development phase) will address: RQ#9.4: What are the requirements for the development of AR apps? and RQ#9.5: What are the requirements for the deployment of AR apps?

Table 9. Research Gap Summary and Research Output Scope

2.10 Chapter Summary

This chapter serves as phase 1 of the graduation thesis where a systematic literature review is conducted to answer eight research questions (i.e., knowledge questions) outlined in section 1.3 Research Objective, Research Questions (RQs), and Research Methodology and set the scientific foundation for the ARIM to be designed in phase 2.

Section 2.1 Literature Review Protocol explains the procedure of the literature review. Following the protocol, *111 papers* are selected from search query results for detailed examination, and *86* of which are referenced in this report.

Section 2.2 RQ#1: What Is AR? to section 2.9 RQ#7: What Is Existing Research on AR Implementation Methodologies? summarize the findings of the literature review as organized by the research questions.

As summarized in sectionResearch Gap Summary and Research Output Scope, below is a list of research gaps where existing literature does not provide concrete guidance for the creation of AR applications:

1. RQ#8.1 - Process suitability for AR: How to determine if a business process can and should be enhanced with the use of AR?
2. RQ#8.2 - AR platform suitability: How to determine which AR platform (webAR vs. native AR vs. AR headset) is most suitable for a selected business process?
3. RQ#8.3 - AR development tool suitability: How to determine which AR development tools are most suitable to deliver the proposed AR applications for the selected business process and AR platform?
4. RQ#8.4 and 8.5 - Technical guidance for the development and deployment of AR apps: What is technical development and deployment guidance so that the proposed AR applications can be developed and deployed efficiently, securely, cost-effectively, and at scale?

To address these research gap, phase 2 of the graduation thesis will propose an industry-agnostic, tool-agnostic AR implementation methodology (ARIM) as documented in CHAPTER 3: ARTIFACT DESIGN – AR IMPLEMENTATION METHODOLOGY (ARIM), demonstrate how the ARIM can be used in practice by implementing a webAR app as documented in CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9), and evaluate the utility of the proposed ARIM by conducting expert interviews as documented inCHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10). The expert feedback will be used to further refine the proposed ARIM in CHAPTER 6: ARTIFACT REVISION.

CHAPTER 3: ARTIFACT DESIGN – AR IMPLEMENTATION METHODOLOGY (ARIM)

As discovered in section 2.9 RQ#7: What Is Existing Research on AR Implementation Methodologies?, there are three existing AR implementation frameworks with the identified deficiencies as documented in section Research Gap Summary and Research Output Scope.

This research proposes a new ARIM building off the merits of the existing frameworks and offering additional valuable insights.

Section 3.1 RQ#8: What Does an Industry-Agnostic, Tool-Agnostic AR Implementation Methodology (ARIM) Look Like? discusses the high-level design of the ARIM as shown in Figure 30.
Section

3.2 RQ#8.1: What Makes a Process a Good Candidate for Enhancement with the Use of AR? (Discovery Phase > Process Suitability for AR)3.6 RQ#8.5: What Are the Requirements for the Deployment of AR Applications? (Deployment Phase) answer RQ#9.1-9.5 and adds more details to the high-level design.
Section 3.7 ARIM Final Design consolidates section 3.1-3.6 into the final design of the proposed ARIM as shown in Figure 46.

3.1 RQ#8: What Does an Industry-Agnostic, Tool-Agnostic AR Implementation Methodology (ARIM) Look Like?

Key Inputs for the Proposed ARIM

Five key inputs drive the design of the proposed ARIM:

1. Software development lifecycle
2. Design Thinking methodology
3. AR adoption success factors: documented in section 2.4 RQ#3: What Factors Influence AR Adoption? of CHAPTER 2. SYSTEMATIC LITERATURE REVIEW.
4. Technical knowledge of AR and cloud infrastructure: documented in the remaining sections of CHAPTER 2. SYSTEMATIC LITERATURE REVIEW.
5. The implementation of a WebAR application: documented in CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9).

Input#1: Software Development Lifecycle (SDLC)

Figure 28 illustrates a 10-phase software development lifecycle as defined in the formal procedure that the U.S. Department of Justice uses for the acquisition, design, development, implementation, and maintenance of information systems [88].

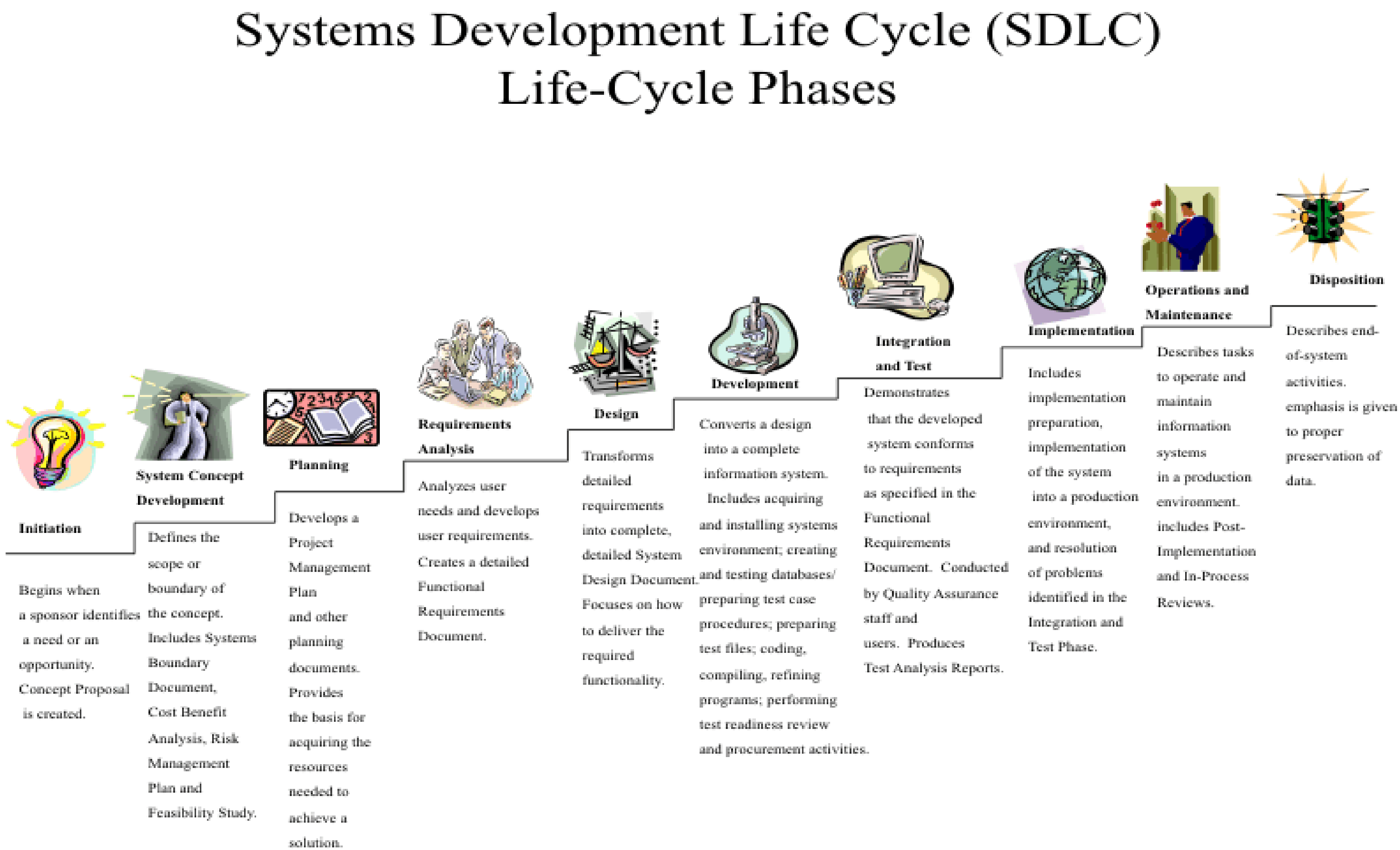


Figure 28. Software Development Lifecycle (SDLC) [88]

Input#2: Design Thinking Methodology

Design Thinking is a human-centric problem-solving approach first proposed by cognitive scientist and Nobel Prize laureate Herbert A. Simon and later expanded by many industry professionals. The methodology consists of 5 non-linear, iterative stages as shown in Figure 29 and summarized as follows [89]:

1. Empathize: Interact with users to gain an empathetic understanding of the users' needs and challenges
2. Define: Analyze the observations collected during the Empathize stage to define the core problems the users are facing
3. Ideate: Brainstorm different solutions based on the problem statements from the Define phase
4. Prototype: Develop small-scaled prototypes for the proposed solutions from the Ideate phase
5. Test: Try out the prototypes with users and make further modifications.

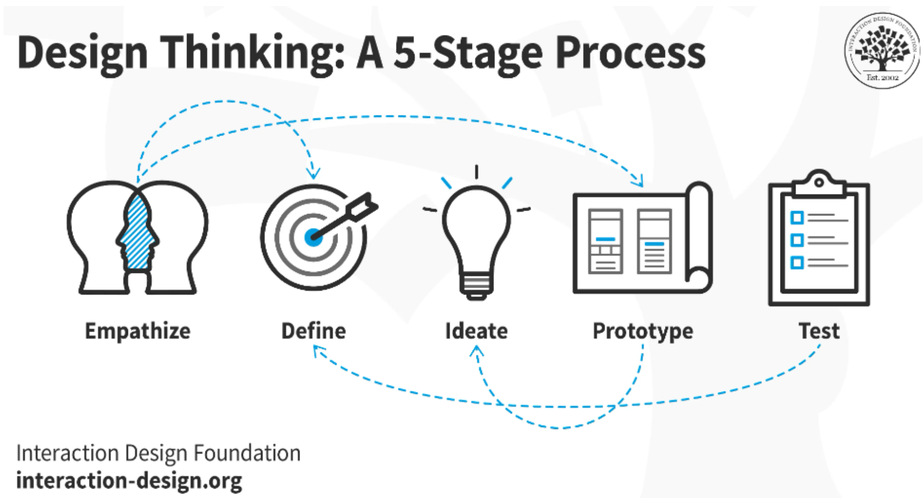


Figure 29. Design Thinking Methodology [89]

Input#3: AR Adoption Success Factors

To design an ARIM, a possible approach is to read up on the technical specifications of different AR platforms and AR development tools. However, this approach turns out to be ineffective since it focuses too much on the technicality and not enough on the users who eventually decide whether to adopt or reject the proposed AR solutions.

Therefore, this research pursues an alternative approach of deriving selection criteria and requirements for the ARIM directly from AR adoption success factors as examined in section 2.4 RQ#3: What Factors Influence AR Adoption? of the literature review. The advantage of this approach is that the proposed ARIM will be built upon AR adoption success factors, enabling AR solutions implemented using the ARIM to achieve high user acceptance.

Detailed descriptions of each AR adoption success factor can be found in Table 4.

Input#4+5: Technical Knowledge of AR and Cloud Serverless Infrastructure + The Implementation of a WebAR Application

Having a technical knowledge of AR enables the author to assess different AR platforms and development tools as part of the proposed ARIM.

In parallel to designing the proposed AR, the author also implements a webAR application which serves two purposes:

1. Demonstrate how the proposed ARIM can be put in practice.
2. Insights gained while working on the webAR application helps the author finetune the proposed ARIM, especially section 3.4 RQ#8.3: How to Choose Between Different AR Development Tools for a Selected Process and AR Platform? (Design Phase > AR Development Tool Suitability) where the ARIM proposes a list of selection criteria for benchmarking different AR development tools.

CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9) provides more details on the implemented webAR application.

How Key Inputs Shape the Design of ARIM

In summary, because AR applications are in essence a piece of software, the proposed ARIM mirrors the 10-phase SDLC in Figure 28 and incorporate insights from the Design Thinking methodology in Figure 29, AR adoption success factors from section 2.4 RQ#3: What Factors Influence AR Adoption?, technical knowledge of AR/ cloud infrastructure obtained from the remaining sections of CHAPTER 2. SYSTEMATIC LITERATURE REVIEW, and the implementation of a webAR application documented in CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9).

For simplicity purposes, the proposed ARIM consists of five, non-linear, iterative phases:

Phase 1: Discovery

- During the Discovery phase, business stakeholders and developers uncover problem(s) the business stakeholders are facing and assess whether such problems can be solved with the use of AR.
- Corresponds to the System Concept Development, Planning and Requirement Analysis phase of the SDLC in Figure 28.
- This phase is concerned with the question of *process suitability for AR* as phrased in *RQ#9.1* (What Makes a Process Ideal to Be Enhanced with the Use of AR?) – a research gap identified during the literature review (see section Research Gap Summary and Research Output Scope). The term “suitability” here refers to whether a technology, hardware or software, can be and should be used to solve a problem at hand.

Phase 2: Design

- Once it is confirmed that the selected business process can be and should be enhanced with the use of AR, the implementation process enters a series of iteration cycles where each cycle consists of the design, development, deployment, and validation phase and where at the end of each cycle, a working AR app is delivered as built upon the previous iterations and tested with users. This approach aligns with the Design Thinking methodology discussed above.
- During the Design phase, business stakeholders and developers design the AR solution based on 1) the requirements of the business process to be enhanced with the use of AR and 2) the capabilities and limitations of existing AR platforms and AR development tools.
- Corresponds to the Design phase of the SDLC in Figure 28.
- This phase is concerned with the questions of *AR platform suitability* as phrased in *RQ#9.2* (How to Choose Between WebAR vs. Native AR vs. AR Headsets for a Selected Process?) and *AR development tool suitability* as phrased in *RQ#9.3* (How to Choose Between Different AR Development Tools for a Selected Process and AR Platform) – research gaps identified during the literature review.

Phase 3: Development

- During the Development phase, developers program the AR software solutions as specified by the technical and non-technical requirements collected in the Discovery and Design phase using appropriate AR development tools.
- Corresponds to the Development and Integration and Test phase of the SDLC in Figure 28.
- This phase is concerned with *RQ#9.4* (What Are the Requirements for the Development of AR Applications?) – a research gap identified during the literature review.

As stated in section

- WebAR vs. Native AR vs. AR Headset, the technical mechanism underlying AR is the same for all AR platforms: webAR, native AR, and AR headset. This mechanism includes two key technologies: 1) A tracking engine 2) A rendering engine.
- For this reason, the ARIM's Development phase does not segregate into technical guidance for webAR vs. native AR vs. AR headset development. Instead, it aims to provide high-level technical guidance for the development of high-quality AR software solutions irrespective of the chosen AR platform.

Phase 4: Deployment

- During the Deployment phase, developers deploy the previously developed AR software solution to some infrastructure so that the users can access and use the AR solutions.
- Corresponds to the Operation and Maintenance phase of the SDLC in Figure 28.
- This phase is concerned with *RQ#9.5* (What Are the Requirements for the Deployment of AR Applications?) – a research gap identified during the literature review.
- Like the Development phase, the Deployment phase is not specific to any AR platform. Instead, it aims to provide high-level technical guidance for the deployment of AR software solutions to the cloud environment.

Phase 5: Validation

- During the Validation phase, a working AR app which has been designed, developed, and deployed earlier is now tested with business users for continuous feedback.
- This phase does not correspond to any phase of the SDLC in Figure 28. Instead, it is adapted from the "Test" stage of the Design Thinking methodology with the objective of keeping the users engaged throughout the implementation process and ensuring the AR solution being developed meets the user requirements.

Note that the Initiation and Disposition phase of Figure 28 are not covered in the ARIM:

- In the context of Accenture's client projects, the Initiation phase typically refers to the period where a prospective client makes a request for proposal (RFP), and Accenture will then submit a bid establishing itself as a credible technology vendor to help the client deliver the project. For that reason, the Initiation phase is excluded from the ARIM.
- The Disposition phase is also considered out of scope to keep the ARIM simple.

Next, the AR adoption success factors from the literature review are re-visited to further shape the five phases of the ARIM mentioned above. Specifically, each *success factor* is translated to *high-level selection criteria/ requirements*, contextualized with *example use cases*, and eventually decomposed into *low-level selection criteria/ requirements*.

For example, *success factor* "F1. Task identification" cataloged in Table 4 states that AR solutions are more likely to be adopted when the AR solutions are carefully designed to accommodate the working conditions of the task intended to be enhanced with the use of AR. This success factor is translated to a <u>*high-level selection criterion*</u> for AR platforms called "usage condition-specific selection criteria."

Working conditions mean different things for different types of jobs. For example, construction workers are required to always wear a hardhat while being on site. For this example use case, the ARIM proposes a <u>*low-level selection criterion*</u> for AR platforms called "compliance with health and safety standards". Using that low-level selection criterion, we can benchmark different AR platforms and discover that HoloLens 2 is one of a very few AR headsets which offer a hardhat-integrated version, making it a strong AR platform to be considered for this example use case.

Table 10 maps 17 AR adoption success factors to 17 different high-level selection criteria/ requirements as categorized into 5 phases of the proposed ARIM and explains the rationale behind the mappings.

In section 3.2-3.6, the high-level selection criteria/ requirements will be further contextualized with example use cases and decomposed into low-level selection criteria/ requirements.

Category	Sub-category	Success factor (SF)	Mapped to: ARIM phase	Mapped to: High-level selection criteria (C)/ requirements (R)	Rationale behind the mapping
Organization	Company	SF1. Task identification: AR solution is suitable for working conditions + helps enhance worker performance	Discovery > Define problem	CR1. Identify process requirements	In order to make sure AR solutions properly support target business processes, business stakeholders and developers must first understand the requirements of the processes of interest.
			Design > AR Platform suitability	CR2. Usage conditions-specific selection criteria	This SF is mapped to a high-level selection criterion for AR platforms called "PR1. Usage conditions-specific selection criteria" which is intended to group all factors which are unique to the conditions in which AR solutions are used and should be considered when choosing between different AR platforms. For example, when a surgeon requires the use of AR for easy access to patient's medical data during a surgery, AR headsets stand out as a strong contender since the surgeon can operate the headsets hands-free, allowing them to continue using both of their hands to carry out surgery procedures. This is an advantage that AR headsets have over native AR and webAR where users need to hold a mobile device with one of their hands.
			Discovery > Define problem; Validation	CR3. Define value drivers + KPIs CR4. Validate value drivers + measure KPIs	To confirm that the AR solutions help enhance worker performance, key performance indicators (KPIs) need to be defined during the Discovery phase and measured during the Validation phase. For example, if an AR solution is intended to help a warehouse order picking worker works more efficiently, a possible KPI is the time it takes the worker to finish picking an order with and without the use of the AR solution.
		SF2. Compliance with health + safety concerns: distractions, visual fatigue, etc.	Discovery > Define problem	CR1. Identify process requirements: Health and safety standards	This SF is mapped to a high-level requirement in the Discovery phase which considers health and safety standards as one of the requirements of the process to be enhanced with the use of AR.

			Design > AR platform suitability	CR2. Usage conditions-specific selection criteria: Health and safety standards	This SF is also mapped to a high-level selection criterion for AR platforms called “PR1. Usage conditions-specific selection criteria” which includes a low-level selection criterion called “Compliance with health and safety standards” since health and safety standards are different job-by-job. For example, for surgeons to use an AR solution during surgery, the hardware the AR solution runs on (mobile device or AR headsets) must be certified for operating in clean room environments [90].
		SF3. Implementation cost: design, develop, deploy, maintain	Design > AR platform suitability	CR5. Hardware-specific selection criteria: Hardware + subscription cost	This SF is mapped to a high-level selection criterion for AR platforms concerning the cost of the hardware and the subscriptions required to use the hardware.
			Design > AR development tool suitability	CR6. Ease of development	This SF is mapped to a high-level selection criterion for AR development tools called “Ease of development.” The more user friendly the tools are, the lower the learning curve for the teams to learn and use the tools.
			Design > AR development tool suitability	CR7. Licensing cost	This SF is mapped to the licensing cost of the AR development tools used to develop the AR solutions.
			Development	CR8. Flexible architecture for maintainability, agility, and time-to-market	This SF is mapped to a high-level requirement for the development of AR solutions regarding the architecture of the AR solutions. The more flexible the architecture is, the easier it is to develop and maintain the AR solutions.

			Deployment	CR9. Continuous integration and continuous delivery (CI/ CD)	CI/CD is a practice of continuous integration and continuous delivery, meaning when a developer makes a change to the source code to implement a new feature for the AR solution, the CI/CD pipelines automatically trigger the build process where the source code is converted into executable code (called built artifacts), run testing on the built artifacts, and eventually deploy the built artifacts to the production environment where the new feature is available to all users [91]. Implementing CI/CD pipelines automate tedious developer's tasks and enable developers to work more efficiently, which in returns cuts down costs for the development, deployment, and maintain of AR solutions.
			Deployment	CR10. Design for cost optimization	The cost of running AR solutions in a cloud infrastructure is another cost factor. Cloud infrastructure can be configured to minimize idle capabilities and optimize on costs.
		SF4. Change management: adequate training and support	Validation	CR11. Create training materials	After a new AR app has been deployed, the AR app will be tested with end users during the Validation phase. In parallel, based on user feedback, training materials should be developed to provide users with adequate instructions for how to use the AR solutions effectively.

		SF5. User experience during AR usage: user-friendly UI, intuitive interactions, etc.	Design > AR platform suitability	CR5. Hardware-specific selection criteria: user interactions	To provide intuitive interactivity for users, the capabilities, and limitations of different AR platforms for implementing different types of user interactions should be taken into consideration. Examples of user interactions include but not limited to voice commands and gesture commands (using hand tracking technology). For example, when designing an AR solution for workers working on a shopfloor, it is important that the users can comfortably use the AR solution in a loud working environment where industrial machines are running in the background. In that case, gesture commands might be more appropriate than voice commands.
			Deployment	CR12. Design for performance	When deploying an AR solution in the cloud, the cloud architecture affects user experience in terms of latency. For example, if the AR solution runs on a cloud server located in Europe, European users should experience acceptable latency. If the AR solution is now also used by Asian users, Asian users might experience higher latency due to the long distance the data has to travel between the server in Europe and the user's device in Asian. Therefore, the cloud architecture of the AR solution should be designed so that the AR solution can scale up to support more users dispersed across geographies.
	User	SF6. Privacy and data protection	Deployment	CR13. Design for security	The cloud infrastructure the AR software solution is running on should be secure so that all data access requests initiated by the AR solution must be authenticated and authorized properly.
		SF7. User technology quotient: the more tech savvy users are, the higher adoption rate	Validation	See CR11. Develop training materials	See CR11. Develop training materials

		SF8: Address psychological risks e.g., fear of over-reliance on technology, job loss, etc.	Discovery > Define problem	CR14. Conduct a risk assessment for psychological and non-psychological risks	This SF is mapped to an activity in the Discovery phase where business stakeholders and developers conduct a risk assessment. This risk assessment covers psychological and non-psychological risks which might impede the implementation of the AR solutions. For example, workers who are tasked with inspecting shopfloor machines using an AR solution might fear that the AR solution will display inaccurate sensor data, making them unable to detect problems with the machines and causing the workers to reject the AR solution. Based on the identified risks, different countermeasures can be implemented. Additionally, the risk assessment should be kept updated throughout the implementation process.
Technology	Hardware	SF9. Essential hardware requirements	Design > AR platform suitability	CR5. Hardware-specific selection criteria	This SF outlines a few essential hardware requirements such as battery life and display resolution. All these requirements along with additional ones proposed in this research are grouped under a high-level selection criterion called “Hardware-specific selection criteria”, which help business stakeholders and developers choose between different AR platforms.
		SF10. Ergonomics	Discovery > Define problem	CR1. Identify process requirements	During the Discovery phase, business stakeholders and developers collect the requirements of a process to be enhanced. One of these requirements should be the ergonomics of the developed solution. For example, if AR is deemed appropriate for the process and AR headset is chosen as the AR platform, it is important that the users feel comfortable wearing the AR headset at work without experiencing too much head fatigue.
			Validation	CR15. Collect user feedback	When an AR app is tested during the Validation phase, users can give feedback on whether the AR app is ergonomic enough for long-term usage.

	Software	SF11. Essential software requirements: acceptable latency while ensuring accurate tracking of markers and realistic rendering of AR contents	Design > AR development tool suitability	CR16. Tracking capabilities CR17. Rendering capabilities	This SF is mapped to high-level selection criteria called “tracking capabilities” (CR16) and “rendering capabilities” (CR17). Different use cases require different tracking and rendering capabilities. The ARIM will propose a set of specific tracking and rendering capabilities as low-level selection criteria and explain how these capabilities might be applicable for different use cases.
		SF12. Integration with enterprise systems	N/A	N/A	This SF refers to whether an AR solution is integrated with enterprise systems. For example, Microsoft HoloLens has native integration with some Microsoft products such as Microsoft Dynamics 365 Guides. Even when the required integrations between AR solutions and some enterprise (or 3rd party) systems are not currently supported, developers can create new APIs (application programming interface) to allow for the required integrations. It is hard to generalize a requirement for this SF since the types of required integrations are use case specific. Therefore, this SF is not mapped to any selection criteria/ requirements.
		SF13. Easy authoring systems	Design > AR development tool suitability	CR6. Ease of development: Low-code no-code development	This SF is mapped to a high-level selection criterion for AR development tool called “Ease of development” which includes low-level criterion called “low-code/ no-code development” which allow business users with no or limited programming knowledge to create new AR experiences.
Environment	Regulations	SF14. Compliance with employee protection regulations: health and safety concerns	See SF2	See SF2	See SF2
	Third party	SF15. Integration with 3rd party systems	See SF12	See SF12	See SF12

		SF16. External support to guide the implementation of AR solutions	N/A	N/A	This success factor is not mapped to any high-level selection criteria/ requirement of the ARIM since SF16 is more of a project management decision of whether to involve external experts to guide the implementation process.
		SF17. Reference industry-wide tried-and-tested AR solutions	N/A	N/A	This success factor is not mapped to any high-level selection criteria/ requirement of the ARIM since SF17 is more of a brainstorming exercise where companies investigate how their competitors are innovating and see if they can adopt some of those innovations for their own benefits.

Table 10. Mapping AR Success Factors to ARIM's High-Level Selection Criteria and Requirements

ARIM High-Level Design

Figure 30 illustrates the high-level design of the proposed ARIM building off the high-level selection criteria/ requirements mapped from AR adoption success factors in Table 10.

The ARIM consists of a Discovery, Design, Development, Deployment, and Validation phase. Each phase has several activities, and each activity has a set of sub-activities or selection criteria/ requirements for guiding how the activity should be carried out. Several outcomes are expected after a phase has completed. Finally, the output of a phase serves as the input for the subsequent phase.

The remaining of this section 3.1 will discuss in detail the high-level design of the proposed ARIM.

Section

3.2 RQ#8.1: What Makes a Process a Good Candidate for Enhancement with the Use of AR? (Discovery Phase > Process Suitability for AR) - 3.6 RQ#8.5: What Are the Requirements for the Deployment of AR Applications? (Deployment Phase) will address RQ#9.1-9.5 and add more details to different components of the ARIM, giving us the final design of the ARIM as shown in section 3.7 ARIM Final Design.

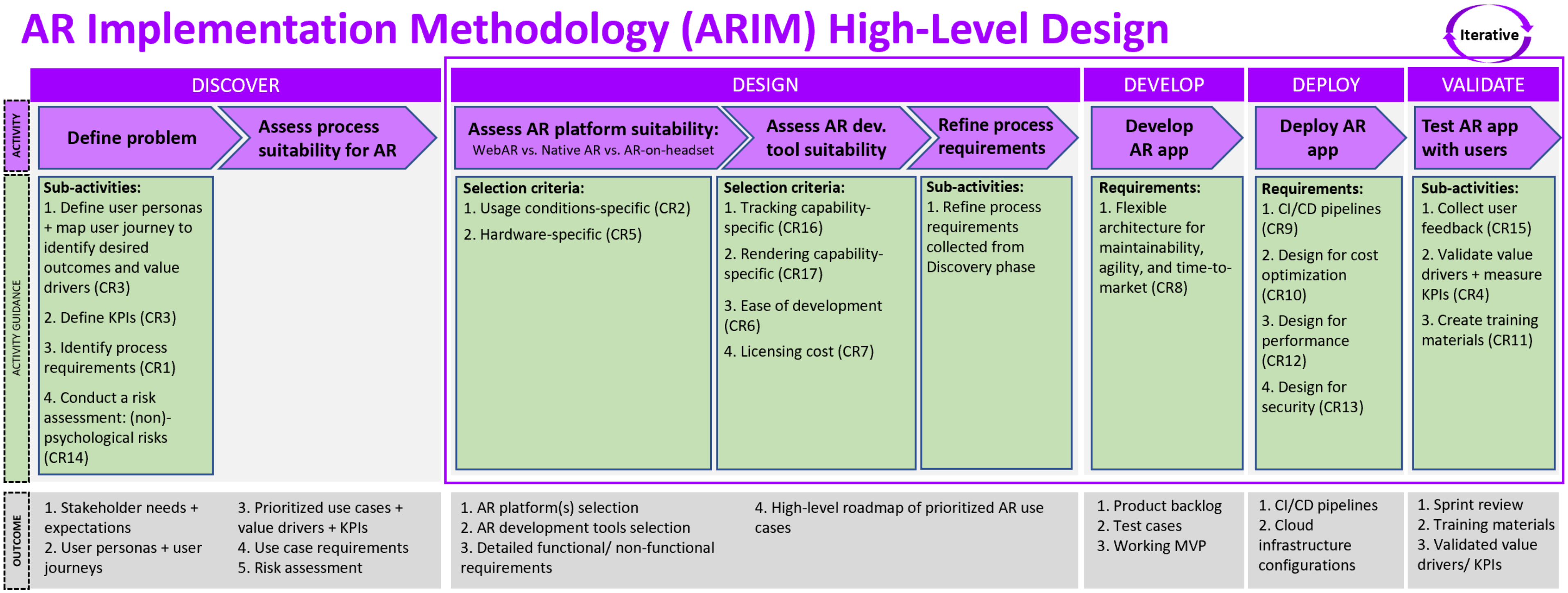


Figure 30. ARIM High-Level Design

Phase 1: Discovery

Define Problem

This is the first activity of the Discovery phase. To define a problem, the ARIM proposes four sub-activities.

Sub-activity #1: Define user personas and map user journey to identify desired outcomes and value drivers
This activity is adapted from the Empathize stage of the Design Thinking methodology and requirement “CR3 Define value drivers and KPIs” in Table 10 where user personas are drafted with each user persona represents a group of users with similar characteristics. Subsequently, a user journey is mapped out for each of the identified user personas to explore the interactions the users have with their environment and identify which checkpoints in the user journey present a pain point to the users.

For example, suppose a warehouse would like to cut down costs by optimizing its operations, so the *desired outcome* is cost reduction. We then define different *user personas*, one of which represents stock picking workers working at the warehouse, and shadow them at work to map out the *user journey* a stock picker typically goes through:

- First, receive a notification for a new order on their mobile device.
- Second, look up in an internal system to figure out which isles on the warehouse shopfloor store the items for the order.
- Third, walk to those isles to retrieve the items.
- And so on.

During this user journey, we might discover that the tasks of looking up the location of the items and walking to the isles to retrieve the items is time-consuming. Therefore, optimizing these tasks in the order picking process and helping the workers work more efficiently is a *value driver* of the desired outcome of cost reduction.

Sub-activity #2: Define key performance indicators (KPIs)
This activity is the result of the high-level requirement “CR3 Define value drivers and KPIs” in Table 10.

Once a *value driver* has been identified, a set of *KPIs* should be defined to measure the value driver. If a value driver is to optimize the order picking process, a possible KPI is the time it takes an order picker to finish picking an order.

Sub-activity #3: Identify process requirements
This activity is the result of the high-level requirement “CR1. Identify process requirements” in Table 10.

During this activity, we collect the *requirements* of the process to be optimized. The requirements should over multiple dimensions such as compliance with health and safety standards, the ergonomics of the proposed solutions, data and privacy regulations.

For the order picking process from the previous example, suppose an AR headset is chosen to guide order picking workers to the correct isles where they can retrieve items for an order, the weight of the headset must be considered to ensure that the users are comfortable wearing the headset throughout their shift with minimal head fatigue.

Sub-activity #4: Conduct a risk assessment for (non)-psychological risks
This activity is the result of the high-level requirement "CR14 Conduct a risk assessment" mapped from AR adoption success factors as shown in Table 10.

Since order picking workers are the ones working on the warehouse shopfloor and deciding whether to use the proposed solution, it is important to understand what *risk factors*, psychological and non-psychological, might impede the implementation of the AR solution.

A psychological factor might be fear of job loss. In this case, engaging the users throughout the implementation process and upskilling them are effective at addressing their fear of job loss.

A non-psychological factor might be inadequate training, making the users unable to effectively use the AR solution. In this case, asking user's feedback on the provided training is effective in ensuring the right amount of training is provided.

Assess Process Suitability for AR

This is the second objective of the Discovery phase. After having a clear understanding of the requirements of the process to be enhanced, we now determine if the use of AR can enhance the process as measured by the KPIs defined in the previous activity "Define Problem" – see section 3.2 RQ#8.1: What Makes a Process a Good Candidate for Enhancement with the Use of AR? (Discovery Phase > Process Suitability for AR) for more details.

Once the Discovery phase is complete, the following outcomes (i.e., deliverables) are expected:
1. An understanding of stakeholder needs and expectations
2. Documentation of user personas and user journeys.
3. A list of prioritized processes (i.e., use cases) to be enhanced with the use of AR along with identified value drivers and KPIs for each process.
4. The initial set of requirements for each process which was used to assess whether the process can be and should be enhanced with the use of AR.
5. A risk assessment which itemizes psychological and non-psychological risk factors which might impede the implementation of proposed AR solutions.

Phase 2: Design

After the Discovery phase confirms that a process can be and should be enhanced with the use of AR, the implementation process advances to the Design phase where there are three main activities:
1. Assess AR platform suitability
2. Assess AR development tool suitability
3. Define technical requirements

Note that the assessment of AR platform suitability and AR development tool suitability might go through multiple iterations since it is possible that the project starts with, for example, three different AR headset and eventually narrows to one final AR headset which best fulfils the requirements.

Assess AR Platform Suitability

Business stakeholders and developers must determine which AR platform (webAR vs. native AR vs. AR headset) are most appropriate for the target process.

While the technical mechanism for all AR platforms is the same with all using a tracking engine and a rendering engine, each AR platform has its own capabilities and limitations.

Section 3.3 RQ#8.2: How to Choose Between WebAR vs. Native AR vs. AR Headsets for a Selected Process? (Design Phase > AR Platform Suitability) proposes a list of selection criteria which highlight the capabilities and limitations of different platforms along with example use cases to demonstrate situations where an AR platform is suitable for one use case but not suitable for another. The selection criteria are categorized into "CR2. Usage conditions-specific selection criteria" and "CR5. Hardware-specific selection criteria) as captured Table 10.

Assess AR Development Tool Suitability

For each AR platform, several AR development tools are available, each with its own capabilities and limitations. For example, a tracking engine like MindAR[71] is open-source, but supports a limited number of tracking capabilities, and only works for webAR whereas another tracking engine called Vuforia[72] offers a wider range of tracking capabilities but only supports native AR and a number of (but not all) AR headsets.

To assist developers in choosing the right tools for the requirements of the selected process and AR platform, section 3.4 RQ#8.3: How to Choose Between Different AR Development Tools for a Selected Process and AR Platform? (Design Phase > AR Development Tool Suitability) proposes a list of selection criteria for AR development tools. The selection criteria are categorized into four groups: "CR.16 Tracking capabilities", "CR 17. Rendering capabilities", "CR6. Ease of development", "CR7. Licensing cost" as captured in Table 10.

Refine Process Requirements

Based on the process requirements collected in the Discovery phase (see section Define Problem), developers now refine these requirements into more detailed functional and non-functional requirements so that the development work can be distributed among the team.

---

[71] https://hiukim.github.io/mind-ar-js-doc/
[72] https://www.ptc.com/en/products/vuforia

Once the Design phase is complete, the following outcomes (i.e., deliverables) are expected:

1. The selection of AR platform(s) to be implemented for the proposed AR solutions: There might be situations where multiple AR platforms are selected. For example, a shopfloor worker might choose to run an AR work instructions application on a tablet while their AR headsets are being charged.
2. The selection of AR development tools for implementing the proposed AR solutions.
3. A high-level roadmap to set the general objectives and directions of the implementation process: For example, a project team can choose to implement the proposed AR solution for Android devices first before expanding to iOS devices.
4. Detailed technical requirements of the proposed AR solutions.

Phase 3: Development

Once the use case (i.e., process to be enhanced with the use of AR), AR platform, and AR development tools have been selected, the implementation process advances to the Development phase where developers build a new AR solution using the selected tools. The AR solutions are subsequently made available to the users in the Deployment phase.

The key objective in this phase is to allow developers work efficiently so that they can deliver high-quality AR software solutions for maximized user acceptance. See section 3.5 RQ#8.4: What Are the Requirements for the Development of AR Applications? (Development Phase) for detailed requirements for the development of AR solutions with a flexible architecture for maintainability, agility, and time-to-market (CR8) as captured in Table 10.

Once the Development phase is complete, the following outcomes (i.e., deliverables) are expected:

1. A product backlog which prioritizes the features to be implemented for the proposed AR solution and/ or bugs to be fixed.
2. Test cases for confirming that the AR solution works as intended.
3. A working software application for the proposed AR solution as known as a minimum viable product (MVP)

Phase 4: Deployment

Once the AR solutions are developed, they need to be deployed to be available for end users. See section 3.6 RQ#8.5: What Are the Requirements for the Deployment of AR Applications? (Deployment Phase) for details on high-level requirements “CR9. CI/CD pipelines”, “CR10. Design for cost optimization”, “CR12. Design for performance”, and “CR13. Design for security” as captured in Table 10.

Once the Deployment phase is complete, the following outcomes (i.e., deliverables) are expected:

1. CI/CD pipeline configuration to allow for the continuous integration and deployment of AR application iterations.
2. Cloud architecture including the configuration of different cloud resources on which the AR application is deployed on.

Phase 5: Validation

This phase concludes one iteration cycle where a new iteration of an AR application, which has previously been designed, developed, and deployed, is now tested with end users.

The methodology proposes the following sub-activities for this phase:

Sub-activity #1: Collect user feedback

This activity is the result of the high-level requirement "CR15. Collect user feedback" as captured in Table 10.

The AR app is tested with end users to gather early feedback. This feedback will be fed into the next iteration cycles to ensure the AR app continues to meet user requirements

Sub-activity #2: Validate value drivers + measure KPIs

This activity is the result of the high-level requirement "CR4 Validate value drivers + measure KPIs" as captured in Table 10.

This step brings us full cycle back to the Discovery phase where value drivers and KPIs were defined. During the Validation phase, the value drivers are re-examined to ensure they are still relevant and contributing to the desired outcome. Additionally, KPIs are measured to confirm that the AR solution is delivering on its promise. For example, if the proposed AR solution is intended to help a worker complete a task faster, we will measure the time it takes the user to complete the task while using the new AR app.

Sub-activity #3: Create training materials

This activity is the result of the high-level requirement "CR11. Create training materials" as captured in Table 10.

As users are testing the new AR app, new training materials can be developed in parallel to instruct the users on how to use the newly introduced features in the latest AR app iteration.

Once the Validation phase is complete, the following outcomes (i.e., deliverables) are expected:

1. Sprint review session which allows users to test out the prototype and give feedback.
2. Training materials developed for the AR solution based on user feedback.
3. Refined and validated value drivers and KPIs to ensure that the AR solution delivers on the promised business objectives.

3.2 RQ#8.1: What Makes a Process a Good Candidate for Enhancement with the Use of AR? (Discovery Phase > Process Suitability for AR)

Even though the proposed ARIM centers around AR, the methodology is not intended to force enhancing a process using AR if AR does not provide tangible benefits. Enter RQ#9.1.

To answer this research question, a total of *39 use cases* from academic literature (22 out of 39) and industry literature (17 out of 39) was reviewed in detail to provide a balanced view on both academics and industry and to synthesize common characteristics which make a process a good candidate for AR.

The use cases from academic literature are from the literature review conducted for CHAPTER 2. SYSTEMATIC LITERATURE REVIEW. The use cases from industry literature are from exploratory research using Google search engine.

These use cases were chosen from a much larger pool of query results based on the following criteria:

1. The use case has a clear description and was implemented successfully using technologies available today:
    a. This criterion excludes use cases which are still in the conceptual phase and can be implemented only with future technologies. This is because the ARIM aims to guide the implementation of AR solutions using today's technologies instead of not-yet-available technologies.
    b. This criterion does not differentiate between "good" AR use cases vs. "not good" AR use cases with "not good" AR use cases defined as ones which lack merits i.e., AR is not the right technological solution for the use case in the first place. This is because it often requires subject matter knowledge and subjective opinion to determine whether an AR use case has enough merits for a specific industry.
2. The use case discusses the specific technologies it uses as part of the implementation:
    a. This criterion excludes a large number of use cases from academic literature since they often do not elaborate on the technical details and focus on alternative objectives such as measuring user acceptance of the proposed AR solution.
    b. On the upside, this criterion allows the author to gain deeper insights into not only the use cases but also how they were implemented. These insights are beneficial for the later phases when the proposed ARIM presents a list of criteria for assessing different AR platforms and AR development tools.
3. The use case was published in the last 5 years between 2017 and 2022 to reflect the latest development and trends in AR.

These use cases represent multiple sectors:

- 14 use cases for industrials sector: e.g., manufacturing
- 8 use cases for entertainment & tourism sector: e.g., museums
- 5 use cases for healthcare sector: e.g., surgery planning and patient care

- 4 use cases for education sector: e.g., classroom learning
- 2 use cases for emergency & rescue: e.g., policing
- 2 use case for military sector: e.g., strategic military planning
- 4 use cases for miscellaneous sectors

The author reviewed the select use cases and synthesized a set of common characteristics these use cases share. Note that these characteristics are intended to be "positive confirmations", meaning if a process possesses one of these characteristics, that process is likely to be a good candidate for AR. Characteristics for "negative confirmations" which indicate when a process is unlikely to be a good candidate for AR are not explored in this research.

The synthesized process characteristics are explained as follows:

Process Characteristic #1: Content Enriching

AR is suitable if a process is enhanced when users are provided intuitive ways to aggregate, publish, and interact with content in real-time as anchored to the real-world environment.

For example, a maintenance worker can inspect an aircraft by visually inspecting the aircraft and viewing the aircraft's sensor data aggregated and displayed in an AR dashboard (content aggregation). After the inspection is finished, the worker can add notes and photos to a virtual inspection logbook to record the current state of the aircraft (content publishing) [92] – see Figure 31.

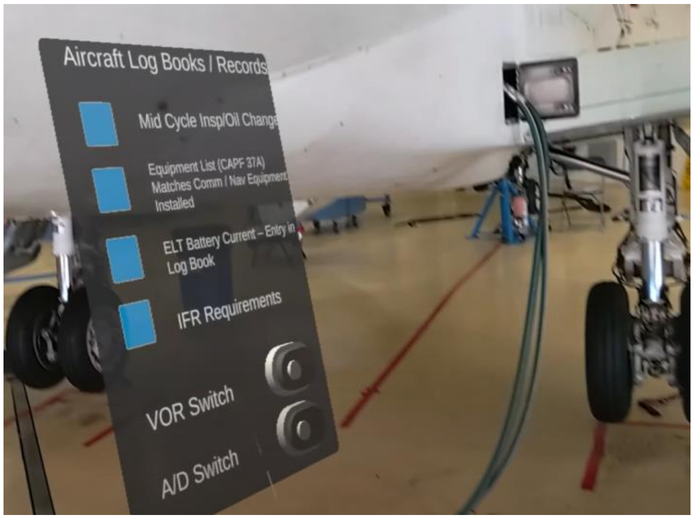


Figure 31. AR for Aircraft Maintenance [92]

Another example is where users can scan an image of a sculpture on an art book and view the virtual 3D model of the sculpture rendered on top of the image for a more engaging user experience [93] – see Figure 32.

Figure 32. CoCa Pop-up (AR)tbook [93]

Process Characteristic #2: Work Instructions

AR is suitable if a process is enhanced when users are provided with step-by-step guidance in 3D for the execution of a complex task.

For example, lab workers use AR-guided work instructions which include step-by-step guidance along with documentations and videos to complete their tasks [94] – see Figure 33.

Figure 33. AR-Guided Work Instructions [94]

Process Characteristic #3: Hard-to-Simulate Environment

AR is suitable if a process is enhanced when users need to work with 3D environments which are otherwise hard to simulate in real life.

For example, traditionally surgeons rely on 2D MRI and CT scans to visualize the patient anatomy in their head so that they can plan for the surgery. With current technologies, 2D MRI and CT scans can be transformed to a 3D model of patient anatomy so that doctors can, for example, assess the contact points of tumors and the patient's surrounding organs and plan for how to best remove the tumors. Furthermore, doctors can walk the patient through the surgery procedure using the 3D model to put the patient at ease [95] – see Figure 34.

Figure 34. AR for Patient Care [95]

Another example is when police use an AR device to scan the crime scene and create a virtual 3D mesh of a crime scene, allowing detectives to later review the virtual crime screen and solve the crime. It would have been more difficult for the detectives to visualize the crime scene only using pictures and police reports compared to viewing the 3D crime screen on an AR device [96] – see Figure 35.

Figure 35. AR for Policing [96]

Another example is the use of AR in constructions where construction managers can view 3D models of the structures of a building (e.g., a pillar) to be built as anchored in a physical space. As the physical structures are getting built, construction managers can quickly confirm if the construction is being carried out as intended by checking if the physical structures match up with the virtual structures [97] – see Figure 36.

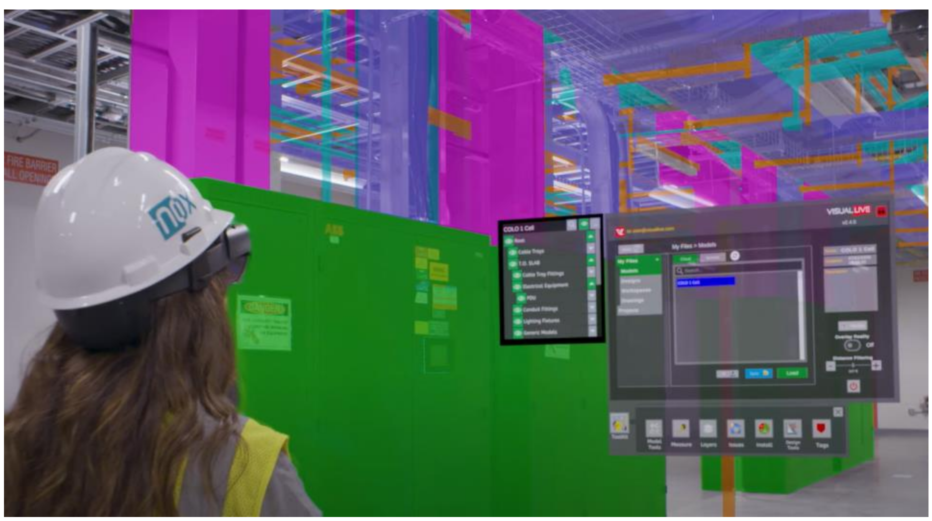


Figure 36. AR for Constructions [97]

Process Characteristic #4: Hands-Free Access to Content

AR is suitable if a process a process is enhanced when users need to access content hands-free while performing a task.

For example, at nuclear power plant, when workers use a glovebox to interact with radioactive materials, they need to scan their hands with a radiation monitor every time they pull their hands out of the glovebox. Therefore, two workers typically work together with one worker reading aloud the work instructions while another worker putting his/ her in the glovebox and performing the task as described in the work instructions.

This is an example where workers can benefit from being able to control an AR device hands-free e.g., using voice commands to pull up the work instructions augmented in user's field-of-view while they keep their hands in the glovebox and continue performing their tasks according to the work instructions [98] – see Figure 37.

The use of AR during surgery is another example of hands-free access control since surgeons can benefit from being to view patient data in their field of view while working both of their hands to carry out the surgery.

Figure 37. Example Glovebox for Nuclear Power Plants[73]

Process Characteristic #5: Navigation

AR is suitable if a process is enhanced when users are guided to find a location or an object.

For example, paper [59] implemented an AR-guided navigation system which allows rescue & emergency personnel to navigate underground mines and rescue victims during an emergency.

Additionally, Google Maps now offers Live View AR Directions where virtual arrows and annotations are rendered on top of the physical street view to guide users, especially at busy intersections [99] – see Figure 38.

[73] https://www.seattletimes.com/business/nuclear-industry-hopes-to-expand-output-with-new-reactors/

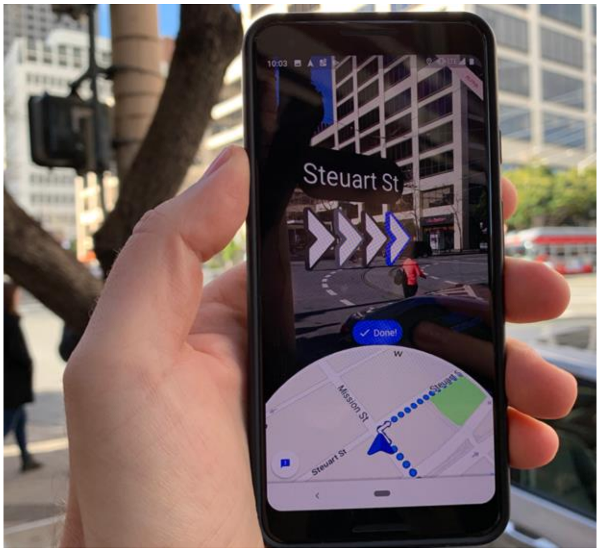


Figure 38. Google Maps Live View AR Directions [99]

Process Characteristic #6: Collaboration

AR is suitable if a process is enhanced when users across geographies need to work together with 3D contents.

For example, when troubleshooting a machine, a local worker initiates a video call with a remote worker for support. Through live video streaming from the local worker's AR device, the remote worker can see the view of the local worker and, for example, pull up relevant documentations and make live annotations to help the local worker complete his/ her task [100] – see Figure 39.


Figure 39. Collaboration in AR [100]

APPENDIX A: AR USE CASE SUMMARY provides a full summary of all selected use cases and explanations for how the use cases illustrate the six characteristics explained above. If a business process possesses at least one of these characteristics, the process is likely to be a good candidate for augmentation with the use of AR. It is up to the project teams to make the final decision whether AR is applicable to their target process.

3.3 RQ#8.2: How to Choose Between WebAR vs. Native AR vs. AR Headsets for a Selected Process? (Design Phase > AR Platform Suitability)

Once the *Discovery* phase is complete, and it is decided that a business process can be and should be enhanced with the use of AR, the *Design* phase commences with the first design point being: RQ#9.2 What AR platform is most suitable for the selected process?

From Table 10 where AR adoption success factors are mapped to high-level selection criteria/ requirements for the ARIM, two high-level selection criteria for AR platform suitability are identified:

1. Usage conditions-specific: Groups all low-level selection criteria which are unique to the environment where AR solutions are deployed and influences the choice of AR platforms.
2. Hardware-specific: Groups all low-level selection criteria which communicate the capabilities and limitations of the hardware on which different AR platforms run.

Next, section Usage Conditions-Specific Selection Criteria and section Hardware-Specific Selection Criteria will decompose the two high-level selection criteria into low-level, concrete selection criteria which can be used to benchmark different AR platforms. These sections also explain the motivation for each of the low-level selection criteria using example use cases.

APPENDIX B: AR PLATFORM DECISION MATRIX benchmarks different AR platforms using all the low-level selection criteria discussed in this section. The objective is to help business stakeholders and developers understand the capabilities and limitations of different AR platforms so that they can choose the platform most suitable for a target process.

Note that the key contribution of this section is a list of selection criteria for choosing a suitable AR platform. Although valuable, section AR Platform Decision Matrix do not represent all available AR platforms and should be kept up to date as AR platforms evolve and expand their capabilities.

Usage Conditions-Specific Selection Criteria

This research proposes the following low-level, concrete selection criteria for benchmarking different AR platforms based on usage conditions:

1. Lighting conditions
2. Network connectivity
3. Noise
4. Operating temperature
5. Operating humidity
6. Drop-safe
7. Water-/ dust-proof
8. Compliance with health/ safety standards
9. Hand usage
10. User interactions
11. Movement restrictions

Criteria #1: Lighting conditions

Since AR tracking engine relies on computer vision algorithms to detect a marker in each video frame, the quality of the video frames captured in different lighting conditions can make it hard or easy for the tracking engine to detect the marker. For example, an AR solution deployed in a dimly lit working condition like an underground mine needs a more robust tracking engine to work properly.

Because different tracking engines support different AR platforms, the decision of choosing a specific AR platform might be driven by how robust the supported tracking engines for that AR platform are.

To the author's best knowledge, there is currently no standard methods for measuring the performance of different tracking engines in different lighting conditions to definitively say, for example, tracking engine [A] performs best between [X] and [Y] lux with [A] being the name of an arbitrary tracking engine, [X] and [Y] being arbitrary numbers, and lux being the unit of illumination.

This research proposes three different approaches for choosing an AR platform best suited to accommodate a target lighting condition:

1. Test different candidate AR platforms in the target lighting condition and choose the most performant AR platform
2. Customize the tracking engine's computer vision algorithm for more robust tracking: This approach might be possible with open-source tracking engines whose source-code is publicly available but not be possible when working with commercial tracking engines which implement their own proprietary computer vision algorithms.
3. Customize the camera's properties to improve the quality of the captured video frames e.g., reduce the camera's exposure level in an overly bright environment to allow less light entering the camera lens as shown in paper [101].

Criteria #2: Network connectivity

Some working environments might have intermittent internet while high-security working environments like nuclear power plants or secret government facilities might not have access to the public internet at all.

In case that case, AR solutions only work if they can support offline mode. For example, like any native application, native AR can run in offline mode. HoloLens can also run AR work instructions created with Microsoft Dynamics 365 Guides[74] in offline mode with some limitations. It might be, however, more challenging to implement offline mode support for webAR.

Criteria #3: Noise

If an AR solution is deployed in a loud environment like a factory shopfloor, it is important that the AR solution offers noise-canceling microphones so that users can still use voice commands to control the AR solution.

[74] https://learn.microsoft.com/en-us/dynamics365/mixed-reality/guides/

Criteria #4-7: Operating temperature, humidity, drop-safe, waterproof, dustproof
Operating temperature, humidity, drop-safe, waterproof, and dustproof selection criteria are intended to measure how well an AR platform is suited for a rugged working condition.

For example, if a researcher in Antarctica wants to use an AR solution in sub-zero temperature, webAR and native AR are likely ill-suited since most mobile devices (phones and tablets) mostly do not work when the temperature drops below zero. The same is true for most AR headsets except for Vuzix M400[75] smart glass which can operate in temperatures as low as -20 Celsius degree.

Criteria #8: Compliance with health and safety standards
Some working conditions impose specific health and safety standards AR solutions need to comply with.

For example, construction workers are required to always wear a hardhat while being on site. HoloLens is one of a very few AR headsets which offer a hardhat-integrated version, making it a strong AR platform to be considered over other AR headsets, native AR, and webAR.

Criteria #9: Hand usage: Hands-free to hand-occupied
This criterion refers to whether users need to control an AR solution hands-free so that they can use both of their hands for completing their task, or they are okay with using one hand to hold up a mobile device and use the other hand to complete their task.

For example, when a surgeon requires the use of AR for easy access to patient's medical data during a surgery, AR headsets stand out as a strong contender since the surgeon can operate the headsets hands-free, allowing them to continue using both of their hands to carry out surgery procedures. This is an advantage that AR headsets have over native AR and webAR where users need to hold a mobile device with one of their hands.

Criteria #10: User interactions
Having intuitive interactions is one of the success factors for AR adoptions as captured in success factor "SF5. User experience during AR usage" in Table 4.

Common types of user interactions include gesture commands (using hand tracking engine), gaze commands (using eye tracking engine), voice commands (using speech recognition engine), headset and handheld controller commands (using headset and handheld controller tracking engine).

Not every AR platform supports all these types of user interactions. For example, only AR headsets support gaze commands since the headsets which sit on the user's head can track the movements of the user's eyes while native AR and webAR which run on a hand-held mobile device cannot. Gaze commands are beneficial in scenarios when users do not have a free hand for gesture commands or when environment is too loud for voice commands.

---

[75] https://www.vuzix.eu/products/vuzix-blade-2-smart-glasses

Criteria #11: Movement restrictions

This criterion refers to whether the end users need wide-range mobility while performing a task. If yes, tethered AR headsets are not ideal as they restrict users' movements.

For a use case where car designers need to review a high-quality 3D model of a car being designed, the designers would want to walk around the 3D model to examine it from different angles. However, no wide-range movements like walking across the entire building are necessary. Being a tethered headset with a robust rendering engine, Varjo XR-3[76] is a great contender because it can render high-fidelity 3D graphics and can accommodate small-range movements.

Hardware-Specific Selection Criteria

This research proposes the following low-level, concrete selection criteria for benchmarking different AR platforms based on hardware capabilities:

1. Battery life
2. Device weight
3. Field of view
4. Camera resolution
5. Display resolution
6. Refresh rate
7. Storage capacity
8. Computation capacity: CPU/ GPU
9. Remote rendering
10. Cross-device development
11. Application download required
12. Cost: hardware + subscription

Criteria #1: Battery life

Battery life of AR platforms is critical for tasks which take a long time to complete. For example, most AR headsets have a battery life around 2-3 hours – see section AR Platform Decision Matrix for detailed benchmarking of different AR platforms. This means if a surgeon wants to use an AR headset for a 5-hour surgery, the surgeon will need to charge the headset between or use multiple headsets.

Criteria #2: Device weight

The heavier the AR platform, the more fatigue the user will experience when holding the mobile device (for native AR and webAR) or wearing the AR headset.

Criteria #3: Field of view (FoV)

FoV is the maximum area that we can observe as measured in degrees. Each human eye can cover approximately 130°, allowing us to observe nearly 180° with both eyes [102]. For AR headsets, the larger the FoV, the larger the area through which we can view virtual contents augmented to the real physical environment and the larger those virtual contents can be.

---

[76] https://varjo.com/products/xr-3/

FoV is not applicable to webAR and native AR since we can only view virtual contents through the mobile device's screen. AR headset providers want to provide bigger and bigger FoV for their headsets so that bigger virtual contents can be viewed. For example, the first HoloLens[77] has a horizontal FoV of 30° while the HoloLens 2[78] has a FoV of 43°.

Different use cases have different requirements for FoV. For example, when using AR for virtual product prototyping, large FoV is preferred for an immersive experience whereas, an average-sized FoV works fine for, say, AR-guided work instructions.

Criteria #4-5: Camera + display resolution

Resolution defines video frames' width and height in pixel. The higher the resolution, the sharper the video looks.

High camera resolution is important for capturing high quality video frames of the physical environment so that AR tracking engines can detect and track physical markers in those video frames. Display resolution refers to the quality of the virtual contents the users can view through the AR applications.

Criteria #6: Refresh rate

Refresh rate as measured in Hz refers to the number of video frames rendered in a second. The higher refresh rate, the smoother an AR experience. Low refresh rate may result in headache or nausea whereas refresh rate of 120+ Hz offers higher video quality but not distinguishable to human eyes [102].

The refresh rate for native AR and webAR depends on the implemented tracking engine, rendering engine, and the mobile device the native AR/ webAR application is running on since the tracking engine and rendering engine rely on the CPU and/ or GPU of the device to process video frames and render virtual objects in each video frame.

The refresh rate for most AR headsets falls between 60-120 Hz – see section AR Platform Decision Matrix for detailed comparison of different AR headsets.

Criteria #7-9: Storage + computation capacity + remote rendering

Storage and computation capacity dictates how much local tracking and rendering work can be done locally on the mobile device or AR headset an AR app runs on.

In Figure 59, engineers can construct a high-fidelity 3D model of a physical bridge from drone images so that they can inspect the virtual bridge for any fractures. Rendering such a 3D model requires significant computing power. To address this challenge, several AR headsets like Varjo XR-3[79] and HoloLens 2[80] offer a feature of running their rendering engine remotely in the cloud and stream the rendered video frames directly to the headsets, making these AR headsets more suitable

[77] https://vr-compare.com/headset/microsofthololens
[78] https://vr-compare.com/headset/microsofthololens2
[79] https://varjo.com/use-center/get-started/varjo-services/varjo-reality-cloud/
[80] https://azure.microsoft.com/en-us/products/remote-rendering/

for this example bridge inspection use case compared to other AR platforms which do not natively support remote rendering.

Criteria #10: Cross-device development

Cross-device development refers to the ability for developers to develop and maintain a single code base for an AR application and deploy that code base across multiple devices. This is more efficient compared to another approach where developers have to develop and maintain multiple code bases, one for each device.

For webAR, most webAR tracking and rendering engines are built upon browser's natively supported features such as WebGL. Therefore, developers can develop an AR application once and deploy it across the recent releases of major browsers like Chrome and Safari on both Android and iOS devices.

For native AR, some tracking engines like Vuforia[81] support both Android and iOS. If developers develop an AR application using Vuforia, the application will work on both Android and iOS devices. However, some tracking engines only support a specific OS. For example, ARCore[82] tracking engine supports only Android, and ARKit[83] tracking supports only iOS.

For AR headsets, there is an emerging standard API called OpenXR[84] – see Figure 40. Before OpenXR, tracking and rendering engines (rendering engines are also called "game engine") need to use a proprietary API for each AR headset. With OpenXR, tracking and rendering engines can access different AR headsets using OpenXR standardized API. This means developers can now develop an AR application once and deploy it across OpenXR-compliant AR headsets.

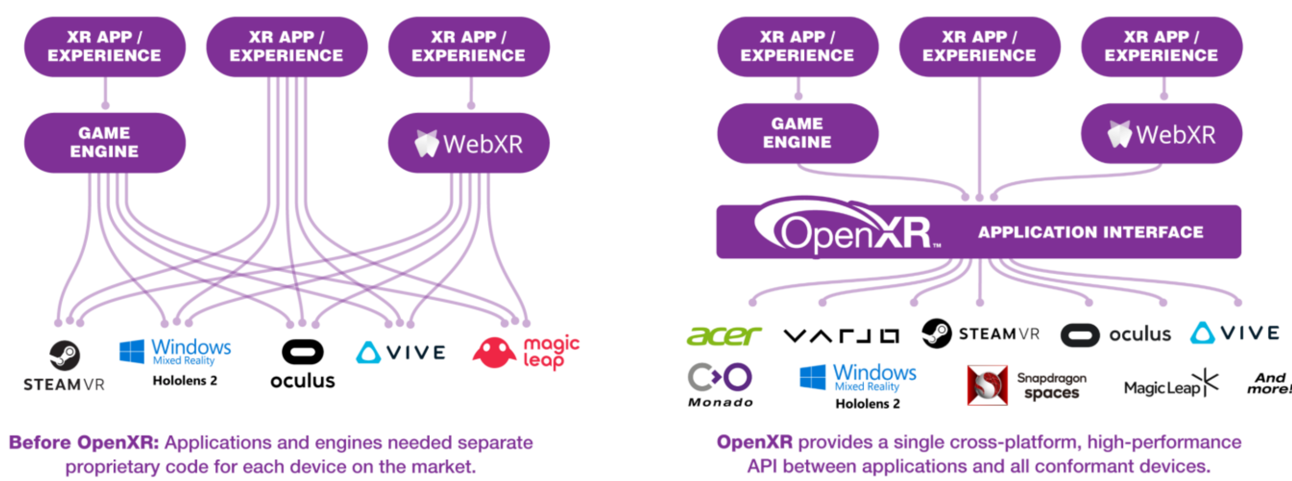


Figure 40. OpenXR (source: https://www.khronos.org/api/index_2017/openxr)

[81] https://library.vuforia.com/getting-started/vuforia-features
[82] https://developers.google.com/ar/develop
[83] https://developer.apple.com/augmented-reality/arkit/
[84] https://www.khronos.org/api/index_2017/openxr

Criteria #11: Application download required

For native AR and AR-on-headset, users need to download AR applications to a mobile phone/ tablet or an AR headset. For webAR, users can run AR applications directly from a browser, making webAR more accessible compared to native AR and AR-on-headset.

For example, if a retail store wants to develop an AR application which allows shoppers to stand in front of their store window, scan the mannequin, and view a 3D video of a catwalk show where the outfit worn by the mannequin is being featured, webAR might be a better fit since shoppers might be hesitant to download and install an application on their phone for this single use.

Criteria #12: Cost: Hardware + subscription

Among all AR platforms, webAR and native AR are more cost-effective compared to AR-on-headset since webAR and native AR can run on existing mobile devices (smartphones and tablets) owned by employees or consumers while AR headsets are often expensive. For example, Microsoft HoloLens 2 costs 3.849€.

Addition to the cost of the headsets, some headsets like the Varjo XR-3[85] also require users to pay for a subscription fee to use the software and services running on the headset.

AR Platform Decision Matrix

APPENDIX B: AR PLATFORM DECISION MATRIX benchmarks different AR platforms (webAR, native AR, and 7 different AR headsets) using the selection criteria explained above.

Note that webAR and native AR each have their own column while there are multiple columns for different AR headsets. This is because webAR and native AR run on mobile phones and tablets which have relatively similar hardware capabilities whereas AR headsets tend to differ significantly in terms of their hardware capabilities.

[85] https://varjo.com/use-center/get-started/varjo-services/varjo-xr-and-vr-subscriptions/

3.4 RQ#8.3: How to Choose Between Different AR Development Tools for a Selected Process and AR Platform? (Design Phase > AR Development Tool Suitability)

After choosing an AR platform, the next step is to choose the appropriate AR development tools (i.e., tracking and rendering engines) which supports the selected AR platform and can fulfill the requirements of the selected business process.

From Table 10 where AR adoption success factors are mapped to high-level selection criteria/ requirements for the ARIM, four high-level selection criteria for AR tracking/ rendering engines are identified:

1. Tracking capabilities
2. Rendering capabilities
3. Ease of development
4. Cost

Section Tracking Capability-Specific Selection Criteria, Rendering Capability-Specific Selection Criteria, Ease of Development-Specific Selection Criteria, and Licensing Cost-Specific Selection Criteria discuss low-level selection criteria for AR development tools and the motivations for each of those criteria.

Section APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX benchmarks several AR tracking and rendering engines (non-exhaustive) using these low-level selection criteria. Note that Tracking Capability-Specific Selection Criteria are only applicable for benchmarking AR tracking engines and Rendering Capability-Specific Selection Criteria are only applicable for benchmarking AR rendering engines while Ease of Development-Specific Selection Criteria and Licensing Cost-Specific Selection Criteria are applicable for all tracking and rendering engines.

Tracking Capability-Specific Selection Criteria

As different use cases require different tracking capabilities, this group of selection criteria is intended to assess which tracking engines support which tracking capabilities, allowing developers quickly identify the tracking engines appropriate for their target use cases.

This research proposes the following selection criteria for tracking engines:

1. Fiducial tracking
2. Image tracking
3. Face tracking
4. Location tracking
5. Object tracking
6. Surface tracking
7. Spatial tracking
8. Other tracking
9. Multi-marker tracking

10. Multi-user AR session
11. Depth tracking
12. Lighting effects

Criteria #1-7 are seven different tracking techniques already discussed in section AR by Common Tracking Techniques of the literature review. Each tracking technique is appropriate for different use cases.

For example, virtual try-on AR applications typically use face tracking technique which, for example, renders a pair of virtual sunglasses on a user's face. Surface tracking technique powers the Ikea Place AR application where users can render a virtual armchair in their living room before making a purchase as shown in Figure 12.

Computer vision algorithms are what power AR tracking engines. As computer vision technology evolves, it is possible that more tracking techniques will be developed to support a wider range of use cases. Therefore, it is important to keep the ARIM updated on new tracking techniques.

Criterion #8 "other tracking" catches all other tracking techniques available in existing tracking engines. For example, ARKit, a tracking engine for native AR on iOS, offers body position tracking[86] so if users move their body, a virtual model of themselves will mimic the move. This body tracking engine does not have its own category like criteria #1-7 because it is still premature and does not yet have strong use cases in the author's opinion.

Criterion #9 "multi-marker tracking" refers to tracking engines which can track the presence of multiple markers in a single video frame so that the rendering engines can render virtual contents as anchored to those markers as shown in Figure 41.

This tracking technique is handy when, say, an art gallery wants to build an AR application where visitors go to a photography exhibition, scan multiple photographs, and view videos about how the artists get the inspirations for the photographs.

Figure 41. Multi-Marker Tracking (source: https://stemkoski.github.io/AR-Examples/)

[86] https://developer.apple.com/documentation/arkit/content_anchors/rigging_a_model_for_motion_capture

Criterion #10 “multi-user AR session” refers to the ability for tracking engines to sync up across multiple devices, allowing multiple users to join an AR session.

For example, Vuforia, a tracking engine for native AR and AR-on-headset, allows multiple users to join an AR session for remote assist[87]. In Figure 42, a local worker (upper left) is calling her remote colleague (upper right) for assistance on how to operate an electrical panel. While the local worker points her device at the panel, the remote worker can view the panel and make virtual annotations circling different buttons to guide the local worker. The local worker can see the virtual circles made by her remote colleague on her phone in real time (lower left is the local worker’s phone, lower right is the remote colleague’s phone).

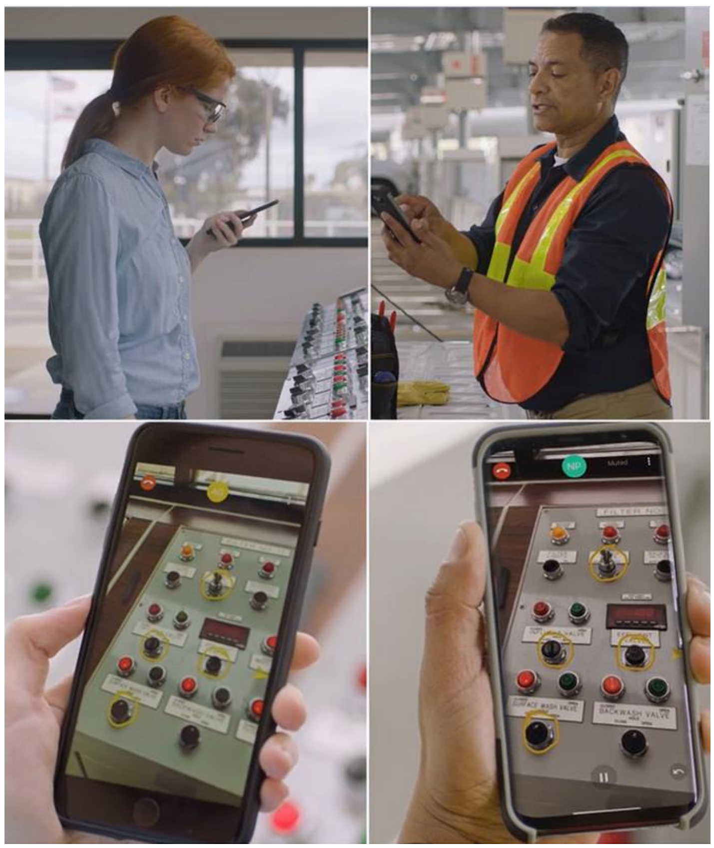


Figure 42. Multi-User AR Session (source: https://www.ptc.com/en/products/vuforia/vuforia-chalk)

Criterion #11 “depth tracking” refers to whether a tracking engine also tracks the depth of the physical objects captured in video frames. Depth tracking is necessary for tracking engines to understand the physical environment so that the tracking engines can instruct rendering engines to occlude a virtual object if it stands behind a physical object, allowing for a more realistic AR experience. In Figure 43, the virtual character on the right is occluded as he is behind a physical door.

---

[87] https://www.ptc.com/en/products/vuforia/vuforia-chalk

Figure 43. Occlusion[88]

Criterion #12 “lighting effects” refers to special capabilities of tracking engines which can enable a more realistic AR experience.

For example, ARCore, a tracking engine for native AR on Android, can analyze the captured video frames and infer the current lighting condition. ARCore then instructs rendering engines to shade virtual objects accordingly. In Figure 44, the virtual cat on the right side is sitting behind a door, so it is rendered as darker than the virtual cat on the left side.

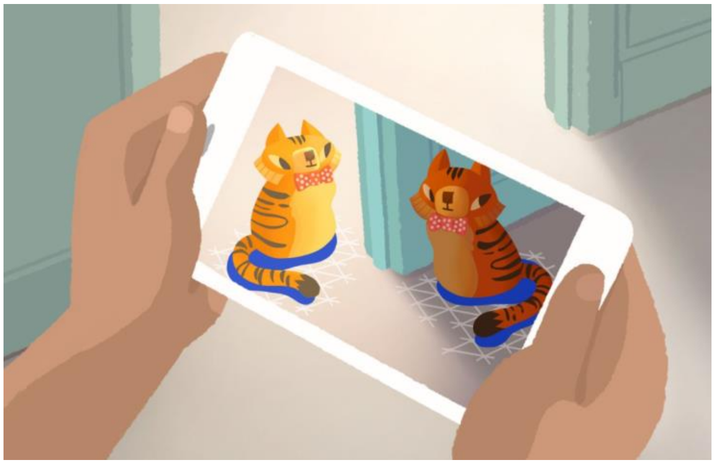


Figure 44. Light Estimate[89]

Another example is realistic reflection effect by ARKit, a tracking engine for native AR on iOS. In Figure 45, a virtual, metallic ball sitting next to a physical cup shows a reflection of the surrounding.

---

[88] https://lightship.dev/docs/ardk/context_awareness/depth/occlusions.html#doxid-occlusions
[89] https://developers.google.com/ar/develop/lighting-estimation

Figure 45. Realistic Reflections[90]

Rendering Capability-Specific Selection Criteria

This group of selection criteria is intended to assess how realistic of an AR content a rendering engine can produce.

This research proposes the following selection criteria for rendering engines:

1. Light estimate
2. Occlusion

Criterion #1 "light estimate" refers to the ability of rendering engines to accept light estimate data produced by tracking engines and properly shade virtual objects to comply with the actual lighting condition as shown in Figure 44.

Criterion #2 "occlusion" refers to the ability of rendering engines to accept depth sensor data produced by tracking engines and properly occlude virtual objects behind physical objects as shown in Figure 43.

Note that these rendering capability-specific criteria "light estimate" and "occlusion" might seem to be duplicates of the tracking capability-specific criteria "depth tracking" and "lighting effects", but they serve different purposes. Tracking capability-specific criteria are intended to benchmark tracking engines while rendering capability-specific criteria are intended to benchmark rendering engines.

For example, suppose a tracking engine has "depth tracking" capability, the tracking engine still requires a rendering engine which has "occlusion" capability so that the rendering engine can understand the depth sensor data collected by the tracking engine and renders virtual contents with occlusion effects.

---

90 https://developer.apple.com/documentation/arkit/camera_lighting_and_effects/adding_realistic_reflections_to_an_ar_experience

Ease of Development-Specific Selection Criteria

This group of selection criteria is intended to assess how easy it is to use an AR development tool.

This research proposes the following selection criteria for assessing the ease of using AR development tools:

1. Integration with other AR development tools
2. Simulation testing
3. Low-code/ no-code development

Criterion #1 "integration with other AR development tools"
Each of the AR development tools assessed in APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX is classified as either a tracking engine, a rendering engine, or both.

Since both a tracking engine and a rendering engine are required to run an AR application, the tracking engine and rendering engine themselves need to integrate with each other. Therefore, this criterion helps assess which tracking engines integrate with which rendering engines.

Criterion #2 "simulation/ remote testing"
When creating an AR application, developers build source code into machine-executable code, run the executable code on a device (mobile phone, tablet, or AR headset), and use live camera feed to test if the application works as intended. This process takes time and slows down developers.

Simulator testing is a feature supported natively some by tracking/ rendering engines which allows developers to use a pre-recorded video to test the application right from the tracking/ rendering engine. This allows developers to test the application faster without having to run the application on a physical device.

Criterion #3 "low-code/ no-code development"
Low-code/ no-code AR development tools allow business users with little to no programming knowledge to create new AR experience like AR work instructions.

However, low-code/ no-code AR development tools typically support only simple AR functionalities. For developing more complex AR functionalities, it is ideal to involve experienced developers.

Licensing Cost-Specific Selection Criteria

There is one selection criterion in this group: Is an AR development tool open source? This criterion is helpful for project teams who would like to use use open-source tools and keep the budget low.

AR Development Tool Decision Matrix

APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX consolidates all AR development tool selection criteria presented in this section and benchmarks 18 different AR development tools (A) categorized as either a tracking engine, a rendering engine, or both and (B) categorized by the AR platforms the tool supports.

3.5 RQ#8.4: What Are the Requirements for the Development of AR Applications? (Development Phase)

From Table 10 where AR adoption success factors are mapped to high-level selection criteria/ requirements for the ARIM, one high-level requirement for the development of AR solutions is identified: The AR solutions should have a flexible architecture for maintainability, agility, and time-to-market.

This high-level requirement is decomposed into two low-level requirements:

1. Microservice architecture
2. Agile development

Microservice Architecture

There are two primary architectural patterns for a software program: 1) a monolithic architecture and 2) a microservice architecture. Each has its own advantages and disadvantages.

Monolithic architecture refers to the practice of developing a software program as a unified unit with self-contained functionalities. These functionalities refer to client-side user interface, server-side operations, business logic, database layer, and integrations [103]. The primary advantage of a monolithic architecture is "fast development speed due to the simplicity of having an application based on one code base", which is a trade-off for flexibility when the application grows in complexity [104].

In contrast to monolithic architecture, microservice architecture relies on a series of independent services with each service responsible for one functionality. The key advantage of this approach is flexibility and agility where smaller teams can work independently on different services and ship them out faster. Microservice architecture also presents some disadvantages including operational overhead since teams working on independent services need to communicate and collaborate to coordinate updates and interfaces [104].

Until 2008, Netflix application was architected as a monolith. Since then, Netflix has embarked on a journey to transition to a microservice architecture and break up its application into over 700 microservices [103].

For the above reasons, this research recommends that AR solutions should be implemented using microservice architecture for flexibility, agility, and time-to-market. For example, an AR solution might be first developed to display temperature sensor data of some machines. Later, another service can be developed and integrated with the existing AR solution to query and display humidity sensor data. Microservice architecture allows the temperature sensor data service and the humidity sensor data service to work independently.

Agile Development

Agile is a project management and software development methodology for delivery of high-quality work products. The Agile manifesto is comprised of four core values [105]:

1. Individuals and interactions are more important than processes and tools
2. Working software is prioritized over comprehensive documentation
3. Customer collaboration is important for delivering the right software and should be prioritized over contract negotiations
4. Changes are welcome even late in the development process since development will be performed in small iterations.

Agile and Design Thinking align in the sense that working prototypes should be delivered and tested frequently with users to gather early feedback and ensure that the application being developed continues meeting user needs. For this reason, this research recommends Agile for the development of high-quality AR solutions.

From the refined functional and non-functional requirements collected in the Design phase (see section Refine Process Requirements), developers can capture functional requirements which dictates how an application should behave as user stories and non-functional requirements as acceptance criteria for relevant user stories.

An example functional requirement captured as a user story is: As a user, I can click on button "View video" to see a video explaining how I should inspect machine A. An example non-functional requirement for this user story is: The video should load within 2 seconds.

The user story is only accepted as complete if clicking the button opens the video, and the video loads within 2 seconds.

3.6 RQ#8.5: What Are the Requirements for the Deployment of AR Applications? (Deployment Phase)

From Table 10 where AR adoption success factors are mapped to high-level selection criteria/ requirements for the ARIM, four high-level requirements for the development of AR solutions is identified:

1. Continuous integration and continuous delivery (CI/ CD)
2. Design for cost optimization
3. Design for performance
4. Design for security

Note that cost optimization, performance and security are complex topics which deserve to be a research topic on their own. Therefore, this research does not attempt to offer detailed guidance for how to design cloud architecture for cost optimization, performance, and security.

Rather, this research consolidates the design principles for cost optimization, performance, and security proposed by three leading cloud vendors – AWS, Google Cloud, and Microsoft Azure – in their respective cloud design frameworks to provide professionals with a quick reference. Additionally, CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9) discusses the implementation of a webAR application to demonstrate how the proposed ARIM can be used in practice. The cloud architecture implemented for this webAR application will feature some configurations materialized from the design principles consolidated in this section to illustrate how the principles are translated to actual cloud architecture design.

Continuous Integration and Continuous Delivery (CI/ CD)

As stated in Table 10, implementing CI/ CD allows developers to work more efficiently by automating the build, test, and deployment process of software applications. Having a good CI/ CD pipeline also enables developers to reap the benefits of microservice architecture since teams can set up their own CI/ CD pipelines for the services they are responsible for so that they can build, test, and deploy the services faster [106].

Design for Cost Optimization

Table 11 consolidates design principles for security as proposed by AWS, Google Cloud, and Microsoft Azure.

#	Design Principle Description	AWS Well-Architected Framework [107]	Google Architecture Framework [108]	Microsoft Azure Well-Architected Framework [109]
1	Encourage employees to consider cost impact when using cloud resources	Principle #1: Implement cloud financial management	Principle #1: Adopt and implement finops	
2	Pay for only what you need and avoid idle resources	Principle #2: Adopt consumption model		Principle #1: Choose the correct resources Principle #3: Dynamically allocate and de-allocate resources Principle #4: Optimize workloads, aim for scalable costs.
3	Monitor and manage costs	Principle #5: Analyze and attribute expenditure	Principle #2: Monitor and control cost Principle #3-7: Strategies for cost optimization for compute, storage, database and analytics, networking, and cloud operations.	Principle #2: Set up budgets and maintain cost constraints. Principle #5: Continuously monitor and optimize cost management
4	Move non-sensitive workloads to public cloud to offload operational burdens for managing your own data centers	Principle #4: Stop spending money on undifferentiated heavy lifting		
5	Manage cloud costs by measuring the business output associated with the costs	Principle #3: Measure overall efficiency: Measure the business output and the cloud cost of delivering such output		

Table 11. Cost Optimization Design Principles

Design for Performance

Table 12 consolidates design principles for performance as proposed by AWS, Google Cloud, and Microsoft Azure.

#	Design Principle Description	AWS Well-Architected Framework [110]	Google Architecture Framework [111]	Microsoft Azure Well-Architected Framework [112]
1	Offload operational burdens of managing your own infrastructure to cloud providers and allow your teams to focus on value-added tasks.	Principle #1: Make advanced technology implementation easier for your team by delegating complex tasks to your cloud vendor. Principle #3: Use serverless architectures		
2	Deploy your applications in multiple regions to allow for lower latency and better user experience.	Principle #2: Go global in minutes		
3	Design a capacity model for your services and choose the right resources and the right size.	Principle #4: Experiment more often Principle #5: Consider mechanical sympathy: Use the technology approach that aligns best with your goals	Principle #1: Define performance requirements Principle #2: Design and deploy your applications	Principle #1: Design for horizontal scaling
4	Run performance testing early in the development process to detect potential issues early.			Principle #2: Shift-left on performance testing
5	Monitor the workload performance to identify bottlenecks and opportunities for improvement.		Principle #3: Monitor and analyze performance. Principle #4: Optimize performance	Principle #3: Continuously monitor for performance in production

Table 12. Performance Design Principles

Design for Security

Table 13 consolidates design principles for security as proposed by AWS, Google Cloud, and Microsoft Azure.

#	Design Principle Description	AWS Well-Architected Framework [113]	Google Architecture Framework [114]	Microsoft Azure Well-Architected Framework [115]
1	Implement the principle of least privilege and zero-trust zone where each interaction between resources must be authenticated and authorized.	Principle #1: Implement a strong identity foundation.		Principle #2: Automate and use least privilege.
2	Implement security measures at different layers (networking, load balancers, compute/ storage resources, etc.) in case the security measures at one layer fails.	Principle #3: Apply security at all layers.	Principle #1: Build a layered security approach. Principle #2: Design for secured decoupled systems.	Principle #1: Plan resources and how to harden them. Principle #5: Identify and protect endpoints.
3	Classify data by their security requirements and protect them accordingly at rest and in transit	Principle #5: Protect data in transit and at rest. Principle #6: Keep people away from data to avoid mishandling and accidental modification/ human error.	Principle #5: Meet the compliance requirements for your region. Principle #6: Comply with data residency and sovereignty requirements.	Principle #3: Classify and encrypt data.
4	Detect and mitigate code-level vulnerabilities		Principle #7: Shift security left and test for security issues early in the deployment pipelines.	Principle #6: Protect against code-level vulnerabilities.
5	Implement security best practices such as using infra-as-code tools to version control cloud resource configurations to avoid configuration drift.	Principle #4: Automate security best practices	Principle #3: Automate deployment of sensitive tasks.	
6	Automate monitoring the cloud environment and incident responses.	Principle #2: Enable traceability. Principle #7: Prepare for security events.	Principle #4: Automated security monitoring	Principle #4: Monitor system security, plan incident response Principle #7: Model and test against potential threats.

Table 13. Security Design Principles

3.7 ARIM Final Design

Figure 46 consolidates section 3.1-3.6 into the final design of the proposed ARIM

3.8 Chapter Summary

This chapter documents the design of the proposed ARIM, a key deliverable of this thesis.

Section 3.1 RQ#8: What Does an Industry-Agnostic, Tool-Agnostic AR Implementation Methodology (ARIM) Look Like? explains key inputs of the ARIM and how they shape the high-level design of the ARIM as shown in Figure 30. The key inputs are:

1. Software development lifecycle (SDLC)
2. Design Thinking methodology
3. AR adoption success factors
4. Technical understanding of AR and cloud infrastructure
5. A webAR application created by the author

The ARIM mimics a 10-phase SDLC proposed by the US Department of Justice since AR applications are software in essence. Additionally, the ARIM incorporates insights from the Design Thinking methodology and AR adoption success factors from existing literature so that AR solutions created using the ARIM can achieve optimal user acceptance.

Specifically, AR adoption success factors are mapped to high-level selection criteria and requirements of the ARIM – see Table 10. These high-level selection criteria/ requirements are contextualized with example use cases and decomposed into low-level selection criteria/ requirements in section 3.2-3.6.

To illustrate how the ARIM can be put in practice, the author developed a webAR application – see CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9). To be precise, the ARIM and the webAR application were created in parallel as one informs the other. The ARIM guides the author along the creation process of the webAR application while insights gained from working on the webAR application are used to finetune the ARIM.

Finally, Figure 46 illustrates the final design of the ARIM.

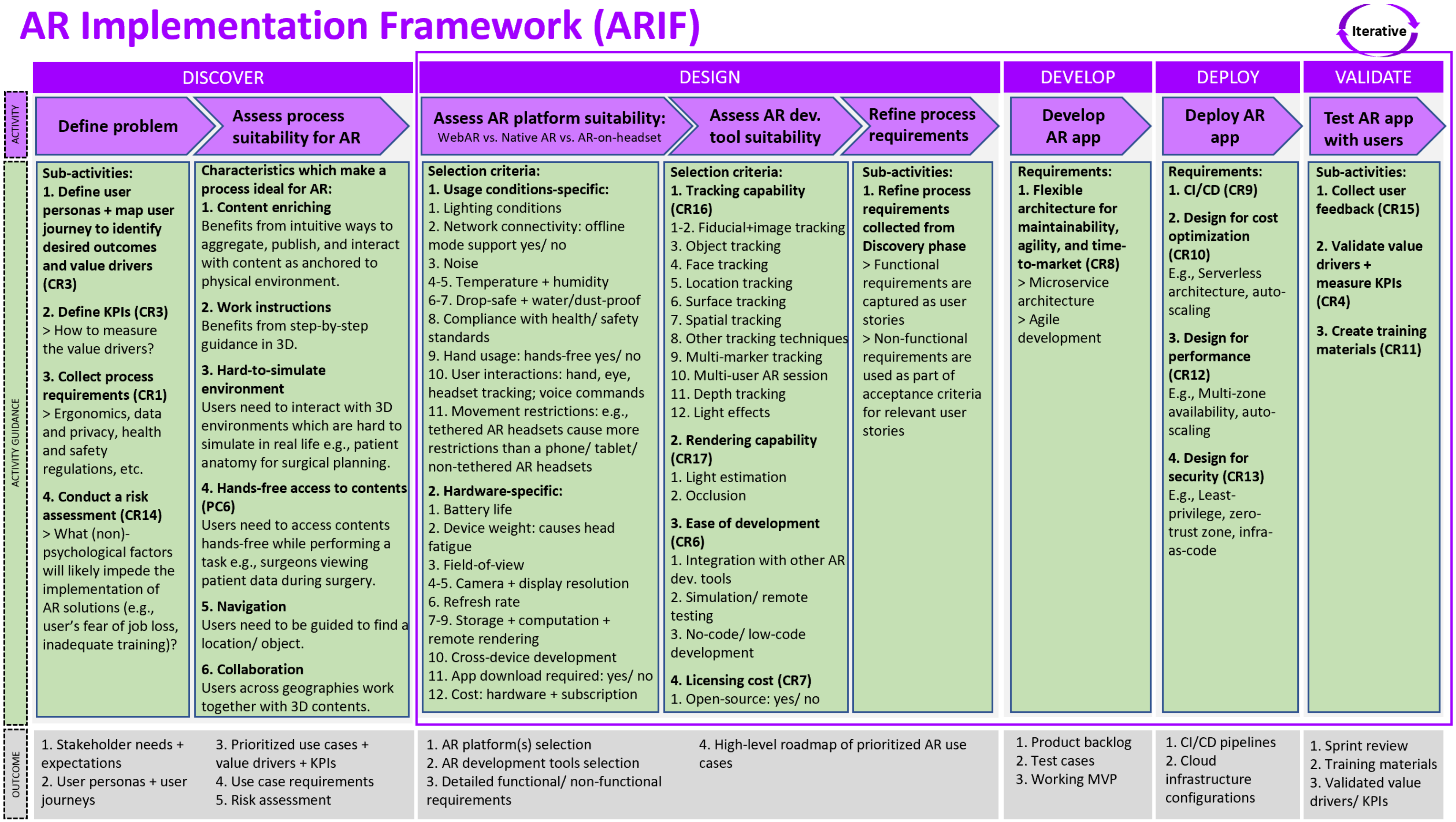


Figure 46. ARIM Final Design

CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9)

4.1 Use Case Introduction

From section 1.3 Research Objective, Research Questions (RQs), and Research Methodology, RQ#10 (demonstration question) states: How can the proposed ARIM be used in practice?

To answer the research question, this chapter illustrates an example use case where a webAR application is implemented to help plant owners better care for their houseplants. The implementation of this webAR application is guided by the proposed ARIM to illustrate how the ARIM can be used in practice as documented in section 4.2 Discovery Phase, 4.3 Design Phase, 4.4 Development Phase, 4.5 Deployment Phase, and 4.6 Validation Phase.

Additionally, the author uses insights gained during the implementation to finetune the ARIM.

Fast forward to the end of the implementation, Figure 47 is the wireframe of the webAR application. Note that the wireframe might look slightly different from the final implementation.

When opening the webAR application in browser, users enter a full-screen camera mode where they can see their houseplant captured in a live camera feed. When scanning a physical image marker attached near the house plant, an AR dashboard pulls up and allows users to:

1. Record the health status of the plant every time the users water them
2. View the watering and health status history
3. Get the updated soil moisture level to determine if the plant needs water
4. View more information about the plant and how to care for it.

A live demonstration of the application will be conducted at the thesis presentation.

In a larger sense, this webAR application is also applicable in an industrial setting where like plant owners, shop floor operators want to better monitor their machines by keeping track of the health status of machines (for example, by checking the machines' sensor data) while visually inspecting the machines and detecting any early malfunctions.

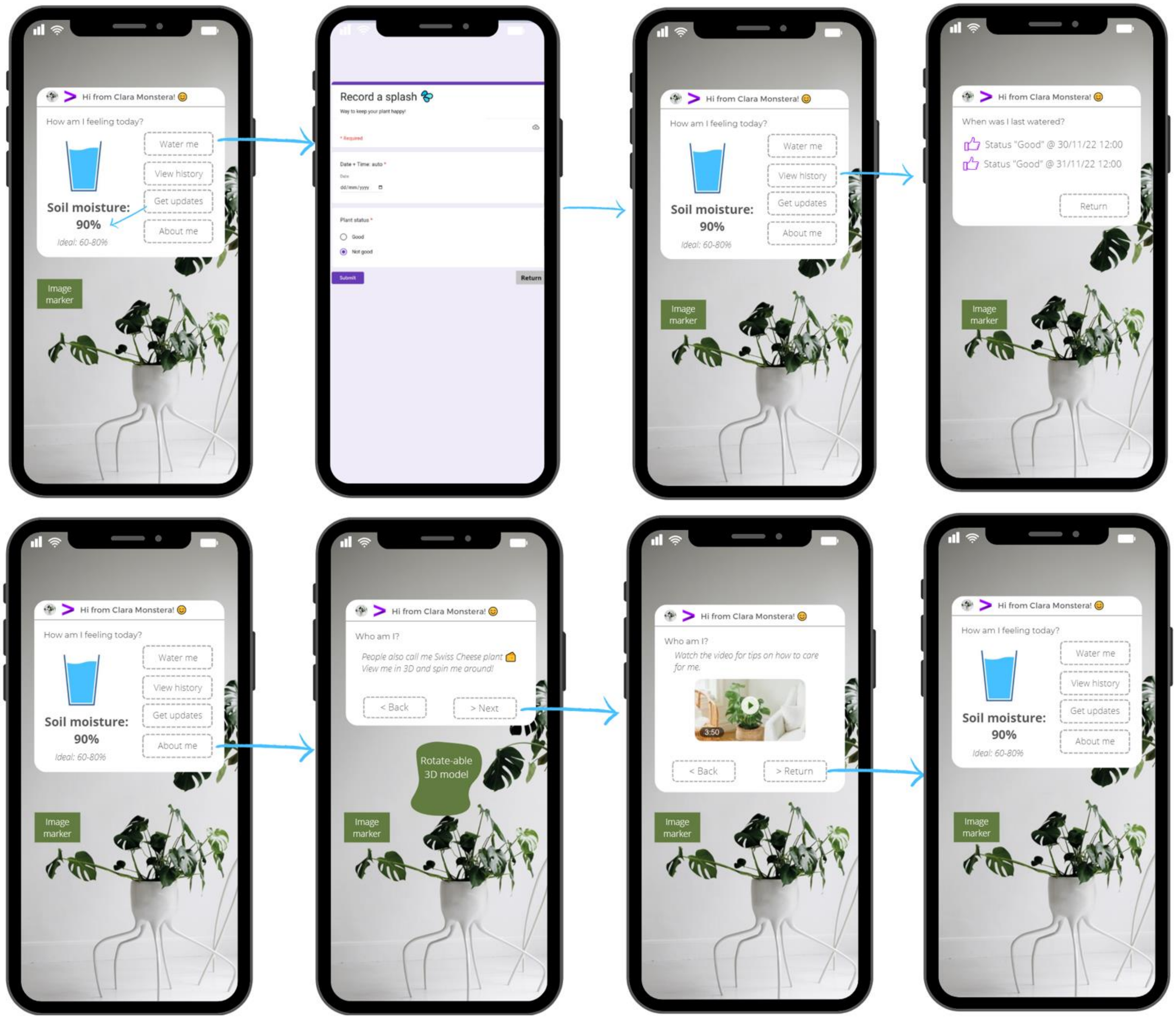


Figure 47. WebAR Application Wireframe

4.2 Discovery Phase

While babies cry and grown-ups walk themselves to the kitchen when they get hungry, plants cannot do that. It is often the case where plant-owners over-water or under-water their houseplants because they are not sure how much water their plants need and how to best care for them. The situation gets dire when the owners are out of town and the plants are left with a stranger who is babysitting the plants and even less unfamiliar with the needs of the plants.

In the Discovery phase, we try to understand plant owners' needs and investigate the possibility of using AR to help plant owners better care for their houseplants.

Activity: Define Problem

Project teams can conduct workshops and brainstorming sessions to understand the stakeholders' needs and expectations. The outcomes of this activity are described below.

Outcome: Stakeholder's needs and expectations

To better care for their plants, plant owners need to know the following:

- When to water their plant: How much water does a specific plant need? When was the plant last watered? Does the plant need water now?
- How to care for their plant: What is the plant starting to have brown leaves?
- How to instruct a house sitter when to water and how to care for the plants while the plant owners are away.

Additionally, suppose one of the expectations of the plant owners is to keep the cost of the proposed solutions as low as possible.

Outcome: User personas and user journey

There are two user personas:

1. Plant owners who are familiar with the plants.
2. House sitters who help take care of the plants while the plant owners are away and are less familiar with the plants

Each user persona has their own user journey as captured in Figure *48*.

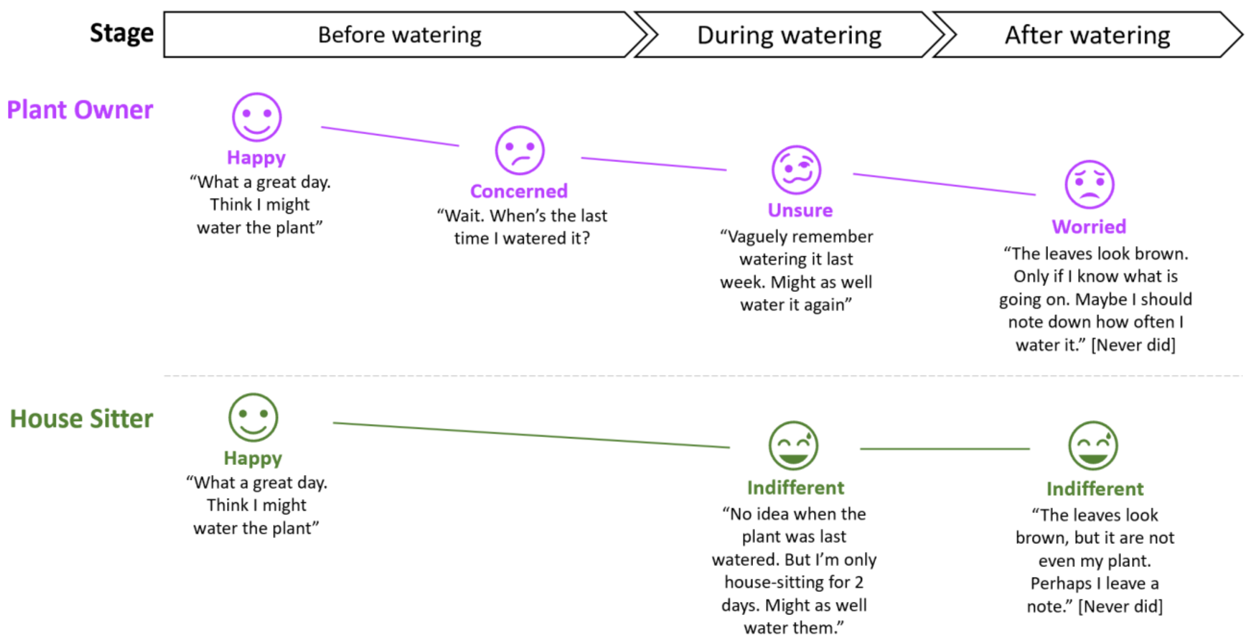


Figure 48. User Persona and User Journey

Outcome: Use case requirements

Based on the understanding of stakeholders' needs and expectations, user personas, and user journeys, below are the requirements of the use case:

1. Plant owners and house sitters can view the history of when the plant was water last
2. Plant owners and house sitters can decide if the plant needs water now by checking the moisture level of the soil
3. Plant owners and house sitters can record every time the plant is watered and the status of the plant at the time (if the plant is in good shape or not)
4. Plant owners and house sitters can view additional information about how to care for the plant.

Outcome: Risk assessment

The main risk associated with this use case is that the plant owners and house sitters view inaccurate records of watering history and/ or soil moisture sensor data, resulting in under- or over-watering.

Activity: Assess Process Suitability for AR

Based on the understanding of the stakeholders' needs and expectations, project teams can create a list of prioritized use cases and assess whether AR is suitable for these use cases. The outcomes of this activity are described below.

Outcome: Prioritized use cases + value drivers + KPIs + use case suitability for AR assessment

The proposed ARIM itemizes six characteristics which make a process ideal for AR:

1. Content enriching
2. Work instructions
3. Navigation
4. Collaboration
5. Hard-to-simulate environment
6. Hands-free control

The use case requirements align with the characteristic "Content enriching."

For example, plant owners and house sitters can scan the plant to view an AR dashboard which aggregates different content about the plant such as the current soil moisture level, the watering history, and additional information about how to best care for the plant. Additionally, the AD dashboard allows users to conveniently push new content: Every time the users view soil moisture data on the dashboard and decide to water the plant, the users can also add a new record which indicates the date, time, and the health status of the plant (good/ bad) as observed by the users.

A value driver for an AR solution for this use case is that it allows plant owners and house sitters to quickly access and interact with content which would have been dispersed in multiple sources (sensor data, watering history, information about the plant) in an engaging and intuitive way, allowing them to better care for the plant.

This value driver can be measured in different KPIs such as the time it takes for users to access and publish these contents (soil sensor data, watering history, other information about the plant) with and without the use of the proposed AR dashboard.

Figure 49 is the revised user journey to reflect how the proposed AR solution will fulfill the needs of plant owners and house sitters.

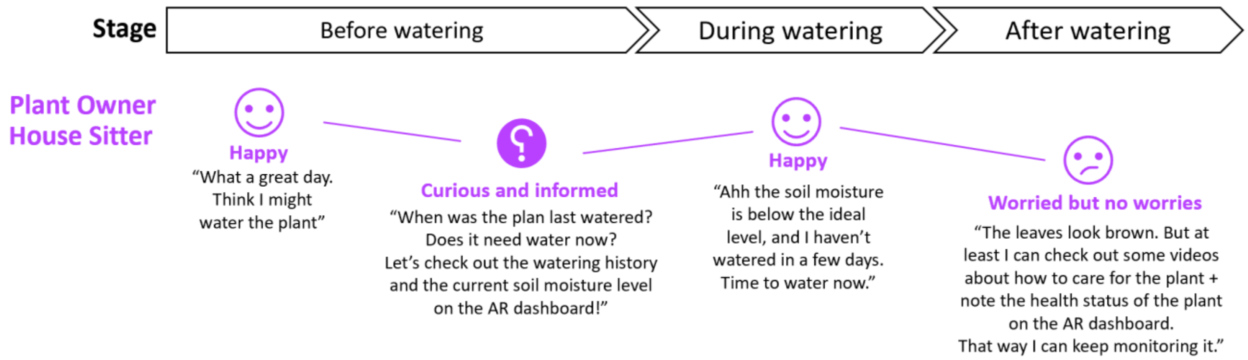


Figure 49. Revised User Journey

4.3 Design Phase

Once deciding that an AR dashboard will be helpful for this use case, we now enter a series of iterations with each iteration comprised of the Design, Development, Deployment, and Validation phase.

Activity: Assess AR Platform Suitability

In the Design phase, we first need to choose an AR platform and development tools for developing this AR dashboard.

Outcome: AR platform selection

The ARIM proposes choosing an AR platform based on two dimensions:

1. Usage conditions-specific selection criteria
2. Hardware-specific selection criteria

There is special no usage conditions required for this use since the plant will be sitting in a residential house with a standard environment in terms of lighting condition, temperature, etc.

In terms of hardware, it would be ideal if a house sitter who infrequently visits do not have to download an app to view the dashboard. Since different house sitters might use different types of phones, it would be great if the AR dashboard runs on both Android and iOS devices.

By cross-referencing APPENDIX B: AR PLATFORM DECISION MATRIX of the ARIM, webAR stands out as a strong contender.

Activity: Assess AR Development Tool Suitability

This activity aims at selecting suitable AR development tools (i.e., tracking and rendering engines) for the selected process and AR platform.

Outcome: AR development tool selection
Now that webAR is chosen as the target AR platform, we now need to choose appropriate AR development tools.

In the Discovery phase, one of the expectations of the plant owners is to keep the cost of the proposed solution as low as possible. Therefore, we decide to use only open-source tools to develop the AR dashboard.

By cross-referencing APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX of the ARIM, several options stand out.

For tracking engine, we decide to use MindAR.js because it supports image tracking technique which should work if we attach an image marker close to the plant so that plant owners and house sitters can easily scan the image to pull up the AR dashboard. MindAR.js is chosen over another open-source alternative AR.js since MindAR.js appears to be more well-maintained with regular updates: https://github.com/hiukim/mind-ar-js repository shows 8 releases compared to 4 releases for https://github.com/AR-js-org/AR.js repository.

For rendering engine, we decide to use three.js because it appears to be more flexible than another open-source alternative A-frame and has a bigger support community. https://github.com/mrdoob/three.js repository has 141 releases compared to 26 releases for https://github.com/aframevr/aframe repository. Furthermore, because A-frame is an abstraction on top of three.js, which means anything that can be done in A-frame can also be done in three.js, but not everything that can be done in three.js can be done in A-frame. Simply put, three.js offers more flexibility than A-frame.

Activity: Refine Process Requirements

This activity aims at finetuning the use case's requirements, which will be the input for the developers to program the AR solution.

Outcome: Detailed functional and non-functional requirements
The process requirements (i.e., use case requirements) collected from the Discovery phase are furthered refined and categorized into functional and non-functional requirements. Note that requirement gathering is an on-going activity as the project evolves.

Below are the functional requirement(s) of the proposed AR dashboard:

1. As a user, I want to be able to view the current soil moisture data of the plant so that I can decide whether the plant should be watered now
2. As a user, I want to be able to record a new watering event along with the plant's current health status after I've watered a plant
3. As a user, I want to be able to view the watering history of that plant along with the records of the plant's health status (good/ bad)
4. As a user, I want to be able to review additional information about how to care for the plant in the form of text and videos.

Outcome: High-level roadmap
A high-level roadmap sets the direction for the project.

For example, in the first stage, we can aim for quick wins such as incorporating the soil sensor data to the AR dashboard as it offers immediate insights into whether the plant needs water right now.

In the later stage, we can expand the AR dashboard to include advanced functionalities such as using computer vision algorithms to detect if the plant has any disease based on the color of the leaves as a user opens the AR dashboard and captures the plant on live camera feed, render the result in the dashboard for the user to view, and automatically record the result as the current health status of the plant.

4.4 Development Phase

Using the requirements defined and the AR development tools selected in the Design phase, developers carry out the actual coding work to build the proposed AR solution and produce the following outcomes.

Outcome: Product backlog + test cases
The functional and non-functional requirements along with other tasks (e.g., bug fixes) are prioritized in a product backlog. For each iteration (called “sprint” in Agile), each developer is responsible for certain items from the product backlog.

New items can be added to the backlog to reflect, for example, new requirements proposed by the users – a core value of Agile where changes are welcome even late in the development process.

Developers write functional and non-functional test cases to confirm that the software is working as intended.

When working on the webAR application, the author uses Microsoft Azure DevOps[91] for requirements management and testing because it is the preferred platform for Accenture’s internal projects.

Outcome: Working MVP
In practice, when an Agile team plans for a sprint, they choose to work on items from the product backlog so that at the end of the sprint, there will be a working software (also called “MVP” or minimum viable product) so that users can test it and give feedback.

4.5 Deployment Phase

Outcome: CI/CD pipelines
In addition to requirements management and testing capabilities, Microsoft Azure DevOps also allows developers to set up CI/ CD pipelines so that once developers commit a change to a

---

[91] https://azure.microsoft.com/en-us/products/devops/

source code repository, the pipelines will automatically run build (i.e., transforming source code to machine-executable code called "build artifacts"), deploy the build artifacts to a test environment, run test on build artifacts, and deploy the build artifacts to production environment if the tests passed.

Outcome: Cloud infrastructure configurations

Figure 50 is the final cloud architecture of the webAR application.

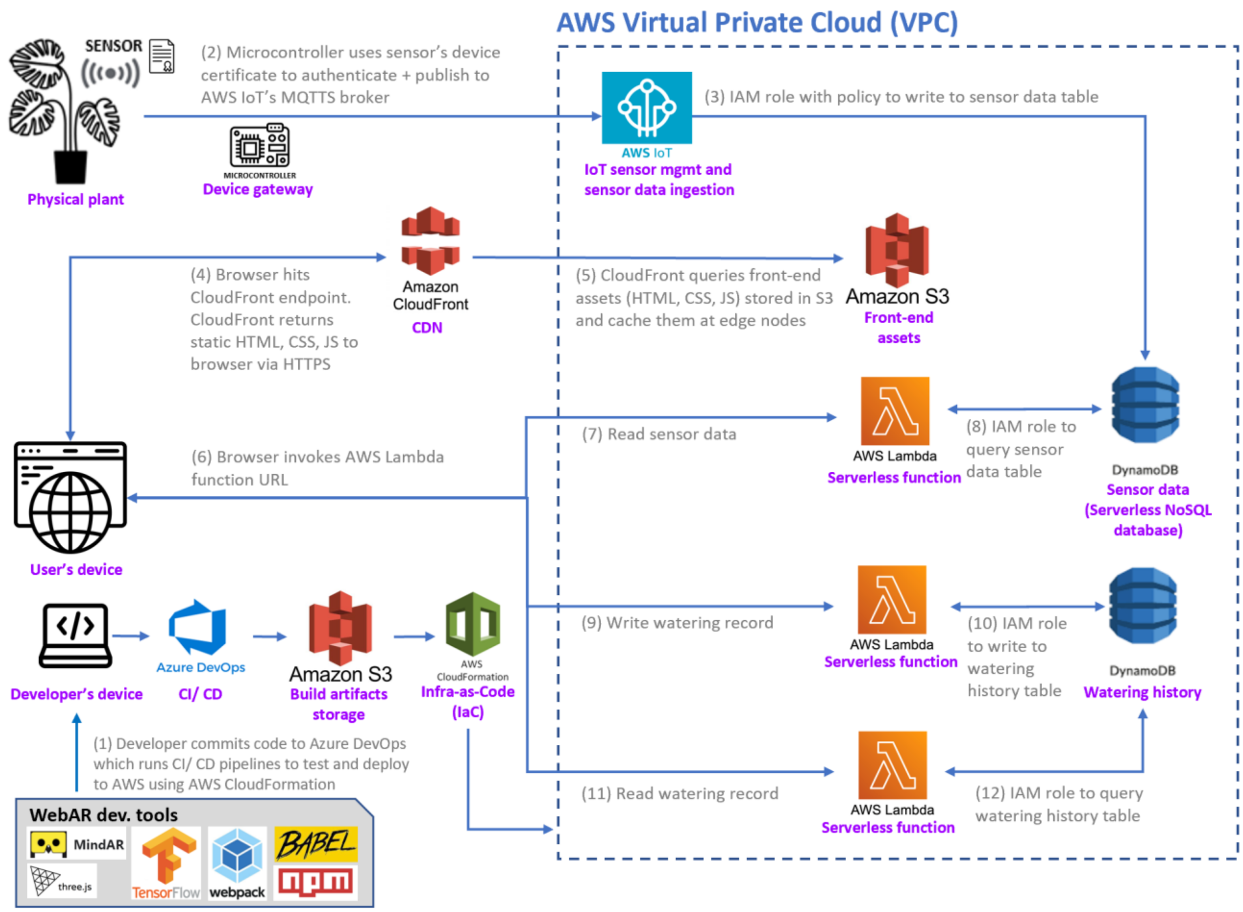


Figure 50. WebAR Application's Cloud Infrastructure

Table 14 discusses technical details of the webAR application's cloud architecture to demonstrate how the concept of CI/ CD and the design principles for cost optimization, performance, and security discussed in section 3.6 RQ#8.5: What Are the Requirements for the Deployment of AR Applications? (Deployment Phase) are translated to actual cloud architecture design decisions. This table elaborates on the 12 activities captured in Figure 50.

#	Activity	Description
1	Developer commits code to Azure DevOps. Azure DevOps runs CI/ CD pipelines to test code and deploy to AWS resources using AWS CloudFormation	To develop the webAR application, the author uses *MindAR.js*[92] as the tracking engine, *Three.js*[93] as the rendering engine along with tools including tensorflow, webpack, babel, and npm. The source code of the application is stored in *Azure DevOps*. Once the author commits new code to Azure DevOps, the CI/ CD pipelines configured in Azure DevOps are automatically triggered to build source code into machine-executable code (called "build artifacts") and deploy them on AWS resources. For this deployment to happen, in the background, Azure DevOps' CI/ CD pipelines push the build artifacts to a *AWS S3* bucket from which *AWS CloudFormation*, an infrastructure-as-code tool, pulls the artifacts from and use them to provision necessary AWS resources such as DynamoDB database and AWS Lambda. Infrastructure-as-code (IaC) refers to the practice of defining the configurations of infrastructure resources as code. For example, when deploying a database instance, we need to specify several parameters such as the geographic region the instance runs in, its disk capacity in GB, etc. If we use a configuration file to capture the configurations of the database, every time we provision a new database instance using the configuration file, we should get an identical instance. IaC aligns with the security design principle "Implement security best practices such as using infra-as-code tools to version control cloud resource configurations to avoid configuration drift" in Table 13.
2	Microcontroller uses sensor's device certificate to authenticate + publish to AWS IoT's MQTT broker	The house plant we want to monitor using the AR dashboard is fitted with a soil moisture sensor. The sensor is registered with *AWS IoT*, a platform for sensor management and sensor data ingestion. The sensor is assigned a device certificate produced by AWS IoT. This device certificate is attached to a policy which allows the certificate holders to publish data to AWS IoT. The sensor measures the moisture level and sends it to a microcontroller (the author uses the ESP32 microcontroller[94]). The microcontroller uses the sensor's device certificate to publish sensor data to AWS IoT via MQTTS protocol which encrypts the sensor data in transit and aligns with security design principle "Classify data by their security requirements and protect them accordingly at rest and in transit" in Table 13. The use of device certificate aligns with the security design principle "Implement the principle of least privilege and zero-trust zone where each interaction between resources must be authenticated and authorized" in Table 13.

92 https://github.com/hiukim/mind-ar-js
93 https://github.com/mrdoob/three.js
94 https://www.espressif.com/en/products/socs/esp32

3	IAM role with policy to write to sensor data table	AWS IoT is assigned an IAM role (IAM = identity access management) which allows it to write sensor data to a table in a *DynamoDB* instance, a serverless NoSQL database. Instead of configuring a virtual machine and run a NoSQL database on the virtual machine, the author chooses DynamoDB serverless solution to offload the heavy-lifting tasks of managing virtual machines and database instances to AWS. Furthermore, the DynamoDB instance is charged based on the actual usage and can scale based on traffic volume, so the author does not need to worry about idle capacities. The use of serverless solution aligns with the performance design principle “Offload operational burdens of managing your own infrastructure to cloud providers and allow your teams to focus on value-added tasks” in Table 12 and the cost optimization design principle “Pay for only what you need and avoid idle resources” in Table 11. When using serverless databases, there are multiple settings to allow for cost optimization. Note that serverless databases are charged based on read/ write capacity and the storage space of the data. For the webAR application, the DynamoDB sensor data table is configured to have a 30-day time-to-live, meaning the table automatically deletes sensor data which is more than 30 days old. This is because the soil moisture data from 30+ days ago are not highly relevant for assessing the current health of a house plant. Without setting up the time-to-live setting, the sensor data table will eventually grow big and become costly because of the storage space it requires. Without the time-to-live setting, we can manually delete old data from the table, but this manual action will consume write capacity. The time-to-live setting can be tailored for different data sources with different retention policies. Therefore, setting up time-to-live setting is an example of the cost optimization design principle “Monitor and manage costs” in Table 11. Furthermore, the database instance is configured to use encryption keys, allowing for data stored in the database instance to be encrypted at rest per security design principle “Classify data by their security requirements and protect them accordingly at rest and in transit" in Table 13.
4	AWS Amplify serves static HTML, CSS, JS to browser	When users open the webAR application in a browser, the browser connects to *AWS Amplify* to request the HTML, CSS, and JS and renders the AR dashboard for the users to view. AWS Amplify serves as a web server for the webAR application.
5+6	Browser authenticates to AWS Cognito for permissions to invoke AWS Lambda	When users want to view the watering history and/ or soil moisture sensor data in the AR dashboard, the browser authenticates with *AWS Cognition*, an authentication and authorization, to gain permissions to invoke *AWS Lambda* functions to retrieve the watering history and/ or soil moisture sensor data from the backend DynamodDB database instances. AWS Lambda are serverless functions which perform a small action. For example, the browser passes on the ID of the soil sensor to a Lambda function which then queries for the latest reading of the sensor.

7-12	Read/ write sensor data + IAM role to query/ write sensor data table Read watering record + IAM role to query watering history table	Once triggered by the browser, a Lambda function can read or write sensor data from/ to the backend DynamoDB database instance. Note that there is a Lambda function to read the watering history and another Lambda function to write a new watering record to the watering-history table in the DynamoDB database instance. By isolating read and write traffic, we can allow the two Lambda functions to scale independently. This aligns with the performance design principle “Design a capacity model for your services and choose the right resources and the right size” in Table 12. Additionally, each of these Lambda functions is given an appropriate IAM role for its task. For example, the Lambda function which writes a new watering record to the watering-history table in the DynamoDB database instance should have a role with “write” permission to the watering-history table, not “write” permission to the sensor-data table. This aligns with the security design principle “Implement the principle of least privilege and zero-trust zone where each interaction between resources must be authenticated and authorized” in Table 13.

Table 14. WebAR Application Technical Descriptions

4.6 Validation Phase

Outcome: Sprint review + training materials + validated value drivers and KPIs
In general, after each sprint, a working application is presented to users for testing. The users can try out the application and give feedback pertaining to not only functional requirements of the application (i.e., does the application do what the users want) but also non-functional requirements such as the ergonomics of the solution (e.g., does wearing AR headset cause too much head fatigue?).

Based on the feedbacks, existing (non)-functional requirements can be modified, and new requirements can be added to the product backlog. Therefore, requirements gathering is an on-going activity.

The team can also iteratively develop training materials based on the user feedback. Finally, developers and business stakeholders should revisit the value drivers and measures KPIs to ensure that the value drivers and KPIs are still relevant and that the proposed AR solutions help achieve the promised business objectives.

For this use case, the key focus is to validate that the functional requirements are fulfilled, and the application is intuitive for users to monitor the health of their house plants.

4.7 Chapter Summary

This chapter discusses the implementation process of a webAR application as guided by the proposed ARIM from CHAPTER 3: ARTIFACT DESIGN – AR IMPLEMENTATION METHODOLOGY (ARIM) to demonstrate how the ARIM can be used in practice and answer RQ#9 (demonstration question) “How can the proposed ARIM be used in practice?”.

The implementation process undergoes all phases in the ARIM (Discovery, Design, Development, Deployment, and Validation) and uses the selection criteria/ requirements from the ARIM to guide the design decisions.

In essence, the webAR application is an AR dashboard to help users better care for their houseplants – see for Figure 47 the application wireframe. The webAR application runs directly in a browser so that the users do not have to download/ install any app nor buy expensive AR headsets. Table 14 discusses the technical implementation of the application along with the cloud architecture the application is deployed to.

During the thesis presentation, a live demonstration of the application will be conducted.

CHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10)

From section 1.3 Research Objective, Research Questions (RQs), and Research Methodology, RQ#10 (validation question) states: What is the utility of the proposed ARIM?

To answer RQ#11, three semi-structured interviews with industry experts are conducted to gather feedback on the overall design and utility of the ARIM. The methodology is then revised based on the feedback in CHAPTER 6: ARTIFACT REVISION.

Section 5.1 Interview Protocol specifies how the interviews were conducted.
Section 5.2 Interviewee Profile gives insights into the interviewees and why they were chosen for the interviews.
Section 5.3 Key Findings from the Interviews synthesizes key take-aways from the interviews regarding the ARIM overall design and its utility as quantified on a 1-5 scale.

5.1 Interview Protocol

APPENDIX D: INTERVIEW PROTOCOL captures the structure of the interview process and the interview questions. The open-ended interview questions are designed to assess the overall design of the ARIM and the individual activities in the Discovery, Design, Development, Deployment, Validation phase. Towards to the end of the interview, the interviewees are asked to score the utility of the ARIM on different dimensions on a 1-5 scale (1 = strongly disagree, 5 = strongly agree).

In addition to the interview questions outlined below, interviewers and interviewees are allowed to ask additional, unplanned questions for clarification and elaboration during the interviews.

Three Accenture's subject matter experts were interviewed, and each interview lasted for approximately an hour.

Note that the use case demonstrated in CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9) has a relatively simple and limited scope. For example, the only key factor which led to the choice of WebAR as the platform for the use case is that it's preferred that users can run the application directly in the browser and do not need to download and install any app on their device. Therefore, this research relies on interviews with experts who have worked on more complex AR solutions (e.g., AR for factory floor operations) to further validate the ARIM.

As noted in section 7.3 Limitations & Future Works, more interviews can be conducted in future efforts with Accenture and non-Accenture experts to gain deeper insights into how to improve the ARIM.

5.2 Interviewee Profile

Table 15 captures the details about the AR experts who was interviewed for the validation of the proposed ARIM. The experts are selected based on their expertise and experience working with AR. To protect the proposed ARIM as Accenture's intellectual property, only Accenture employees are chosen to participate in the interview process.

#	Interviewee Name	Interviewee's Organization	Interviewee's Role
1	[Name #1]	Accenture Netherlands	Leading the metaverse capability (AR and VR) of Accenture Netherlands' Liquid Studio.
2	[Name #2]	Accenture Netherlands	Business & technology delivery consultant with experience working on several AR projects.
3	[Name #3]	Accenture Belgium	Leading the metaverse capability (AR and VR) of Accenture Belgium's Liquid Studio.

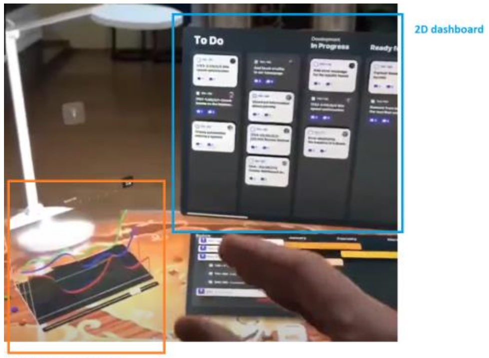
Table 15. Interviewee Profile

5.3 Key Findings from the Interviews

Feedback for the Proposed ARIM

Table 16 synthesizes and organizes feedback from the interviewees by the high-level design and the activities of different phases of the ARIM, allowing for effective revision of the methodology:

Topic	Feedback from interviewee #1	Feedback from interviewee #2	Feedback from interviewee #3
ARIM high-level design	**Feedback:** The methodology is well-structured and provides a clear thought process for AR implementation. It is important, however, to consider how the ARIM complements Accenture's existing frameworks to better align the ARIM with Accenture's current practice.	**Feedback:** The methodology is thorough yet easy-to-understand.	**Feedback:** The methodology is simple yet well-designed. When asked how he would rank the statement, "The methodology is useful to me when delivering AR solutions to clients", he said the methodology would be useful not only during the implementation process of AR projects but also during the sales process where Accenture approaches prospective clients and tries to explain the essence of AR projects to them.
ARIM Discovery phase – "Define problem" activity	**Feedback:** "Define problem" activity of the Discovery phase is an example where it is better to adopt Accenture's existing frameworks > See *S1*. In other words, the Discovery phase should be conducted separately using Accenture's existing frameworks to produce the necessary inputs for the ARIM e.g., user personas, user journeys. This is because the Discovery phase is the starting point for every project, AR-related or not, whereas the ARIM should only be used when project teams have considered AR as a potential technological solution and want to use the ARIM to validate their assumptions. **Suggestions (S) for improvement:** *S1. Downscale the ARIM so that the methodology contains only AR-specific guidance: process suitability for AR assessment, AR platform suitability assessment, and AR development tool suitability assessment.* *For non-AR specific guidance, adopt Accenture's existing frameworks instead.*	**Feedback:** The proposed key activities and outcomes for the "Define problem" high-level activity generally align with Accenture's approach. Regarding the phrasing "Define problem", it is important to note that sometimes the existing processes have been working well for years, so there is no problem to be defined per se. Rather, the clients might look for opportunities to better improve the processes and make them more efficient with the help of suitable technologies.	**Feedback:** The proposed key activities and outcomes for the "Define problem" high-level activity generally align with Accenture's approach. Regarding risk assessment, it is important to note that in practice, it is much more difficult to conduct a thorough risk assessment at the beginning of a project, especially for psychological factors. Therefore, project teams need to continuously collect user feedback and update the risk assessment accordingly.

Topic	Feedback from interviewee #1	Feedback from interviewee #2	Feedback from interviewee #3
ARIM Discovery phase – “Assess process suitability for AR” activity	**Feedback:** The proposed process characteristics are well-supported by the example use cases but can better rephrased for clarity > See *S2*. Specifically, the proposed characteristics are phrased for different levels of granularity. For example, “work instructions” is quite low-level whereas “content enriching” is rather high-level as it includes multiple layers: content aggregation, content publishing, and content interaction. Furthermore, the proposed characteristic “collaboration” is too broad since most processes require workers to collaborate. **Suggestions (S) for improvement:** *S2. Rephrase the process characteristics for more clarity and comparable levels of granularity.*	**Feedback:** The characteristic “hands-free access to content”, “collaboration”, and “hard-to-simulate environment” are well-supported by example use cases. The proposed characteristic “work instructions” sounds like a specific use case than a process characteristic. The proposed characteristic “content enriching” can be further unpacked for more low-level characteristics as commented by interviewee #1 > See *S2*. Interviewee #2 suggests that “location-based context”, although already implied in the proposed characteristic “content enriching”, should be added as an explicit characteristic. AR-guided work instructions as anchored to physical machines and AR dashboards for monitoring the status of physical machines by aggregating different sensor data are examples of how AR provides values through location-based context > See *S3*. The proposed characteristics can be further rephrased to avoid overestimating the value of 3D contents in AR solutions. In the example of an AR dashboard with sensor data for monitoring the status of physical machines, a 2D augmented dashboard typically works well - no need for a 3D augmented dashboard (see Figure 51 for examples of what a 2D vs. 3D dashboard look like) > See *S4*.  *Figure 51. Examples of 2D and 3D Dashboard (source: https://youtu.be/V8tYj_6yN54)* **Suggestions (S) for improvement:** *S3. Add a characteristic called “location-based context”.* *S4. Rephase the process characteristics to emphasize the value of both 2D and 3D augmented contents in AR solutions.*	**Feedback:** The proposed characteristics are well-supported by example use cases.

Topic	Feedback from interviewee #1	Feedback from interviewee #2	Feedback from interviewee #3
ARIM Design phase – “Assess AR platform suitability” activity	**Feedback:** The selection criteria along with example use cases effectively highlight the (dis)-advantages of different AR platforms. No additional selection criteria were proposed.	**Feedback:** The selection criteria are well-supported with the example use cases. Interviewee #2 proposes “heat distribution” as an additional selection criterion since AR smart glasses when heating up can cause skin burns. Therefore, “heat distribution” selection criterion is intended to measure how much heat AR devices produce and how fast the heat can dissipate > See *S5*. **Suggestions (S) for improvement:** *S5. Add “heat distribution” as an additional selection criterion for AR platforms.*	**Feedback:**
ARIM Design phase – “Assess AR development tool suitability” activity	**Feedback:** The selection criteria are well-supported with the example use cases.	**Feedback:** The selection criteria are well-supported with the example use cases. Interviewee #2 mentions rendering speed as a critical factor since lagging AR contents can degrade user experience. The author explains that rendering speed is influenced by multiple factors and, therefore, hard to measure. For that reason, the ARIM incorporates several proxies for measuring rendering speed including network connectivity, computation and storage capacity, and remote rendering. Interviewee #2 mentions that the tracking engine-specific and rendering engine-specific selection criteria seem burdensome since use cases typically require some but not all the listed tracking and rendering capabilities > See *S6*. **Suggestions (S) for improvement:** *S6. Note in the ARIM that the choice of tracking and rendering capabilities should be driven by use case requirements.*	**Feedback:** The selection criteria are well-supported with the example use cases. Interviewee #3 agrees with interviewee #2 that the required tracking and rendering capabilities are specific to use cases > See *S6*. For example, when using AR for product prototyping, it is important that the rendering engine can render high-fidelity and realistic AR contents so that the engineers and designers can examine the small details of the product design. However, high-fidelity rendering is computationally expensive and might be unnecessary for other use cases.
ARIM Design phase – “Refine process requirements” activity	**Feedback:** All interviewees agree that after confirming the use case is suitable for AR, it is necessary to conduct this step and provide developers with necessary inputs (i.e., detailed process requirements) for the to-be-developed AR solution.		
ARIM Development phase – “Develop AR app” activity	**Feedback:** All interviewees agree that the Development, Deployment, and Validation phase of the ARIM pertain to software development best practices and are not AR-specific. Although valuable, the contents overlap and are better substituted with Accenture’s existing frameworks > See *S1*.		
ARIM Deployment phase – “Deploy AR app” activity			
ARIM Validation phase – “Test AR app with users” activity			

Table 16. Feedback for the Proposed ARIM

Evaluation Scores for the Proposed ARIM

After a series of open-ended questions, interviewees were asked to rank several statements regarding the utility of the proposed ARIM on a scale from 1 to 5 (1 = strongly disagree, 5 = strongly agree). Table *17* documents the evaluation scores.

Statement #5 receives the lowest average score of 2.7, echoing the expert feedback that although valuable, the Development, Deployment, and Validation phase of the methodology are not AR-specific and, therefore, better substituted with Accenture's existing frameworks – see suggestion S1 in Table 16. Feedback for the Proposed .

Statement #2 receives an average score of 3.7 because the six identified process characteristics of potential AR use cases can be better rephrased for clarity – see suggestions S2-4 in Table 16.

Statement #3 and #4 receive an average score of 5 and 4 respectively, reflecting the positive feedback from the experts, especially regarding the thoroughness of the selection criteria for AR platforms and AR development tools – see suggestions S5-6 and feedback from in Table 16.

Statement #1 and #6 receive an average score of 4.3, reflecting the experts' appreciation for the methodology not only as a valuable guidance for AR implementation projects but also as discussion material with prospective clients – see feedback from in Table 16.

Statement	Interviewee #1 (a)	Interviewee #2 (b)	Interviewee #3 (c)	Average (a+b+c)/3
From 1 (strongly disagree) to 5 (strongly agree), how would you rank this statement: The overall design of the ARIM is easy to understand and aligns with the current practice of my organization.	3	5	5	4.3
The ARIM is useful in identifying potential use cases for AR (i.e., guidance for the activity "Assess process suitability for AR" of the methodology in Figure 46).	4	3	4	3.7
The ARIM is useful in choosing between AR platforms (i.e., guidance for the "Assess AR platform suitability" activity of the methodology in Figure 46).	5	5	5	5
The ARIM is useful in choosing between AR development tools (i.e., guidance for the "Assess AR development tool suitability" activity of the methodology in Figure 46).	5	3	4	4
The ARIM is useful in guiding the development, deployment, and validation of AR solutions (i.e., guidance for the "Develop", "Deploy", and "Validate" phase of the methodology in Figure 46).	2	2	4	2.7
Overall, the ARIM would be useful to me when delivering AR solutions to clients.	4	4	5	4.3

Table 17. Evaluation Scores for the Proposed ARIM

5.4 Chapter Summary

This chapter details the semi-structured interviews conducted with industry experts to validate the proposed ARIM from CHAPTER 3: ARTIFACT DESIGN – AR IMPLEMENTATION METHODOLOGY (ARIM) and answer RQ#10 (validation question) "What is the utility of the proposed ARIM?"

The interview protocol is documented in section5.1 Interview Protocol, outlining the structure of the interview process and interview questions. Interviewees were asked a series of open-ended questions and to rank several statements regarding the utility of the proposed ARIM on a scale from 1 to 5 (1 = strongly disagree, 5 = strongly agree).

Three Accenture experts were interviewed based on their expertise and experience in AR as documented in section 5.2 Interviewee Profile.

Section 5.3 Key Findings from the Interviews catalogs the feedback from the interviewees, which are summarized in Table 16 and serve as inputs for revising the proposed ARIM in CHAPTER 6: ARTIFACT REVISION, and the evaluation scores from the interviewees, which are summarized in Table *17*.

Overall, the interviewees responded positively to the methodology, specifically regarding how simplistic, well-structured, and thorough the methodology is. Moreover, the interviewees gave an average evaluation score of 4.3 to these statements: "The overall design of the ARIM is easy to understand and aligns with the current practice of my organization" and "The ARIM would be useful to me when delivering AR solutions to clients".

In future efforts, more interviews can be conducted with Accenture and non-Accenture experts to further validate the ARIM.

CHAPTER 6: ARTIFACT REVISION

6.1 Revised ARIM

Table 18 maps the expert feedback synthesized in Table 16 from CHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10) to specific revisions to be made to the original design of the ARIM from CHAPTER 3: ARTIFACT DESIGN – AR IMPLEMENTATION METHODOLOGY (ARIM).

Figure 52 illustrates the revised ARIM.

6.2 Chapter Summary

Based on expert feedback from CHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10), the following key revisions are made to the proposed ARIM:

Revision #1: The ARIM is revised to complement Accenture's existing frameworks instead of replacing them or producing duplicate contents.

The original ARIM has a Discovery, Design, Development, Deployment, and Validation phase.

The revised ARIM presents itself as an in-depth AR use case assessment phase which sits between the Discovery and the Agile phase. The Discovery and Agile phase can be conducted using Accenture's existing frameworks. During the Discovery phase, project teams aim to identify a problem to be solved or an opportunity to be capitalized. If project teams consider AR as a potential technological solution, they can use the ARIM to further validate the use case. Once the use case is validated and project teams conclude that AR is the right solution, project teams pass refined use case requirements to developers to iteratively deliver the proposed AR solutions using Agile framework.

Revision #2: Rephase the process characteristics which make a process a good candidate for AR for better clarity
The original ARIM proposes 6 characteristics:

1. Content enriching
2. Work instructions
3. Hard-to-simulate environment
4. Hands-free access to content
5. Navigation
6. Collaboration

The revised ARIM proposes 5 characteristics:

1. Location-based context (NEW)
2. Hard-to-simulate environment
3. Hands-free access to content
4. Remote assist (NEW)
5. Navigation

The definitions of the newly proposed 5 characteristics are also rephrased based on expert feedback.

Revision #3: Add "heat distribution" as an additional selection criterion for AR platform suitability.

Heat distribution is added per expert feedback that the expert has encountered situations where users complain about heating AR headsets cause mild skin burns and discomfort.

Figure 52 illustrates the revised ARIM.

Expert Suggestions from Table 16	ARIM Revisions
S1. Downscale the ARIM so that the methodology contains only AR-specific guidance: process suitability for AR assessment, AR platform suitability assessment, and AR development tool suitability assessment. For non-AR specific guidance, adopt Accenture's existing frameworks instead.	The ARIM is downscaled to contain only AR-specific guidance per suggestion S1. This guidance pertains to the following activities from the ARIM's original design: 1. Assess process suitability for AR 2. Assess AR platform suitability 3. Assess AR development tools suitability 4. Refine process requirements Using the outputs from the Discovery phase conducted separately, project teams use the ARIM to validate if AR is suitable for the potential AR use cases identified in the Discovery phase. If AR is suitable, the ARIM's outputs are passed onto a series of iterations with each iteration consisting of the Design, Development, Deployment, and Validation phase in accordance with Accenture's Agile methodology to materialize the AR use cases.
S2. Rephrase the process characteristics for more clarity and comparable levels of granularity. S3. Add a characteristic called "location-based context".	The originally proposed characteristic "collaboration" is defined as "Users across geographies work together with 3D contents." This characteristic is now rephrased as "remote assist" for a more clearly defined scope per suggestion S2 and defined as "Remote workers can visually guide and collaborate with local workers without geographical constraints" per suggestion S4. Unlike the definition of "collaboration", the definition of "remote assist" does not specify "3D contents" to avoid equating AR contents with 3D contents. For example, in Figure 42, a local worker (upper left) is calling her remote colleague (upper right) for assistance on how to operate an electrical panel. While the local worker points her device at the panel, the remote worker can view the panel and make virtual annotations circling different buttons to guide the local worker. The local worker can see the virtual circles made by her remote colleague on her phone in real time (lower left is the local worker's phone, lower right is the remote colleague's phone). In this scenario, the 2D virtual annotations suffice – no need for 3D virtual annotations.

<table>
<tr><th>Expert Suggestions from Table 16</th><th>ARIM Revisions</th></tr>
<tr><td>S2. Rephrase the process characteristics for more clarity and comparable levels of granularity.

S4. Rephase the process characteristics to emphasize the value of both 2D and 3D augmented contents in AR solutions.</td><td>The originally proposed characteristic “collaboration” is defined as “Users across geographies work together with 3D contents.”

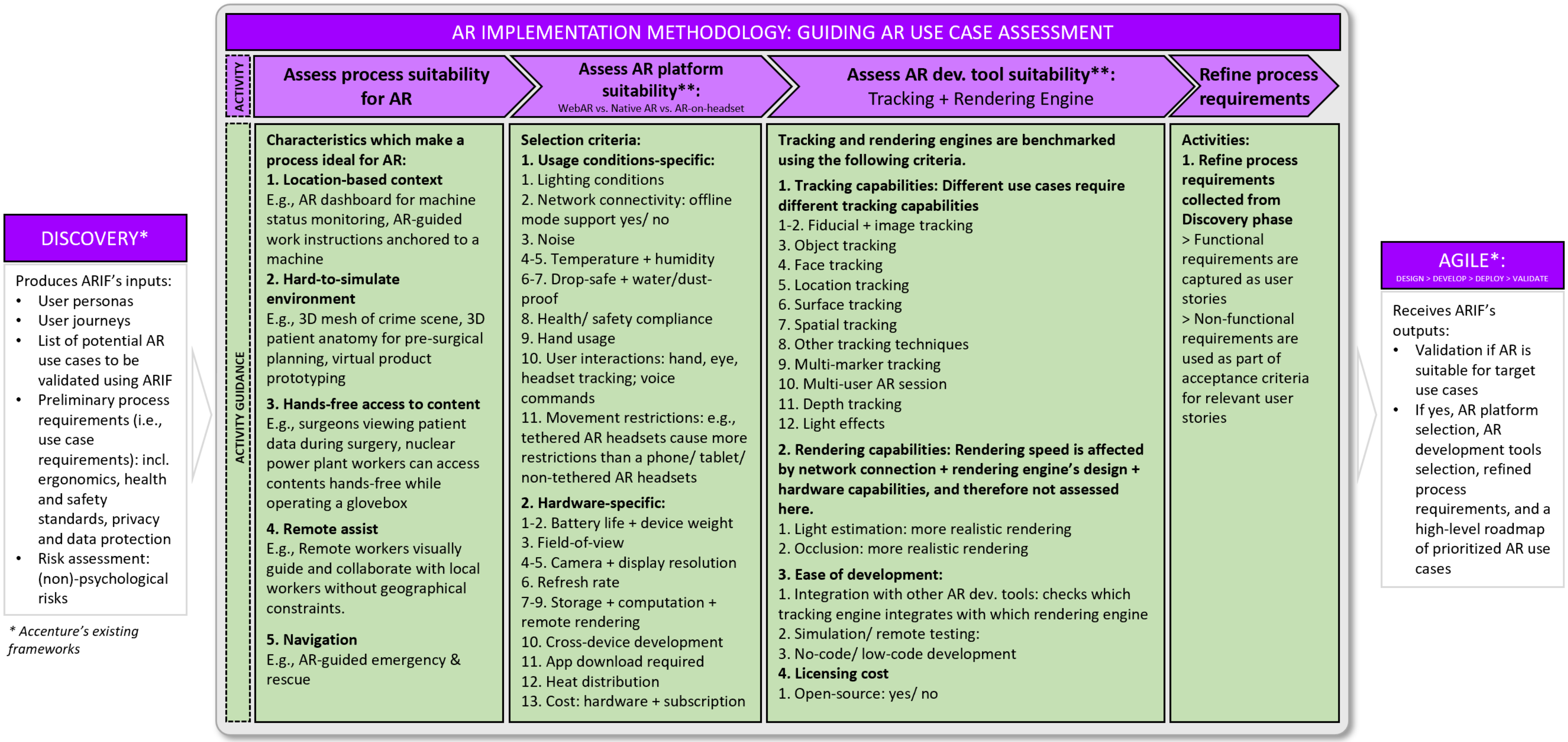
This characteristic is now rephrased as “remote assist” for a more clearly defined scope per suggestion S2 and defined as “Remote workers can visually guide and collaborate with local workers without geographical constraints” per suggestion S4.

Unlike the definition of “collaboration”, the definition of “remote assist” does not specify “3D contents” to avoid equating AR contents with 3D contents.

For example, in Figure 42, a local worker (upper left) is calling her remote colleague (upper right) for assistance on how to operate an electrical panel. While the local worker points her device at the panel, the remote worker can view the panel and make virtual annotations circling different buttons to guide the local worker. The local worker can see the virtual circles made by her remote colleague on her phone in real time (lower left is the local worker’s phone, lower right is the remote colleague’s phone). In this scenario, the 2D virtual annotations suffice – no need for 3D virtual annotations.</td></tr>
<tr><td>S5. Add “heat distribution” as an additional selection criterion for AR platforms.</td><td>Add “heat distribution” to the list of selection criteria for activity “Assess AR platform suitability.”

“Heat distribution” is also added to APPENDIX B: AR PLATFORM DECISION MATRIX.</td></tr>
<tr><td>S6. Note in the ARIM that the choice of tracking and rendering capabilities should be driven by use case requirements.</td><td>Add the following notes to the revised ARIM under “Assess AR development tools suitability” activity:
• Different use cases require different tracking capabilities.
• Different use cases require different rendering capabilities. Not measured here are rendering speed + fidelity as they are affected by network connectivity and AR device’s storage, computation, remote rendering, camera/ display resolution, refresh rate, and field-of-view.</td></tr>
</table>

Table 18. ARIM Revisions



*** Supplemented with Definition Tables + Decision Matrices*

Figure 52. Revised ARIM

CHAPTER 7: DISCUSSION & CONCLUSION

7.1 Procedural Summary

This thesis consists of two phases:

- Phase 1 – Literature review:
  - CHAPTER 1. INTRODUCTION sets the stage for this thesis by providing a soft introduction to Augmented Reality (AR), the research topic, and defining the research objectives, knowledge/ define/ demonstration/ validation research questions, and research methodology.
  - CHAPTER 2. SYSTEMATIC LITERATURE REVIEW conducts search queries on select databases as specified in the literature review protocol and summarizes search query results to answer knowledge research questions RQ#1-8 and set the scientific foundation for the research output.
- Phase 2 – Design artifact (i.e., research output):
  - CHAPTER 3: ARTIFACT DESIGN – AR IMPLEMENTATION METHODOLOGY (ARIM) presents an ARIM based on insights obtained from phase 1 and answers design research questions RQ#9 “What does an industry-agnostic, tool-agnostic AR implementation methodology (ARIM) look like?”
  - CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9) documents the implementation process of an example webAR application guided by the proposed ARIM to demonstrate how the methodology can be used in practice and answer demonstration research question RQ#10 “How can the proposed ARIM be used in practice?”
  - CHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10) documents the semi-structured interview process and the key findings from the interviews to validate the proposed ARIM and answer validation research question RQ#11 “What is the utility of the proposed ARIM?”
  - CHAPTER 6: ARTIFACT REVISION revised the proposed ARIM based on the expert feedback from chapter 5.

7.2 Answers to Research Questions

This section summarizes the answers to the knowledge, design, demonstration, and validation research questions defined in section 1.3 Research Objective, Research Questions (RQs), and Research Methodology.

| RQ#1 (knowledge question): What is AR?
In Figure 3. Reality-Virtuality Continuum [7], AR augments the physical world with virtual contents. AR contents are interactive in real-time, are registered in 3D, and can take the form of visuals, audios, haptics, taste, or smell to augment the respective human’s senses (vision, hearing, touch, taste, and smell).

There are three different AR platforms where AR applications can run on, each with its own advantages and disadvantages:

1. WebAR: AR applications run directly from a web browser
2. Native AR: AR applications need to be downloaded from an app store and installed on a mobile device (smartphone or tablet)
3. AR headset: AR applications run on AR-dedicated hardware such as the Microsoft HoloLens.

AR is typically categorized by tracking techniques (more on "tracking" in *RQ#2*). Following are common tracking techniques:

- Marker-based fiducial/ image tracking: Virtual contents are anchored to a 2D image - see Figure 6, Figure 7, Figure 8
- Object tracking: Anchored to a 3D object see Figure 9
- Face tracking: Anchored to a human face - see Figure 10
- Location tracking: Anchored to a location - see Figure 11
- Surface tracking: Anchored to a planar surface - see Figure 12
- Spatial tracking: Anchored to a 3D mesh of a physical environment - Figure 13

Lastly, with the rising of enabling technologies such as fast 5G internet connections and maturing AR hardware/ software, AR is poised to graduate from a novel technology to an accessible medium which provides an intuitive and enhanced way to consume and interact with content – see section 7.6 Author's Thoughts on the Current State and Future State of AR Technology for how 5G might a critical role in AR adoption.

| RQ#2 (knowledge question): How does AR work?
Two technologies underlie AR:

1. Tracking engine: detects and tracks the location and orientation of a physical object/ environment e.g., a human head
2. Rendering engine: renders virtual content as anchored to the physical object/ environment e.g., rendering a virtual bunny ears as anchored to the human's head

Virtual contents can be classified into:

1. Static AR contents (e.g., 3D models): can be generated using various tools like https://www.blender.org/.
2. Dynamic AR contents (e.g., sensor data): can be sourced from different data sources.

Additionally, AR solutions typically support some forms of user interactivities such as gesture commands, speech commands, gaze commands, and remote assistance. Remote assistance is when, for example, a local maintenance worker can call a colleague working remotely for guidance on how to perform certain tasks. When the local worker points an AR device (mobile phone, tablet, or AR headset) at an electrical panel, the remote worker can view the electrical panel and make real-time, live annotations instructing the local worker to, for example, turn off the red switch.

| RQ#3 (knowledge question): What factors influence AR adoption?
While there is a large number of papers on AR adoption, most of them test for AR adoption using traditional factors (e.g., "perceived ease of use") from well-documented technology acceptance models such as TAM[95]. For this research, only papers presenting new, AR-specific factors for AR adoption are reviewed.

As the result, 7 papers were reviewed, and 17 success factors were extracted from these papers as summarized in Table 4.

| RQ#4 (knowledge question): From a technical standpoint, what is the difference between webAR vs. native AR vs. AR-on-headset application development?
From a technical standpoint, there is no significant difference in the development of webAR vs. native AR vs. AR-on-headset applications since all require a tracking engine and rendering engine.

What is lacking in existing literature is concrete guidance on how to choose between different AR platforms (webAR vs. native AR vs. AR headset) for specific use cases. This research gap (referred to as "AR platform suitability" in this research) is addressed in RQ#8.2.

| RQ#5 (knowledge question): From a technical standpoint, what is the difference between webAR vs. regular, non-AR web application development?
From a technical standpoint, the biggest difference between webAR and regular, non-AR web applications is that webAR requires the use of tracking and rendering engines while regular, non-AR web applications do not.

The architectures of webAR and regular, non-AR web applications are, however, the same:

- Front-end web server: Serves contents to the browser. The browser then renders the contents as a web page for users to review.
- Back-end server(s): Regular web applications typically offload computationally expensive operations to back-end servers. Similarly, webAR can offload, for example, advanced computer vision machine learning models for hand tracking to a back-end server and allow users to interact with the applications using gesture commands.

[95] https://en.wikipedia.org/wiki/Technology_acceptance_model

| RQ#6 (knowledge question): What is cloud serverless infrastructure?
Serverless infrastructure relies on the use serverless cloud services so that developers do not have to manage the physical infrastructure (i.e., servers) their applications run on. By delegating heavy lifting infrastructure management tasks to cloud providers, developers can focus on what they do best – developing applications.

Additionally, serverless architecture allows for automatic scaling of the applications based on traffic (i.e., provisioning new servers and run the application on these additional servers so that the application can serve more traffic) and pay-for-use billing.

Serverless cloud services are typically classified by:

- Category:
  - Compute: Cloud resources which provide processing capabilities e.g., virtual machines
  - Storage: Cloud resources which provide storage capabilities e.g., SQL database instances
  - Service Integration: Cloud resources which integrates different application microservices e.g., middleware service
- Service type:
  - Infra-as-a-Service (IaaS)
  - Container-as-a-Service (CaaS)
  - Platform-as-a-Service (PaaS)
  - Function-as-a-Service (FaaS)
  - Software-as-a-Service (SaaS)

| RQ#7 (knowledge question): What is existing research on AR implementation frameworks?
Three AR implementation frameworks were identified in existing literature – each with its own merits and deficiencies.

Paper [85] is a six-step framework for implementing AR solutions in the marketing industry. The paper provides a list of design considerations for maximized user acceptance, but not specify 1) how to identify if a general use case is a potential candidate for AR as the paper is specific to the marketing industry (referred to as "process suitability for AR" in this research) 2) choose between different AR platform: webAR v. native AR v. AR headset ("AR platform suitability") and 3) choose between AR development tools ("AR development tool suitability).

Paper [86] is a six-step framework with high-level guidance for business leaders who want to understand the general process of AR implementation, the impact of AR, for example, to current business operations and user experience, and the type of human/ technical resources required for the implementation. Unlike paper [85], paper [86] is not industry-specific. However, paper [86] shares the same downfalls of not having concrete guidance for assessing 1) process suitability for AR 2) AR platform suitability 3) AR development tool suitability.

Paper [87] is not an AR implementation framework but presents a design thought process for AR implementation by proposing that a task should be decomposed into perceptual tasks (i.e., tasks which are accomplished through senses such as sights or hearing) to be assisted with the use of AR. The approach makes a grand assumption that the process should be enhanced with AR and ignores the fact that AR might be ill-suited for some processes. Like paper [85] and [86] also do not provide concrete guidance for assessing 1) process suitability for AR 2) AR platform suitability and 3) AR development tool suitability.

Furthermore, all three papers [85], [86], [87] do not provide technical guidance for the development and deployment of AR applications as no prototype was implemented as to demonstrate how the proposed methodologies work.

In summary, the following research gaps were identified in existing literature:

1. A lack of concrete guidance for assessing “process suitability for AR”: this gap is addressed in RQ#8.1
2. A lack of concrete guidance for assessing “AR platform suitability”: this gap is addressed in RQ#8.2
3. A lack of concrete guidance for assessing “AR development tool suitability”: this gap is addressed in RQ#8.3

   A lack of technical guidance for the development and deployment of AR applications this gap is addressed in RQ#8.4 and 8.5

| RQ#8 (design question): What does an industry-agnostic, tool-agnostic AR implementation methodology (ARIM) look like?

Five key inputs drive the high-level design of the proposed ARIM – see Figure 30. ARIM High-Level Design:

1. Software development lifecycle (SDLC): Figure 28 illustrates the formal procedure that the U.S. Department of Justice uses for the acquisition, design, development, implementation, and maintenance of information systems.
2. Design Thinking methodology: A user-centric problem-solving process which focuses on empathizing with users’ challenges, rapidly developing different iterations of a solution prototype, and testing the prototype with users for early feedback - see Figure 29.
3. AR adoption success factors: see answer to *RQ#3*.
4. Technical knowledge of AR and cloud serverless infrastructure: see answers to *RQ#1, 2, 4, 5, 6, 7*.

Since AR applications are in essence a piece of software, the proposed ARIM adopts the 10-phase SDLC and Design Thinking methodology into its five non-linear, iterative phases: Discovery, Design, Development, Deployment, and Validation. Each phase is associated with a set of activities, high-level selection criteria/ requirements/ sub-activities for each activity, and expected outcomes of each activity.

The high-level selection criteria/ requirements are derived from AR adoption success factors so that AR applications created using the proposed ARIM can achieve optimal user acceptance – see Table 10. Mapping AR Success Factors to ARIM's High-Level Selection Criteria and Requirements.

RQ#8 is further decomposed into *sub research questions RQ#8.1-8.5* to address research gaps in existing literature and add more details to the ARIM's high-level design. The insights from *RQ#8.1-8.5* are synthesized into the final design of the ARIM in Figure 46.

|| RQ#8.1: What makes a process ideal to be enhanced with the use of AR (i.e., Process suitability for AR)?
A total of 39 use cases from both industry/ academic literature and from multiple industries was reviewed in detail to synthesize common characteristics which make a process ideal for AR.

Six characteristics were proposed in the original design of the ARIM and later rephrased as five characteristics based on expert interviews from CHAPTER 6: ARTIFACT REVISION.

Process characteristic #1: Location-based context
AR is suitable if users can make better decisions when provided with information about a physical object or environment at a specific location such as AR-guided work instructions as anchored to a physical machine – see Figure 33.

Process characteristic #2: Hard-to-simulate environment
AR is suitable if a process is enhanced when users need to work with 3D environments which are otherwise hard to simulate in real life such as 3D patient anatomy – Figure 34.

Process characteristic #3: Hands-free access to content
AR is suitable if a process is enhanced when users need to access content hands-free while performing a task such as the use of AR headsets by nuclear plant power workers to access work instructions using voice/ gaze control while having both of their hands inside a glovebox – see Figure 37.

Process characteristic #4: Remote assist
AR is suitable when worker productivity is enhanced by allowing remote workers to visually guide and collaborate with local workers without geographical constraints – see Figure 42.

Process characteristic #5: Navigation
AR is suitable if a process is enhanced when users are guided to find a location or an object – see Figure 38.

Note that the proposed process characteristics are intended to serve as "positive confirmations", meaning if a process possesses any of these six characteristics, the process is likely to be an ideal candidate for AR. "Negative confirmation" characteristics which signal a process as a non-ideal candidate for AR are not explored in this research as noted in section 7.3 Limitations & Future Works.

|| RQ#8.2: How to choose between webAR vs. native AR vs. AR headsets for a selected process (i.e., AR platform suitability)?
When creating the high-level design of the ARIM (*RQ#9*), AR adoption success factors are mapped to 2 high-level selection criteria for AR platforms:

1. Usage conditions-specific: The chosen AR platform must be suitable for the conditions in which the AR solution is deployed. For example, shopfloors might not have reliable network connection so, for example, an AR-guided work instructions solution for shopfloor workers must support offline mode.
2. Hardware-specific: Different AR platforms have different hardware-specific characteristics (e.g., battery life, field of view) which are important for the decision of which AR platform to implement for a use case.

Section 3.3 RQ#8.2: How to Choose Between WebAR vs. Native AR vs. AR Headsets for a Selected Process? (Design Phase > AR Platform Suitability) proposes 11 usage conditions-specific and 12 hardware-specific selection criteria for AR platforms and uses example use cases to illustrate the motivation behind each selection criterion. One additional hardware-specific selection criterion called “heat distribution” is proposed by industry experts from CHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10) and added to the list of AR platform selection criteria.

Table 19 summarizes the proposed AR platform selection criteria.

APPENDIX B: AR PLATFORM DECISION MATRIX benchmarks different AR platforms (webAR, native AR, and 7 different AR headsets) using the proposed selection criteria to help business stakeholders and developers quickly identify an appropriate AR platform for their target use cases.

Category	Selection criteria	Description
Usage condition-specific	1. Lighting conditions	Harsh lighting conditions (too dim or too bright) will require AR platforms whose tracking engine must be robust enough to process live camera feed of degrading quality and detect a marker.
	2. Network connectivity	When deployed in environments without stable internet connection like shopfloors, ideal AR platforms are the ones with offline mode support.
	3. Noise	When deployed in loud environments like shopfloors, ideal AR platforms are those with noise-cancelling microphones.
	4-7. Operating temperature and humidity, drop-safe, water/dustproof	When deployed in rugged conditions, ideal AR platforms are those built for extreme temperature, humidity, drop-safe, waterproof, and dustproof.
	8. Compliance with health/ safety standards	Some use cases will require AR platforms to comply with specific health and safety standards e.g., hardhat integration, clean rooms, eye safety.
	9. Hand usage	Some tasks require users to use both of their hands e.g., surgeons carrying out a surgery (making AR headsets most ideal) while some tasks occupy only one hand, allowing users to use their other hand to hold a mobile device (making it possible to use webAR or native AR).3
	10. User interactions	Different AR platforms support different built-in user interactions e.g., voice/ gesture/ gaze controls for an intuitive and enhanced user experience.
	11. Movement restrictions	Some tasks require users to have a full range of movement (e.g., shopfloor operators), making non-tethered AR headsets, webAR, and native AR ideal options. Some other tasks require only a limited range of movements, making tethered AR headsets an acceptable option in exchange for better tracking and rendering performance and lower latency since the tethered headset usually is connected to a powerful computer.
Hardware-specific	1. Battery life	The longer the battery life, the fewer charges are required between usage.
	2. Device weight	The heavier the device, the more fatigue it causes, especially head fatigue when using AR headsets.
	3. Field-of-view (FoV)	The wider the FoV, the bigger virtual contents users can view and the more immersive the AR experience feels.
	4. Camera resolution	The higher the camera resolution, the higher quality video the camera can capture. It is valuable, for example, when the live camera is streamed to a remote worker for the use case of AR-guided remote assistance.
	5. Display resolution	The higher the display resolution, the higher fidelity the virtual contents are.
	6. Refresh rate	The higher refresh rate, the smoother the virtual contents appear.
	7-9. Storage, computation, and remote rendering	Storage, computation, and remote rendering dictate how much tracking and rendering can be performed locally on the AR device (no latency) or remotely in the cloud (requiring less powerful AR devices but results in higher latency).
	10. Cross-device development	Allows developers to develop a webAR, native AR, and AR-on-headset application once and deploy the application on multiple browsers, Android/ iOS devices, and AR headsets, respectively.
	11. App download required	Refers to if a user needs to download and install the AR app to their device. WebAR is the only platform that does not require users to download and install webAR apps since webAR runs directly from a browser, giving it an advantage over native AR and AR headsets.
	12. Heat distribution	Refers to how much heat an AR device produces while running and how fast the heat can dissipate because of the thermal model of the AR device. This factor is important because heating AR devices, especially AR headsets, can cause skin burns and affect user experience.
	13. Cost	Refers to the cost of using the AR platforms (hardware cost + subscription cost).

Table 19. AR Platform Selection Criteria Summary

|| RQ#8.3: How to choose between different AR development tools for a selected process and AR platform (i.e., AR development tool suitability)?

When creating the high-level design of the ARIM (*RQ#9*), AR adoption success factors are mapped to 4 high-level selection criteria for AR development tools:

1. Tracking capabilities: Each AR development tool is classified as either a tracking engine, a rendering engine, or both (a hybrid engine which performs both tracking and rendering task). Tracking engines are evaluated for their supported tracking capabilities since different use cases require types of tracking.
2. Rendering capabilities: Rendering engines are evaluated for their ability to render realistic contents.
3. Ease of development: Refers to how easy it is to use tracking and rendering engines to develop an AR application.
4. Cost: Refers to the cost of tracking/ rendering engines.

Section 3.4 RQ#8.3: How to Choose Between Different AR Development Tools for a Selected Process and AR Platform? (Design Phase > AR Development Tool Suitability) proposes 12 tracking capabilities, 2 rendering capabilities, 3 ease of development, and 1 cost selection criteria for AR development tools and uses example use cases to illustrate the motivation behind each selection criterion.

Table 20 summarizes the proposed AR development tool selection criteria.
APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX benchmarks 18 different AR development tools using the proposed selection criteria to help business stakeholders and developers quickly choose appropriate tools for materializing their target use cases.

Category	Selection Criteria	Description
Tracking capabilities	1. Fiducial tracking	Different use cases require different tracking techniques.
	2. Image tracking	In the example of an art book in Figure 32, the art book is implemented using an image tracking engine which allows users to scan an image of a sculpture in the artbook and view a 3D model of the sculpture rendered on top of the image. On the other hand, spatial tracking is more appropriate for implementing an AR-guided evacuation route for an underground mine since it would not be feasible to attach image markers to rock faces and make sure the image markers stay put.
	3. Face tracking	
	4. Location tracking	
	5. Object tracking	
	6. Surface tracking	
	7. Spatial tracking	
	8. Depth tracking	Depth tracking is an additional capability of tracking engines to understand the physical environment. With depth knowledge, tracking engines can instruct rendering engines to occlude virtual objects if they are behind physical objects for a more realistic experience.
	9. Other tracking	As more computer vision technology evolves, more tracking techniques are being developed. This criterion serves as a catch-all for those techniques.
	10. Multi-marker tracking	Refers to a capability of tracking engines to detect and track multiple markers in a single video frame, creating a richer experience – see Figure 41.
	11. Multi-user tracking	Refers to the ability of tracking engines to synchronize across multiple devices and allow multiple users to join a single AR session. For example, an AR-guided remote assistance solution requires the use of multi-user tracking – see Figure 42.
	12. Lighting effects	Refers to additional capabilities of tracking engines such as light estimate i.e., predicting the lighting conditions of the physical environment.
Rendering capabilities	1. Light estimate	Using the light estimate data from tracking engines, rendering engines can shade the virtual contents to match the lighting conditions of the physical environment for more realistic virtual contents.
	2. Occlusion	Using the depth data from tracking engines, rendering engines can occlude virtual objects which stand behind physical objects for more realistic virtual contents.
Ease of development	1. Integration with other AR development tools	Cross-references which tracking engines integrate with which rendering engines. Refresher: an AR application requires both tracking and rendering engines, and the two engines need to integrate with each other.
	2. Simulation/ remote testing	Allows developers to test and debug AR applications more quickly.
	3. Low-code/ no-code development	Allows business users with little to no programming knowledge to create new AR experiences e.g., allowing shopfloor operators to create new AR-guided work instructions.
Licensing cost	1. Open-source	Checks if the tracking/ rendering engines are open-source or not.

Table 20. AR Development Tool Selection Criteria Summary

|| RQ#8.4: What are the requirements for the development of AR apps?
When creating the high-level design of the ARIM (*RQ#9*), AR adoption success factors are mapped to 1 high-level requirement for AR applications:

1. A flexible architecture for maintainability, agility, and time-to-market.

This research proposes 2 low-level requirements from this high-level requirement for the development of AR solutions:

1. Microservice architecture: Allows AR applications to be decomposed into small, reusable components which are faster to develop, test, and deploy and make the overall AR application more flexible for future growth.
2. Agile development: Agile provides a methodology for developers to organize their development works into iterative, short (typically 2 weeks) development time periods called "sprints" where a working prototype is delivered and tested with end users for feedback at the end of each sprint.

|| RQ#8.5: What are the requirements for the deployment of AR apps?
When creating the high-level design of the ARIM (*RQ#9*), AR adoption success factors are mapped to 4 high-level requirements for AR applications:

1. Continuous integration and continuous delivery (CI/ CD)
2. Design for cost optimization
3. Design for performance
4. Design for security

CI/ CD refers to the practice of using various tools to automate the tasks of building source code into build artifacts, running tests on the build artifacts, and deploying the build artifacts to the production environment. This practice allows developers to work more efficiently and deliver new functionalities for quickly.

Cloud architecture design for cost optimization, performance, and security are complex topics and deserve to be a research topic on their own. Therefore, this research does not aim to provide concrete guidance for how to design a cost-effectively, performant, and secured cloud architecture for AR applications. Rather, this research consolidates the design principles for cost optimization, performance, and security as proposed by three leading cloud providers AWS, Google Cloud, and Microsoft Azure as a guiding principles for the deployment of AR applications in the cloud – see Table 11, Table 12, Table 13 respectively.

To answer *RQ#10*, a webAR application is implemented and deployed on cloud serverless resources. Table 14. WebAR Application Technical Descriptions gives examples of how the design principles from Table 11, Table 12, and Table 13 are translated to concrete cloud architecture design decisions.

| RQ#9 (demonstration question): How can the proposed ARIM be used in practice?
The author implements an AR application as guided by the ARIM to demonstrate how the methodology can be used in practice. The author also uses insights collected from this implementation process to further finetune the ARIM.

The implementation process is documented in 5 phases of the ARIM with the specific outcomes for each phase: Discovery, Design, Development, Deployment, and Validation.

The AR application allows users to scan an image marker affixed next to their house plants. An AR dashboard is then rendered as anchored to the image marker. Through this dashboard, users can view critical information and use the dashboard's functionalities so that they can better care for their plants:

- Users can view soil real-time moisture sensor data to determine if they should water their plants now
- Users can record the time they water their plants and the current health status of the plants. This allows the users to track when was the plants were watered last and what was the health status of the plants.
- Users can view additional information in text and videos for how to care for the plants.

Since houseplants are in a relatively standard environment (e.g., no extreme temperature or weather), there are no specific requirements relating to the usage conditions. One hardware-specific criterion is that it would be better if house sitters do not have to download and install the application on their device because they will not use the application frequently. Otherwise, house sitters might find the task of downloading and installing the application too cumbersome and skip using the application all together. For that reason, webAR is chosen as the AR platform.

For selecting the AR development tools, budget is the primary concern. Open source webAR tracking engines only support fiducial/ image tracking, location, and surface tracking. Location tracking will not work for plants located indoors, and surface tracking does not connect the AR dashboard to a specific plant. That leaves fiducial and image tracking as the only two options. Image tracking was chosen for the implementation, but fiducial tracking would have worked just as well.

The following tools were used to implement the webAR application – see Table 14. WebAR Application Technical Descriptions for details:

1. MindAR: webAR tracking engine for image tracking
2. Three.js: webAR rendering engine
3. Tensorflow: a machine learning model for implementing gesture commands
4. Babel, Webpack, NPM: other development tools (build tools and package manager)
5. Azure DevOps: CI/ CD tool
6. AWS CloudFormation: an infra-as-code tool
7. AWS IoT: sensor management and sensor data ingestion platform
8. AWS Cognition: an authentication and authorization service
9. AWS Amplify: a web and mobile application platform
10. AWS Lambda: a function-as-a-service
11. AWS DynamoDB: a serverless NoSQL database service

| RQ#10 (validation question): What is the utility of the proposed ARIM?
To validate the utility of the proposed AR, three semi-structured interviews were conducted with industry experts within Accenture. The interview protocol and interview questions are outlined in APPENDIX D: INTERVIEW PROTOCOL. The interviewees were chosen based on their expertise and experience working with AR solutions – see Table 15. Interviewee Profile.

Interviewees were asked a series of open-ended questions to gather their feedback on how useful the proposed ARIM for guiding AR implementation projects and to rank several statements regarding the utility of the methodology. As demonstrated in Table *17*. Evaluation Scores for the Proposed , the experts responded positively to the methodology and expect it be valuable for guiding not only AR implementation projects but also for discussions with prospective clients.

Table 16 summarizes expert feedback, and Table 18 maps the feedback to specific revisions made to the proposed ARIM. The revised ARIM is in Figure 52.

7.3 Limitations & Future Works

There are several *limitations* to this research effort:

1. The process characteristics proposed in section 3.2 RQ#8.1: What Makes a Process a Good Candidate for Enhancement with the Use of AR? (Discovery Phase > Process Suitability for AR) are synthesized from 39 use cases selected using several inclusion and exclusion criteria because it is not feasible to review all AR use cases ever published. However, the use cases cover both academic (22 out of 39) and industry literature (17 out of 29) and span across multiple sectors (manufacturing, entertainment and tourisms, healthcare, education, emergency and rescue, military, etc.) for a balanced representation.
2. The process characteristics proposed in section 3.3 RQ#8.2: How to Choose Between WebAR vs. Native AR vs. AR Headsets for a Selected Process? (Design Phase > AR Platform Suitability) serve as positive confirmations, which means if a process possesses any of the proposed characteristics, the process is likely an ideal candidate for AR. No "negative confirmation" characteristics which signal a process as non-ideal for AR are explored in this research.
3. APPENDIX B: AR PLATFORM DECISION MATRIX and APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX do not exhaustively benchmark all AR headsets and all AR development tools. However, they sufficiently demonstrate how different AR headsets and development tools fare against each other with respect to the proposed selection criteria.
4. One of the key inputs for the proposed ARIM is existing literature on AR adoption success factors. As AR technology and AR user experience evolve, there might be new AR success factors which are not yet examined in this research. To counter this limitation, the proposed ARIM can be validated with more complicated use cases (more complicated than the webAR application from CHAPTER 4: ARTIFACT DEMONSTRATION – WEBAR APPLICATION (RQ#9) to identify any missing selection criteria and requirements to be incorporated into the methodology.

5. More expert interviews within and outside Accenture can be conducted to further validate the proposed ARIM.

For *future works*, below are possible expansions for the proposed ARIM:

1. Include Virtual Reality and convert the proposed ARIM into an implementation methodology for extended reality covering both AR and VR.
    a. This motivation stems from the similarities that AR and VR share. For example, both VR and AR rely on the use of virtual contents for an enriched user experience. However, there might be use cases where VR is a better fit than AR and vice versa.
    b. An implementation methodology for extended reality will answer a critical question: Is a process a good candidate for extended reality? If yes, AR or VR?
2. Include Assisted Reality as an adjacent technology to AR. While AR overlays virtual contents onto a physical environment, Assisted Reality allows users to view a screen within their field of view, hands-free [116].
    a. An example of Assisted Reality is the RealWear Navigator headset[96] in Figure 53.

Figure 53. Assisted Reality - RealWear Navigator headset[97]

7.4 Academic and Practical Contributions

From an *academic standpoint*, the proposed ARIM addresses 4 research gaps in existing literature – see Table 9 for details:

1. Lack of concrete guidance for assessing if a process can be and should be enhanced with the use of AR: i.e., RQ#9.1 Process suitability for AR
2. Lack of concrete guidance for identifying an appropriate AR platform for a specific process: i.e., RQ#9.2 AR platform suitability
3. Lack of concrete guidance for identifying appropriate AR development tools for a specific process and AR platform: i.e., RQ#9.3 AR development tool suitability
4. Lack of technical guidance for the development and deployment of AR applications: i.e., RQ#9.4 and 9.5.

---

[96] https://www.realwear.com/navigator/
[97] https://youtu.be/7yjQ5LYT9ok

From a *practical standpoint*, the ARIM presents an industry-agnostic and tool-agnostic AR implementation methodology, guiding project teams in delivering AR implementations across different industries and in selecting the most suitable AR platform and tools for their target use cases. Furthermore, the methodology is validated through interviews with Accenture's industry experts further to refine and align the methodology with Accenture's current practice.

7.5 Recommendations for Organization Looking to Implement the Proposed ARIM

Organizations who wish to implement the proposed ARIM should consider the following factors:

1. The organizations should actively maintain APPENDIX B: AR PLATFORM DECISION MATRIX and APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX as AR technology evolves and new AR platforms/ development tools are introduced.
2. The organizations should actively identify new characteristics which make a process an ideal candidate for AR since newly developed AR capabilities might make previously impossible use cases possible.
3. Organizations who do not have existing frameworks for guiding the Discovery, Development, Deployment, and Validation phase are recommended to use the originally proposed methodology presented in Figure 46 instead of the revised methodology in Figure 52 which is intended to complement Accenture's existing frameworks.

7.6 Author's Thoughts on the Current State and Future State of AR Technology

Since Ronald Azuma, a prominent computer scientist and AR researcher, published a now-then widely accepted definition of AR in his 1997 paper [8] (see section AR Definition), AR technology has gone a long way. This research perceives three key areas where today's AR technology is lacking and where future's AR technology might go next.

Disclaimer: This section is written based on the author's observations/ thoughts and is not supported by empirical evidence.

First, AR technology will continue to mature and augment more complex human senses such as touch, taste, and smell.
Existing AR use cases such as AR-guided work instructions primarily focus on augmenting human vision and hearing.

AR for augmenting human touch using haptic gloves is still mostly restricted to gaming use cases. It is, however, exciting to imagine a future where, say, fashion designers across the globe can collaborate on a virtual dress. Using AR devices, they can not only view the dress in 3D but also "feel" the dress material as they iterate through their designs.

AR for augmenting human taste and smell opens another exciting venue where, say, food scientists can collaborate on designing a new condiment. The food scientists can review the 3D model of the condiment's chemical compounds while experiencing how the condiment will taste and smell differently as they iterate through different chemical compositions.

Second, AR hardware will continue to mature and address existing limitations such as high fidelity rendering in real time, heat distribution, and cost.
Virtual product prototyping is an AR use case where high-fidelity rendering is required. For example, as car designers and engineers across the globe collaborate on a new car design, it is important that the car's life-size, hyper-realistic 3D model can be rendered in real-time for high user experience.

This kind of rendering is more computationally expensive than, say, rendering an AR monitoring dashboard for factory floor equipment with relatively simple user interface. The good news is that there are several improvements to enable real time, high fidelity rendering job:

1. AR hardware is becoming more powerful (i.e., more CPU/ GPU + memory/ storage)
2. If AR hardware is not powerful enough, the rendering engine can run in the cloud on powerful GPUs and stream rendered video frames to AR devices as demonstrated in NVIDIA CloudXR streaming service[98]. This is called "remote rendering", a selection criterion for AR platform – see APPENDIX B: AR PLATFORM DECISION MATRIX. The main downside of remote rendering is latency as the rendered video frames produced by the rendering engine need to be sent to the AR device from some remote servers in the cloud.
3. 5G technologies can address the latency problem for remote rendering as the 5G coverage continues to expand throughout the world. As of 2022, 5G in Europe accounts for only 2.8% of the total mobile connections [117].

Even if AR hardware becomes powerful enough to render high-fidelity graphics, heat distribution is another obstacle as a heating up AR device can cause mild skin burns or discomfort, especially AR glasses. New AR devices need to balance between powerful processing power and a thermal model for effective heat distribution.

Lastly, AR hardware costs are still on the high with the HoloLens 2 priced at €3.879. More affordable AR hardware will enable wider adoption among not only industrial but also personal use cases.

Third, AR software (tracking and rendering engines) will continue to mature to address current existing limitations such as accurate tracking, realistic rendering, and a lack of widely available low-code/ no-code tools.
Tracking engines are built on computer vision technology to detect the presence of a marker. As computer vision technology matures, so will tracking technology for more accurate tracking.

For example, section Spatial Tracking discusses spatial tracking as the tracking engine's ability to create a 3D mesh of a physical environment and users can anchor visual contents to this 3D

[98] https://www.nvidia.com/en-us/design-visualization/solutions/cloud-xr/

mesh. One can assert that spatial tracking seems superior to other tracking techniques. However, based on insights collected from CHAPTER 5: ARTIFACT VALIDATION – EXPERT INTERVIEWS (RQ#10), existing tracking engines with spatial tracking capability still fall short in term of accuracy. For example, if we implement an AR-guided navigation app using spatial tracking to guide a factory worker to different workstations, the tracking engine might not be able to provide accurate navigations if the workstations are located too close to each other.

Realistic rendering is another AR software limitation. Section Rendering Capability-Specific Selection Criteria discusses two criteria for rendering engines: 1) light estimations 2) occlusions. If a rendering engine can render virtual contents according to the physical environment's lighting conditions ("light estimations") and occlude the virtual contents as they are behind physical objects ("occlusions"), the virtual contents will appear more realistic. Suppose a car dealership allows potential buyers to view a 3D model of a new car, it would be great if the car's 3D model is rendered as realistic as possible such as the users can see their own reflection as they look closely at the car's body in 3D. Rendering engines of tomorrow will likely expand its capabilities for light estimations, occlusions, and others for hyper-realistic rendering.

Lastly, low-code/ no-code development is a selection criterion for AR development tools - see APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX. There is already several low-code/ no-code AR development tools which allow non-technical audience (i.e., non-developers) to create new AR experiences. For example, shopfloor operators might want to be able to create new AR-guided work instructions themselves instead of having to wait for developers to code up new work instructions. In the future, there will be likely more low-code/ no-code development tools available to bring AR to a larger audience.

7.7 Chapter Summary

This chapter concludes the thesis by summarizing the research procedure and key findings.

Section 7.1 Procedural Summary captures the procedure that guides this research from phase 1 literature review to phase 2 artifact design.

Section 7.2 Answers to Research Questions summarizes answers the knowledge research questions (RQ#1-7), design research question (RQ#8), demonstration research question (RQ#9), and validation research question (RQ#10).

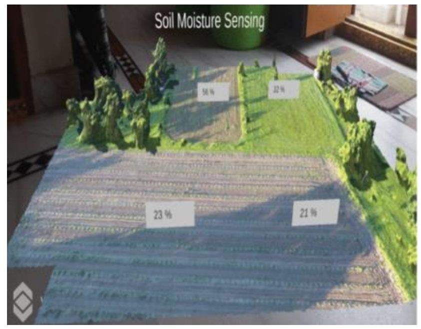
Section 7.3 Limitations & Future Works summarizes the limitations and opportunities for future works for this research. The limitations serve as a precaution for the shortcomings present in the proposed ARIM. The future works present opportunities for extending the methodology for more added values.

Section 7.4 Academic and Practical Contributions captures the key academic and practical contributions of this research.

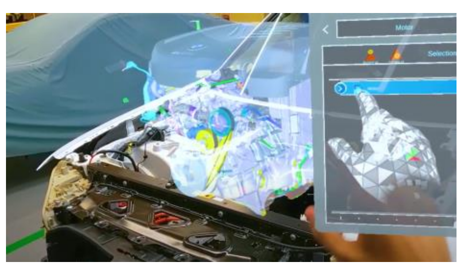
Section 7.5 Recommendations for Organization Looking to Implement the Proposed ARIM provides additional guidance for organizations who wish to adopt the ARIM for their use.

Section 7.6 Author's Thoughts on the Current State and Future State of AR Technology offers the author's personal opinions on the existing limitations of AR technology and where the AR technology might go next.

APPENDIX A: AR USE CASE SUMMARY

#	Use Case	Sector	Process Characteristics	Use Case Description
1	Real-life Applications of integration of Augmented Reality and Internet of Things [118]	Industrials	Content enriching	Content enriching • An engine can remotely monitor the health status of a farm by checking the real-time sensor data for different sections of the farm.  *Figure 54. AR in Sensor Monitoring [118]*
2-3	Research on Visual Monitoring and Auxiliary Maintenance Technology of Equipment Based on Augmented Reality [45] Indoor Augmented Reality Using Deep Learning for Industry 4.0 Smart Factories [119]	Industrials	Content enriching	Content enriching • A worker can use their phone to scan a physical machine and view real-time sensor data indicating the status of the machine augmented in 3D, speeding up the inspection process.  *Figure 55. AR in Machine Monitoring [119]*
4	NETIVAR: NETwork Information Visualization based on Augmented Reality [44]	Industrials	Content enriching	Content enriching • The task of connecting network cables to network switches is time-consuming and requires extensive documentation and preparation beforehand. With AR, workers can scan a physical network switch and view information about the network switch augmented in 3D assisting them in performing network maintenance tasks.

#	Use Case	Sector	Process Characteristics	Use Case Description
				 *Figure 56. Switchboard with AR Contents [44]*
5-6	The future of construction with augmented reality and BIM [120] Microsoft HoloLens 2: Partner Spotlight with VisualLive and NOX Innovations [97]	Industrials	Content enriching; Hard-to-simulate environment; Collaboration;	Hard-to-simulate environment • AR allows construction managers to view 3D models of the structures of a building (e.g., a pillar) to be built as anchored in a physical space • As the physical structures are getting built, construction managers can quickly check if the construction is being carried out as intended by checking if the physical structures match up with the virtual structures. Content enriching • Construction managers can query data from different project management systems as they conduct inspection. Collaboration • Before physical structures go into fabrication, multiple construction managers can review the virtual structures in 3D together to finalize the design. See Figure 36. AR for Constructions [97].
7	Microsoft HoloLens: Design spaces in real-world context with Microsoft Layout [121]	Industrials	Hard-to-simulate environment; Collaboration;	Hard-to-simulate environment: • A shop floor manager can design and assess a new shop floor layout before the physical machines are purchased, transported, and installed on the shop floor. Collaboration • The shop floor manager can call a colleague who then views the virtual layout design as live video feed streaming from the shop floor manager's HoloLens and give feedback on the virtual layout design.  *Figure 57. AR in Layout Design [121]*

#	Use Case	Sector	Process Characteristics	Use Case Description
8	Microsoft HoloLens: Collaborate with Microsoft Remote Assist to solve problems faster [100]	Industrials	Content enriching; Collaboration;	Content enriching • A local worker is attempting to troubleshoot an electrical equipment and pulls up manual instructions as viewed in 3D in her field of view so that she can cross reference the error codes in the manual instructions and the error codes shown on the physical equipment. Collaboration • A local worker is attempting to troubleshoot an electrical equipment and calls in her colleagues working remotely for consultation. The remote colleagues can see what the local worker sees through live camera feed and make live annotations on the camera feed, for example, virtually circling a physical, red cable in front of the local worker while instructing the local worker to inspect that cable. See Figure 39. Collaboration in AR [100].
9-10	Augmented reality training for BMW assembly line workers [122] Laboratory Training with Augmented Reality using Manifest®, Magic Leap, and HoloLens [94]	Industrials	Content enriching; Work instructions;	Content enriching • When a worker goes through a training session, she can access different training materials like images and videos augmented in her view as she is completing a task. Work instructions • As the training session progresses, the worker is presented with step-by-step instructions for each task which she can check off once finished. See Figure 33. AR-Guided Work Instructions [94]
11	Augmented Reality for aircraft maintenance, remote support and training [92]	Industrials	Content enriching; Work instructions; Collaboration;	Work instructions • As an aircraft maintenance worker inspects an aircraft, he can view the step-by-step guidance augmented in 3D Content enriching • As the worker carries on the inspection, he can view information about different aircraft parts in 3D and add log entries to a virtual logbook as he finishes his tasks. Collaboration • Using the HoloLens and Microsoft's Remote Assist, he can call in colleagues who are working remotely for consultation. See Figure 31. AR for Aircraft Maintenance [92].
12	BMW Group Augmented Reality aka AR Prototype Engineering [123]	Industrials	Hard-to-simulate environment;	Hard-to-simulate environment • Car designers can overlay true-to-scale 3D models of car part onto a physical vehicle body during the phototyping phase.  *Figure 58. AR for Product Prototyping [123]*

<table>
<tr><th>#</th><th>Use Case</th><th>Sector</th><th>Process Characteristics</th><th>Use Case Description</th></tr>
<tr><td>13</td><td>Reimagine bridge inspection with Azure Remote Rendering and HoloLens [124]</td><td>Industrials</td><td>Hard-to-simulate environment;</td><td>Hard-to-simulate environment
Traditionally, bridge inspection engineers are tethered to a bridge as they abseil down the bridge (similar to how rock climbers abseil down a boulder) to take pictures of the bridge and detect any fractures. With the use of AR, engineers now can fly drones to take pictures of the bridge, construct a 3D model of the bridge from the pictures, put on an AR headset to view the 3D model, detect and measure any fractures thanks to the high-fidelity of the 3D model.


Figure 59. AR for Bridge Inspection [124]</td></tr>
<tr><td>14</td><td>Augmented Reality for Enabling Smart Nuclear Infrastructure [98]</td><td>Industrials</td><td>Content enriching;

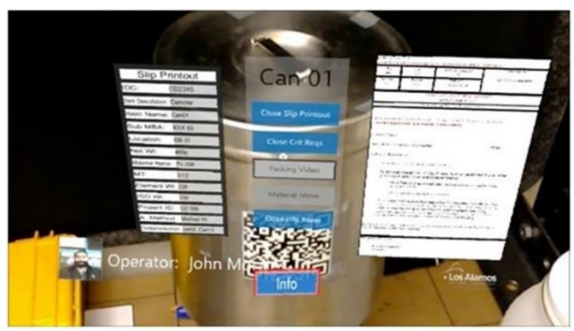
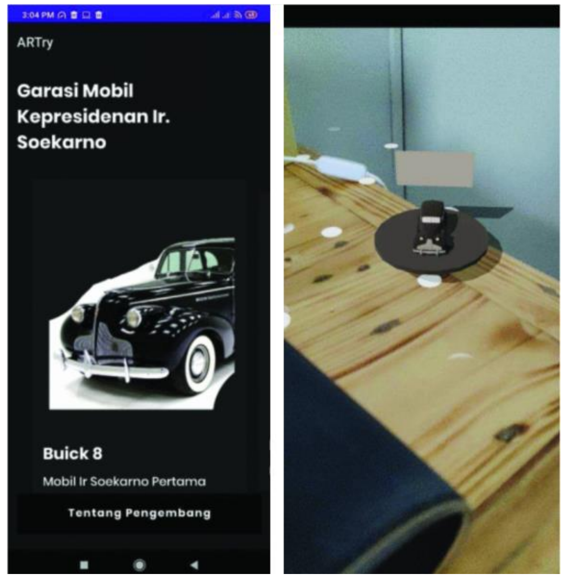
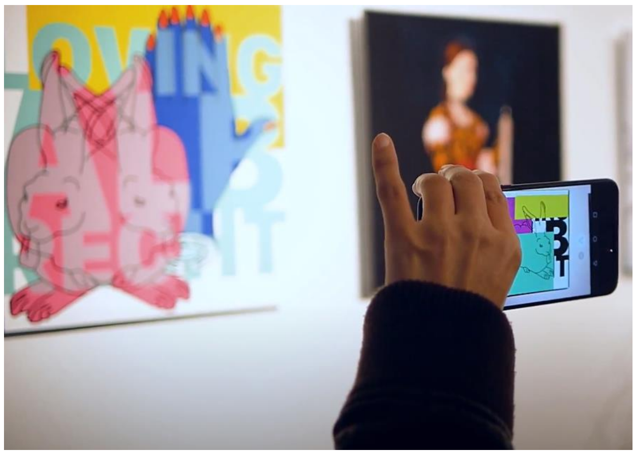
Hands-free control</td><td>Content enriching
When loading a nuclear material canister at a nuclear power plant, a worker can pull up documentation (text and videos) for the handling procedures of the specific canister.


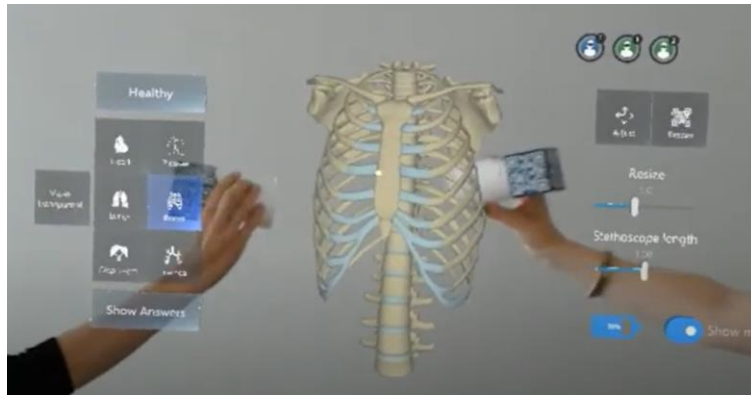
Figure 60. AR for Nuclear Power Plant Operations [98]

Hands-free control
When workers use a glovebox to interact with radioactive materials, they need to scan their hands with a radition monitor every time they pull their hands out of the glovebox. Therefore, two workers typically work together with one worker reading aloud the work instructions while another worker putting his/ her in the glovebox and performing the task as described in the work instructions.
This is an example where workers can benefit from being able to control an AR device hands-free e.g., using voice commands to pull up the work instructions as displayed in 3D while they keep their hands in the glovebox and continue performing their tasks according to the work instructions.</td></tr>
</table>

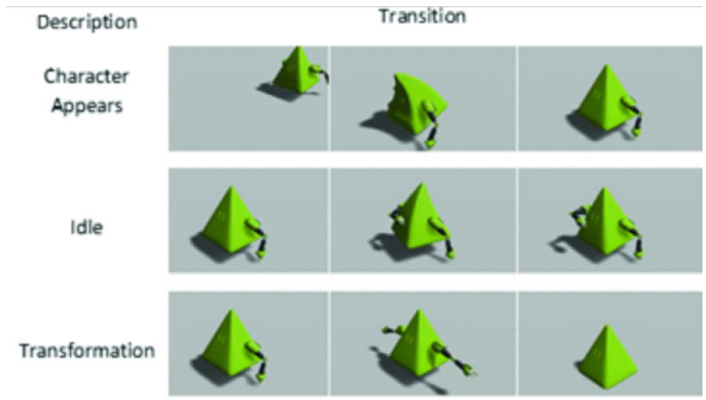
#	Use Case	Sector	Process Characteristics	Use Case Description
16	Coachella and Flume go all in on live concert AR | Spotlight | Unreal Engine [125]	Entertainment & tourism	Hard-to-simulate environment	Hard-to-simulate environment • Through a mobile device, concert goers can see a gigantic, golden 3D bird “floating” above the stage where the artists are performing, creating a visual and immersive effect for the performance.  *Figure 61. AR in Music Concerts [125]*
17	Location based augmented reality game using Kudan SDK [48]	Entertainment & tourism	Hard-to-simulate environment	Hard-to-simulate environment • This game is like the Pokemon Go game discussed in section 1.1 Introduction to Augmented Reality (AR) where users walk around a physical space looking for virtual characters and interact with them.  *Figure 62. AR in Gaming [48]*
18	Design and Development of Augmented Reality (AR) Mobile Application for Malolos' Kameztizuhan (Malolos Heritage Town, Philippines) [126]	Entertainment & tourism	Content enriching	Content enriching • When users scan a postcard, a 3D model is rendered on top of the postcard providing more information about a historical site.  *Figure 63. AR in Entertainment [126]*

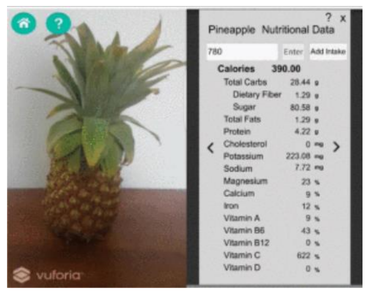
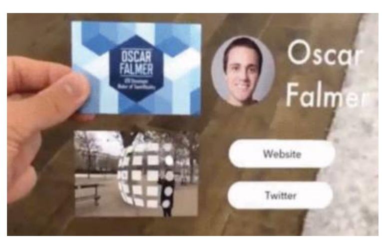
#	Use Case	Sector	Process Characteristics	Use Case Description
19	Design and Development Learning Media Application of Presidential Car History at a Museum in Indonesia using Augmented Reality Technology [127]	Entertainment & tourism	Content enriching	Content enriching • As users scroll through a 2D mobile application providing details about the cars used by previous presidents, the users can turn on AR mode to render a 3D model of the cars in the user's physical environment for a closer look.  *Figure 64. AR in Museums [127]*
20	Real- Time Location based Augmented Reality Advertising Platform [13]	Entertainment & tourism	Content enriching	Content enriching • As users walk around a shopping district, users can use their mobile device to scan different stores and see the stores' signages and other contents in 3D more a more engaging experience.  *Figure 65. AR in Advertising [13]*
21	CoCA Pop-Up (AR)t Book [93]	Entertainment & tourism	Content enriching	Content enriching • 3D model of a sculpture is rendered on top of a physical art book

#	Use Case	Sector	Process Characteristics	Use Case Description
22	Flaneur: Augmented exploration of the architectural urbanscape [128]	Entertainment & tourism	Content enriching	Content enriching • Users can use their mobile device to scan a neighborhood and see 3D markers representing nearby point-of-interests such as restaurants.  *Figure 66. AR for Location Tourism [128]*
23	Augmented Reality Exhibition REALITIES – Vienna Design Week [129]	Entertainment & tourism	Content enriching	Content enriching • Visitors scan a 2D photograph as part of the exhibition to view a video rendered in front of the photograph about the inspiration behind the artist's photograph.  *Figure 67. AR for Art Exhibition [129]*
24	A hospital revolution: VSI HoloMedicine® [95]	Healthcare	Hard-to-simulate environment; Content enriching;	Hard-to-simulate environment • Patient's 2D MRI and CT scans can be transformed into a 3D model to help doctors better assess the patient's anatomy and plan for surgery. Doctor can also explain the upcoming surgery's procedures by walking the patient through the 3D model and keep the patient informed. Content enriching • During patient visits, doctors can pull up the patient's medical records and different test results to discuss any questions from the patients. More importantly, using an AR headset, doctors can access this information hands-free during surgery. See Figure 34. AR for Patient Care [95].

#	Use Case	Sector	Process Characteristics	Use Case Description
25-26	Design and Implementation of “AugMedicine: Lung Cases,” an Augmented Reality Application for the Medical Curriculum on the Presentation of Dyspnea [130] Development of mobile markerless augmented reality for cardiovascular system in anatomy and physiology courses in physiotherapy education [131]	Healthcare	Content enriching	Content enriching • Medical students interact with 3D models of human anatomy for a more engaging and interactive learning compared to traditional methods such as 2D images and slides, plastic models and cadavers.  *Figure 68. AR in Medical Training [132]*
27	IoT Based Augmented Reality System of Human Heart: An Android Application [133]	Healthcare	Content enriching	Content enriching • During a cardiac examination, doctors can show the patients 3D models of the patients’ hearts beating as synchronized to the patients’ real-time heart rate, making it more intuitive for the doctors and the patients to discuss the results of the examination.  *Figure 69. AR for Cardiac Examinations [133]*

#	Use Case	Sector	Process Characteristics	Use Case Description
28	Augmented Reality Guided Laparoscopic Surgery of the Uterus [134]	Healthcare	Navigation	Navigation • During laparoscopic surgeries, surgeons make a small incision through which they insert a laparoscope along with other equipment. The live camera feed from the laparoscope will help guide the surgeons during the surgery. • Doctors are now experimenting with using AR for navigational aid using surgeries: For example, 2D MRI and CT scan images are transformed into 3D models of patient anatomy and tumors. • During surgery, AR tracking engine can detect and track the patient's organs, and AR rendering engine then renders 3D models of tumors as anchored to the patient's organs. When augmenting the live camera feed from the laparoscope with the augmented 3D models of the tumors, surgeons can better locate and remove the tumors.
29	Virtual Lab Using Markerless Augmented Reality [51] Integrating Robo-PEM with AR Application for Introducing Fuel Cell Implementation [135]	Education	Hard-to-simulate environment	Hard-to-simulate environment • Using AR, students can practice their lab exercises without requiring access to the school's physical labs. • For example, the students can set up and configure the parameters of a virtual motor, resistor, ammeter, and voltmeter. Based on the resistor's values, the voltage and current values are calculated based on the logic embedded in the AR application. This use case allows students to continue their learning outside of school or even if they do not have access to expansive lab equipment.  *Figure 70. AR for Virtual Labs [51]*
30	Enhancing STEM Education using Augmented Reality and Machine Learning [57]	Education	Content enriching	Content enriching • When reading a textbook, students can scan images of different learning objects (e.g., a skeleton) and view a 3D model of that object for a more engaging learning experience.  *Figure 71. AR in STEM Education [57]*

#	Use Case	Sector	Process Characteristics	Use Case Description
31	GemAR: Geometry Augmented Reality Application for Elementary School Students [136]	Education	Content enriching	Content enriching • When learning geometries, students can view a 3D human-like character of a specific geometry (e.g., a pyramid shape) dancing around for a more engaging learning experience.  *Figure 72. AR for Math Education [136]*
32	An Evaluation of AR-Assisted Navigation for Search and Rescue in Underground Spaces [59]	Emergency & rescue	Navigation	Navigation • Use AR to guide workers along an evaluation route for underground mines.
33	Microsoft HoloLens: Partner Spotlight with Black Marble [96]	Emergency & rescue	Hard-to-simulate environment; Content enriching;	Hard-to-simulate environment • Police officers can use an AR app to create a 3D map of a crime scene which can be reviewed later by their fellow officers during the investigation. Content enriching • Police offers can pull up case documentations as they investigate the crime scene. Voice memos, images, and videos captured by the officers at the crime screen can be streamed in real-time to the police station for faster processing.
34	Remote Support With Augmented Reality US Army Tank Inspection [137]	Military	Work instructions; Collaboration;	Work instructions + Collaboration • When performing inspection on military tanks, an officer can view step-by-step instructions augmented in 3D and calls a remote officer for consultation.  *Figure 73. AR for Military Tank Inspection [137]*

#	Use Case	Sector	Process Characteristics	Use Case Description
35	Airbus Previews Military Sandbox App for HoloLens [138]	Military	Hard-to-simulate environment;	Hard-to-simulate environment • Military officers view 3D maps of battlefield locations in their physical space for mission planning or training exercises.  *Figure 74. AR for Military Operations [138]*
36	Fruitify: Nutritionally augmenting fruits through markerless-based augmented reality [139]	Miscellaneous	Content enriching	Content enriching • Using their mobile device, users can scan a physical fruit and view the nutritional value for that fruit.  *Figure 75. AR for Nutritional Information [139]*
37	Blind Reader: An Object Identification Mobile- based Application for the Blind using Augmented Reality Detection [35]	Miscellaneous	Content enriching	Content enriching • Using their mobile device, a visually impaired person can scan a physical object. A computer vision algorithm can detect the object and play the sound of the name of the object so that the person can recognize what that object is. This is a form of augmentation through hearing rather than sight like most AR solutions.
38	AR CARD: Interactive Cards using Augmented Reality [140]	Miscellaneous	Content enriching	Content enriching • When scanning a physical business card, a user can view additional information about the person augmented in 3D.  *Figure 76. AR for Business Cards [140]*
39	Google Maps gets improved Live View AR directions [99]	Miscellaneous	Navigation	Navigation • Google Maps renders 3D arrows on top of a street view to better navigate users See Figure 38. Google Maps Live View AR Directions [99].

APPENDIX B: AR PLATFORM DECISION MATRIX

	Selection Criteria	Selection Criteria Description	WebAR (web browser)	Native AR (Android, iOS, etc.)	Varjo XR-3 (2020)	HoloLens 2 (2019)	Magic Leap 2 (2022)	Meta Quest Pro (2022)	Vuzix Blade 2 (2022)	Vuzix M400 (2020)	Google Glass 2 (2019)
1. Usage conditions-specific selection criteria	1. Lighting conditions	Harsh lighting conditions (too bright or too dim) require AR platforms with robust tracking capabilities.	Irrespective of AR platforms, their tracking engine performs better or worse in different light conditions due to the fluctuating quality of video frames being captured in different light conditions, making it easy or hard for the tracking engine's computer vision algorithm to detect a marker in those video frames. To choose an AR platform that best accommodates the target lighting conditions: • Test different candidate AR platforms in different lighting conditions and choose the most performant one • Or: customize the tracking engine's computer vision algorithm • Or: customize the camera's properties to improve the quality of the captured video frames e.g., reduce the camera's exposure level in an overly bright environment to allow less light entering the camera lens [101]								
	2. Network connectivity	Only AR platforms with offline mode support can continue operating when there is no stable network connectivity (e.g., on shopfloors)	Offline mode supported when implementing Service Worker	Offline mode supported	Offline mode supported with offline subscription	Work instructions in Dynamics 365 Guides support offline mode with limitations (e.g., no video calls)	No info about offline mode support	No info about offline mode support	Offline mode supported	Offline mode supported	No info about offline mode support
	3. Noise	AR platforms with audio functionalities (voice commands, calls) must accommodate loud environments (e.g., shopfloors).	Device-specific		No info	5 noise cancelling microphones	No info	No info	2 noise-cancelling microphones	3 noise-cancelling microphones	3 noise cancelling microphones
	4. Operating temperature	Rugged usage conditions with extreme temperature require rugged AR platforms.	Device-specific		No info	10°C to 35°C	10°C to 30°C	No info	0°C to 35°C	-20°C to 45°C	0°C to 35°C
	5. Operating humidity	Rugged usage conditions with extreme humidity require rugged AR platforms.	Device-specific		No info	No info	No info	No info	0% to 95%	0% to 95%	5% to 95%
	6. Drop-safe	Rugged usage conditions require rugged AR platforms which are drop-safe.	Device-specific		No info	No info	No info	No info	No info	MIL-STD-810G certification	No info
	7. Waterproof/ Dustproof	Rugged usage conditions require	Device-specific		No info	IP50 Dust proof	No info	No info	No info	IP67 Dust + water	IP67 Dust + water

Selection Criteria	Selection Criteria Description	WebAR (web browser)	Native AR (Android, iOS, etc.)	Varjo XR-3 (2020)	HoloLens 2 (2019)	Magic Leap 2 (2022)	Meta Quest Pro (2022)	Vuzix Blade 2 (2022)	Vuzix M400 (2020)	Google Glass 2 (2019)
	rugged AR platforms which are waterproof and dustproof.								protection certification	protection certification
8. Compliance with health/ safety standards	AR platforms must comply with health and safety standards e.g., hardhat integration, clean room usage, eye safety.	Device-specific		No info	ANSI Z87.1, CSA Z94.3 and EN 166 certifications for eye safety. Hardhat-integrated; Compliant with regulated environments e.g., clean rooms;	IEC 60601 certifications for medical use	No info	UV protection lens; ANSI Z87.1 for eye safety.;	IEC 60601 certifications for medical use Comes with hardhat mount;	No info
9. Hand usage	Some tasks require users to use both of their hands while some tasks only occupied one hand, allowing users to use their other hand to hold a mobile device (e.g., for webAR and native AR).	One hand occupied to hold mobile device.		Hands-free	Hands-free	Hands-free with one hand-held controller	Hands-free with two hand-held controllers	Hands-free	Hands-free	Hands-free
10. User interactions	Common user interactions for intuitive usage includes voice/ gesture/ gaze commands.	Hand tracking possible when implementing hand tracking engines like Ultraleap and Handtrack.js; Voice commands are possible using speech recognition APIs like Web Speech API (for webAR) or Cordova Speech Recognition API (native AR Android/ iOS);		Eye tracking; Headset and controller tracking;	Voice command; Eye tracking; Headset tracking; Hand tracking;	Eye tracking; Hand tracking; Voice commands; Controller tracking	Eye tracking; Controller tracking; Hand tracking;	Touchpad gestures; Voice control;	Voice control;	Touchpad gestures; voice commands;
11. Movement restrictions	Some tasks require users to have a full range of movement (e.g., shopfloor operators). In that case, tethered AR	Non-tethered		Tethered	Non-tethered	Non-tethered but comes with a wired external, pocket-size processor	Non-tethered	Non-tethered	Non-tethered	Non-tethered

	Selection Criteria	Selection Criteria Description	WebAR (web browser)	Native AR (Android, iOS, etc.)	Varjo XR-3 (2020)	HoloLens 2 (2019)	Magic Leap 2 (2022)	Meta Quest Pro (2022)	Vuzix Blade 2 (2022)	Vuzix M400 (2020)	Google Glass 2 (2019)
		headsets are not an option. Some other tasks require only a limited range of movements.									
2. Hardware-specific selection criteria	1. Battery life	The longer the battery life, the fewer charges are required between usage.	Device-specific		No info	2-3 hours	3.5 hours	1-2 hours	No info	2 hours with external battery possible	8 hours
	2. Device weight	The heavier the device, the more fatigue it causes.	Device-specific		594 g + headband 386 g	566 g	260 g	722 g	No info	182 g	51 g
	3. Field of view – FoV (horizontal)	The wider the FoV, the bigger the virtual contents users can view.	Device's screen size		115°	43°	70°	106°	20°	16.8°	83°
	4. Camera resolution	The higher the camera resolution, the higher quality video the camera can capture. For example, this live camera can be streamed to a remote worker in an AR-guided remote assistance.	Device-specific		12 MP	8 MP	12.6 MP	No info	8 MP	12.8 MP	8 MP
	5. Display resolution (per eye)	The higher the display resolution, the higher fidelity the virtual contents are.	Device-specific		Focus area (27° x 27°) at 1920 x 1920 pixel; Peripheral area at 2880 x 2720 pixel;	1440 x 936 pixel	1440 x 1760 pixel	1800 x 1920 pixel	480 x 480 pixel	854 x 480 pixel	640 x 360 pixel
	6. Refresh rate	The higher refresh rate, the smoother virtual contents appear.	Depending on the implemented tracking and rendering engine		90 Hz	60 Hz	120 Hz	90 Hz	No info	No info	No info
	7. Storage capacity	Storage, computation, and	Device-specific		No info	4GB RAM; 64GB storage;	16GB RAM;	12GB RAM;	40GB storage	6GB RAM; 64GB storage	3GB RAM;

Selection Criteria	Selection Criteria Description	WebAR (web browser)	Native AR (Android, iOS, etc.)	Varjo XR-3 (2020)	HoloLens 2 (2019)	Magic Leap 2 (2022)	Meta Quest Pro (2022)	Vuzix Blade 2 (2022)	Vuzix M400 (2020)	Google Glass 2 (2019)
	remote rendering dictate how much tracking and rendering can be performed locally (no latency) and remotely (requiring less powerful AR devices but results in higher latency).					256GB storage;	256GB storage;			32GB storage
8. Computation capacity		Device-specific		No info	CPU: Octa-core Kryo 385; GPU: Adreno 630	CPU: AMD 7nm Quad-core Zen2 X86 core; GPU: AMD GFX10.2	CPU: Octa-core Kryo 585; GPU: Adreno 650	CPU: Quad Core ARM	CPU: 8 Core 2.52Ghz Qualcomm XR1	CPU: Octa-core Kryo; GPU: Adreno 615
9. Remote rendering		No native integration		Yes – Varjo Reality Cloud	Yes – Azure Remote rendering	Yes – Integrates with NVIDIA CloudXR	No info	No info	No info	No info
10. Cross-device development	Allows developers to develop a webAR, native AR, and AR-on-headset application once and deploy the app on multiple browsers, Android/ iOS devices, and AR headsets respectively.	Likely – Most webAR tracking and rendering engines are built upon browser's natively supported features such as WebGL – see example of browser requirements for 8^{th} Wall tracking engine	Likely – Some tracking engines like Vuforia support build for both Android and iOS. ARCore only supports Android. ARKit only supports iOS – see AR Development Tool Decision Matrix for details	Yes – integrates with OpenXR;	Yes – integrates with OpenXR;	Yes – integrates with OpenXR;	No info	No integration with OpenXR (yet)	No integration with OpenXR (yet)	No integration with OpenXR (yet)
11. App download required	Refers to if a user needs to download and install the AR app to their device.	No	Yes							
12. Heat distribution	Refers to how much heat an AR platform produces while running and how fast the heat can dissipate because of the thermal model of the AR platforms. This factor is important because	It appears that there is no standard procedure for measuring the heat distribution of different AR platforms. Therefore, it is recommended to experiment with different AR platforms for the target usage conditions and select the most comfortable headset through thermal analysis.								

Selection Criteria	Selection Criteria Description	WebAR (web browser)	Native AR (Android, iOS, etc.)	Varjo XR-3 (2020)	HoloLens 2 (2019)	Magic Leap 2 (2022)	Meta Quest Pro (2022)	Vuzix Blade 2 (2022)	Vuzix M400 (2020)	Google Glass 2 (2019)
	heating AR platforms, especially AR headsets, can cause skin burns.									
13. Cost	Refers to the cost of using the AR platforms.	Device-specific		€6.495 + mandatory XR Subscription	€ 3.849,00+ + optional HoloLens subscription	~€3,167+ ($3.299+)	€1.799,99	€1,669.80	€2.311,10	~€1,090 ($1,144)

APPENDIX C: AR DEVELOPMENT TOOL DECISION MATRIX

Legends: **Tracking engine** | **Rendering engine** | **Tracking + Rendering**

	Selection Criteria (SC)	SC Descrip-tion	MindAR	AR.js	WebXR	8th Wall	Lightship ARDK (2021)	Vuforia (2011)	Azure Spatial Anchors	ARCore (2018)	ARKit (2017)	Manifest (2017)	Three.js	A-frame	Unity	Unreal	Reality-Kit	VSI Holo-Medicine	Adobe Aero (2019)	Meta Spark Studio (2017)
	AR platforms supported		WebAR	WebAR	WebAR	WebAR	Native AR (Android, iOS)	Native AR, AR headset (HoloLens 2, Magic Leap 2, Vuzix, RealWear)	Native AR (Android, iOS), AR headset (HoloLens)	Native AR (Android)	Native AR (iOS)	Native AR, AR headset (HoloLens, Magic Leap, Realwear)	WebAR	WebAR	Native AR, AR headset	Native AR, AR headset	Native AR (iOS)	AR Headset (HoloLens)	WebAR (Aero URL); Native AR (Android, iOS)	Native AR (Android, iOS), AR headset (Meta Quest Pro)
1. Tracking capabilities	1. Fiducial tracking	Different use cases require different tracking tech-niques		X				X (VuMark)				No info	N/A	N/A	N/A	N/A	N/A	No info		
	2. Image tracking		X	X	X	X	X	X (Image, cylinder, multi targets)		X	X	No info	N/A	N/A	N/A	N/A	N/A	No info	X (Image anchor)	X (target tracking)
	3. Face tracking		X			X				X	X	No info	N/A	N/A	N/A	N/A	N/A	No info		X
	4. Location tracking			X		X	X			X	X	No info	N/A	N/A	N/A	N/A	N/A	No info		
	5. Object tracking							X (Model targets)			X (Physical objects)	No info	N/A	N/A	N/A	N/A	N/A	No info		
	6. Surface tracking				X (Hit test)	X	X	X (Ground plane)		X (Hit test)	X (Surface detection)	No info	N/A	N/A	N/A	N/A	N/A	No info	X (Surface anchor)	X (plane tracking)
	7. Spatial tracking						X	X (Area targets)	X (Spatial anchors)			No info	N/A	N/A	N/A	N/A	N/A	No info		
	8. Other tracking										Body tracking;	No info	N/A	N/A	N/A	N/A	N/A	No info		Iris tracking; body tracking; Hand tracking;

	Selection Criteria (SC)	SC Descrip-tion	MindAR	AR.js	WebXR	8th Wall	Lightship ARDK (2021)	Vuforia (2011)	Azure Spatial Anchors	ARCore (2018)	ARKit (2017)	Manifest (2017)	Three.js	A-frame	Unity	Unreal	Reality-Kit	VSI Holo-Medicine	Adobe Aero (2019)	Meta Spark Studio (2017)
1. Tracking capabilities	9. Multi-marker tracking		X	X		X (image tracking)		X		X (image tracking)	X (image tracking)	No info	N/A	N/A	N/A	N/A	N/A	No info		X (image tracking; plane tracking)
	10. Multi-user AR session				Possible with tools like Wrapper.js		X	X (Vuforia Chalk remote assist)	X	X (Cloud anchors)	X	No info	N/A	N/A	N/A	N/A	N/A	No info		
	11. Depth tracking				X		X			X	X	No info	N/A	N/A	N/A	N/A	N/A	No info		
	12. Lighting effects				Light estimate;	Light estimate;	Light estimate;				Light estimate; realistic reflection;	No info	N/A	N/A	N/A	N/A	N/A	No info		
2. Rendering capabilities	1. Light estimation		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	X (WebXR)	X (WebXR)	X (ARCore, ARKit)	X (ARCore; ARKit)	X (ARKit)	No info		
	2. Occlusion		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A			X (ARCore, ARKit)	X (ARCore;ARKit)	X (ARKit)	No info		
3. Ease of develop-ment	1. Integration with other AR dev. tools		Three.js; A-frame;	Three.js; A-frame;	Three.js; A-frame;	Three.js; A-frame;	Unity;	MRTK; Magic Leap OS; ARCore/ ARKit;	Unity; ARCore;	Vuforia; Unity; Unreal; WebXR;	Vuforia; Unity; Unreal;	Unity;	WebXR;	WebXR;	OpenXR; MRTK; ARCore; ARKit;	OpenXR; MRTK; ARCore; ARKit;	ARKit	No info		
	2. Simulation/ remote testing				X (WebXR API Emulator)		X (Mock mode)	X (Play mode)		X (Android emulator)		No info			X (AR Foundation Remote for ARCore+ ARKit)	X (In-editor testing)		No info		X (simulator)

	Selection Criteria (SC)	SC Descrip-tion	MindAR	AR.js	WebXR	8th Wall	Lightship ARDK (2021)	Vuforia (2011)	Azure Spatial Anchors	ARCore (2018)	ARKit (2017)	Manifest (2017)	Three.js	A-frame	Unity	Unreal	Reality-Kit	VSI Holo-Medicine	Adobe Aero (2019)	Meta Spark Studio (2017)
3. Ease of develop-ment	3. Low-code/ no-code development							X (Vuforia Chalk, Expert Capture, Instruct, Studio)				X (Manifest Maker)						No info	X	X
4. Licensing cost	1. Open-source		X	X	X								X	X						

APPENDIX D: INTERVIEW PROTOCOL

#	Section	Description
1	General introduction: Purpose of the interview + interviewer/ interviewee introduction	About the author's background and the thesis. Interviewee's name, role, and a brief description of the current role.
2	The ARIM overview	The author explains the ARIM to the interviewees: • High-level design of the ARIM: 5 phases; activities; outcomes • Detailed design of the ARIM: Discovery, Design, Development, Deployment, and Validation phase
3	Interview questions: Part 1 – High-level design of the ARIM	• Are you aware of any existing implementation methodology for AR at your organization? • Do the high-level activities and outcomes in the ARIM capture the essence of an AR project? • Can you suggest some improvements?
4	Interview questions: Part 2 – Discovery phase	• What strategies do you use to assess whether a process is a good candidate for AR? How do you find the ARIM helpful in accomplishing this task? • Can you suggest some improvements?
5	Interview questions: Part 3 – Design phase	• What strategies do you use to choose between different AR platforms and AR development tools? How do you find the ARIM helpful in accomplishing those tasks? • Can you suggest some improvements?
6	Interview questions: Part 4 – Development phase	• What is some general guidance do you have for the development of AR solutions? How do you find the guidance proposed in the ARIM aligned with that guidance? • Can you suggest some improvements?
7	Interview questions: Part 5 – Deployment phase	• What is some general guidance do you have for the development of AR solutions? How do you find the guidance proposed in the ARIM aligned with that guidance? • Can you suggest some improvements?
8	Interview questions: Part 6 – Validation phase	• What strategies do you use to validate that the AR solutions meet the user case requirements? How do you find the ARIM helpful in accomplishing this task? • Can you suggest some improvements?

9	Interview questions: Part 7 – Evaluation scores	• From 1 (strongly disagree) to 5 (strongly agree), how would you rank this statement: The overall design of the ARIM is easy to understand and aligns with the current practice of my organization. • From 1 to 5: This ARIM is useful in identifying potential use cases for AR (RQ#9.1; i.e., guidance for the "Assess process suitability for AR" activity of the methodology in Figure 46). • From 1 to 5: This ARIM is useful in choosing between AR platforms (RQ#9.2; i.e., guidance for the "Assess AR platform suitability" activity the methodology in Figure 46). • From 1 to 5: This ARIM is useful in choosing between AR development tools (RQ#9.3; i.e., guidance for the "Assess AR development tool suitability" activity of the methodology in Figure 46). • From 1 to 5: This ARIM is useful in guiding the development, deployment, and validation of AR solutions (RQ#9.4-9.5; i.e., guidance for the "Develop", "Deploy", and "Validate" phase of the methodology in Figure 46). • From 1 to 5: This ARIM would be useful to me when delivering AR solutions to clients.

REFERENCES

[1] A. B. Craig, "Understanding augmented reality: Concepts and applications," 2013.

[2] X. Qiao, P. Ren, S. Dustdar, L. Liu, H. Ma, and J. Chen, "Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights," *Proceedings of the IEEE,* vol. 107, no. 4, pp. 651-666, 2019, doi: 10.1109/JPROC.2019.2895105.

[3] C. Berger and M. Gerke, "Comparison of Selected Augmented Reality Frameworks for Integration in Geographic Citizen Science Projects," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 43, pp. 223-230, 2022.

[4] S. Schauer, J. Letellier, and J. Sieck, "Augmentation of Printed Content with Web-based Technologies," in *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 22-25 Sept. 2021 2021, vol. 1, pp. 252-256, doi: 10.1109/IDAACS53288.2021.9660839.

[5] S. Eismann *et al.*, "The State of Serverless Applications: Collection, Characterization, and Community Consensus," *IEEE Transactions on Software Engineering,* pp. 1-1, 2021, doi: 10.1109/TSE.2021.3113940.

[6] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," Technical report, ver. 2.3 ebse technical report. ebse, 2007.

[7] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," in *Telemanipulator and telepresence technologies*, 1995, vol. 2351: Spie, pp. 282-292.

[8] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoper. Virtual Environ.,* vol. 6, no. 4, pp. 355–385, 1997, doi: 10.1162/pres.1997.6.4.355.

[9] I. Virag, L. Stoicu-Tivadar, and E. Amăricăi, "Browser-based medical visualization system," in *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 15-17 May 2014 2014, pp. 355-359, doi: 10.1109/SACI.2014.6840092.

[10] N. Ranasinghe and E. Y.-L. Do, "Digital lollipop: Studying electrical stimulation on the human tongue to simulate taste sensations," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),* vol. 13, no. 1, pp. 1-22, 2016.

[11] D. Amin and S. Govilkar, "Comparative study of augmented reality SDKs," *International Journal on Computational Science & Applications,* vol. 5, no. 1, pp. 11-26, 2015.

[12] C. Koch, M. Neges, M. König, and M. Abramovici, "Natural markers for augmented reality-based indoor navigation and facility maintenance," *Automation in Construction,* vol. 48, pp. 18-30, 2014/12/01/ 2014, doi: https://doi.org/10.1016/j.autcon.2014.08.009.

[13] B. I. Batuwanthudawa and K. P. N. Jayasena, "Real- Time Location based Augmented Reality Advertising Platform," in *2020 2nd International Conference on Advancements in Computing (ICAC)*, 10-11 Dec. 2020 2020, vol. 1, pp. 174-179, doi: 10.1109/ICAC51239.2020.9357261.

[14] CIMdata. "PTC Expands Spatial Computing Capabilities with Vuforia Engine Area Targets." CIMdata. https://www.cimdata.com/en/industry-summary-articles/item/15385-ptc-expands-spatial-computing-capabilities-with-vuforia-engine-area-targets (accessed 16/10/2022.

[15] M. Lorenz, S. Knopp, and P. Klimant, "Industrial Augmented Reality: Requirements for an Augmented Reality Maintenance Worker Support System," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 16-20 Oct. 2018 2018, pp. 151-153, doi: 10.1109/ISMAR-Adjunct.2018.00055.

[16] R. Timchenko, O. Grechnyev, S. Skuratovskyi, Y. Chyrka, and I. Gorovyi, "Augmented Reality in Web: Results and Challenges," in *2020 IEEE Third International Conference*

on Data Stream Mining & Processing (DSMP), 21-25 Aug. 2020 2020, pp. 211-216, doi: 10.1109/DSMP47368.2020.9204240.

[17] D. Aschenbrenner *et al.*, "Comparing Different Augmented Reality Support Applications for Cooperative Repair of an Industrial Robot," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 16-20 Oct. 2018 2018, pp. 69-74, doi: 10.1109/ISMAR-Adjunct.2018.00036.

[18] S. Xiang *et al.*, "IV-WA: An Information Visual Tool with WebAR," in *2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, 20-22 Aug. 2021 2021, pp. 353-361, doi: 10.1109/PRAI53619.2021.9550794.

[19] H. Wang, J. Qin, and F. Zhang, "A new interaction method for augmented reality based on ARToolKit," in *2015 8th International Congress on Image and Signal Processing (CISP)*, 14-16 Oct. 2015 2015, pp. 578-583, doi: 10.1109/CISP.2015.7407945.

[20] J. t. Wang, C. N. Shyi, T. W. Hou, and C. P. Fong, "Design and implementation of augmented reality system collaborating with QR code," in *2010 International Computer Symposium (ICS2010)*, 16-18 Dec. 2010 2010, pp. 414-418, doi: 10.1109/COMPSYM.2010.5685477.

[21] K. Group. "OpenGL Overview." https://www.khronos.org/opengl/ (accessed 15/10/2022.

[22] P. Seipel, A. Stock, S. Santhanam, A. Baranowski, N. Hochgeschwender, and A. Schreiber, "Speak to your Software Visualization—Exploring Component-Based Software Architectures in Augmented Reality with a Conversational Interface," in *2019 Working Conference on Software Visualization (VISSOFT)*, 30 Sept.-1 Oct. 2019 2019, pp. 78-82, doi: 10.1109/VISSOFT.2019.00017.

[23] S. Nilsson, "Interaction Without Gesture or Speech -- A Gaze Controlled AR System," in *17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*, 28-30 Nov. 2007 2007, pp. 280-281, doi: 10.1109/ICAT.2007.43.

[24] R. Masoni *et al.*, "Supporting Remote Maintenance in Industry 4.0 through Augmented Reality," *Procedia Manufacturing,* vol. 11, pp. 1296-1302, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.promfg.2017.07.257.

[25] T. A. Mikropoulos *et al.*, "Acceptance and User Experience of an Augmented Reality System for the Simulation of Sensory Overload in Children with Autism," in *2020 6th International Conference of the Immersive Learning Research Network (iLRN)*, 21-25 June 2020 2020, pp. 86-92, doi: 10.23919/iLRN47897.2020.9155113.

[26] S. S. Alam, S. Susmit, C.-Y. Lin, M. Masukujjaman, and Y.-H. Ho, "Factors Affecting Augmented Reality Adoption in the Retail Industry," *Journal of Open Innovation: Technology, Market, and Complexity,* vol. 7, no. 2, p. 142, 2021. [Online]. Available: https://www.mdpi.com/2199-8531/7/2/142.

[27] A. Harikrishnan, A. Said Abdallah, S. K. Ayer, M. El Asmar, and P. Tang, "Feasibility of augmented reality technology for communication in the construction industry," *Advanced Engineering Informatics,* vol. 50, p. 101363, 2021/10/01/ 2021, doi: https://doi.org/10.1016/j.aei.2021.101363.

[28] K. E. Schein and P. A. Rauschnabel, "Augmented Reality in Manufacturing: Exploring Workers’ Perceptions of Barriers," *IEEE Transactions on Engineering Management,* pp. 1-14, 2021, doi: 10.1109/TEM.2021.3093833.

[29] Z. I. Bhutta, S. Umm-e-Hani, and I. Tariq, "The next problems to solve in augmented reality," in *2015 International Conference on Information and Communication Technologies (ICICT)*, 12-13 Dec. 2015 2015, pp. 1-4, doi: 10.1109/ICICT.2015.7469490.

[30] T. Masood and J. Egger, "Augmented reality in support of Industry 4.0—Implementation challenges and success factors," *Robotics and Computer-Integrated Manufacturing,* vol. 58, pp. 181-195, 2019/08/01/ 2019, doi: https://doi.org/10.1016/j.rcim.2019.02.003.

[31] J. Egger and T. Masood, "Augmented reality in support of intelligent manufacturing – A systematic literature review," *Computers & Industrial Engineering,* vol. 140, p. 106195, 2020/02/01/ 2020, doi: https://doi.org/10.1016/j.cie.2019.106195.

[32] M. Quandt, A. Ait Alla, L. Meyer, and M. Freitag, *Success Factors for the Development of Augmented Reality-based Assistance Systems for Maintenance Services*. 2017.

[33] J. Jang, Y. Ko, W. S. Shin, and I. Han, "Augmented Reality and Virtual Reality for Learning: An Examination Using an Extended Technology Acceptance Model," *IEEE Access,* vol. 9, pp. 6798-6809, 2021, doi: 10.1109/ACCESS.2020.3048708.

[34] R. Depietro, E. Wiarda, and M. Fleischer, "The context for change: Organization, technology and environment," *The processes of technological innovation,* vol. 199, no. 0, pp. 151-175, 1990.

[35] J. Y. Mambu, E. Anderson, A. Wahyudi, G. Keyeh, and B. Dajoh, "Blind Reader: An Object Identification Mobile- based Application for the Blind using Augmented Reality Detection," in *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, 22-23 Aug. 2019 2019, vol. 1, pp. 138-141, doi: 10.1109/ICORIS.2019.8874906.

[36] H. Yan, W. Liu, X. Xia, Y. Xu, and T. Ssong, "Design Research of Interactive Picture Books of Cultural Education Based on Augmented Reality Technology," in *2021 16th International Conference on Computer Science & Education (ICCSE)*, 17-21 Aug. 2021 2021, pp. 958-962, doi: 10.1109/ICCSE51940.2021.9569391.

[37] C. G. Low and Y. L. Lee, "SunMap+: An intelligent location-based virtual indoor navigation system using augmented reality," in *International Conference on Frontiers of Communications, Networks and Applications (ICFCNA 2014 - Malaysia)*, 3-5 Nov. 2014 2014, pp. 1-6, doi: 10.1049/cp.2014.1395.

[38] B. A. Koca, B. Çubukçu, and U. Yüzgeç, "Augmented Reality Application for Preschool Children with Unity 3D Platform," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 11-13 Oct. 2019 2019, pp. 1-4, doi: 10.1109/ISMSIT.2019.8932729.

[39] K. Kalarat, "Applying Relief Mapping on Augmented Reality," in *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 22-24 July 2015 2015, pp. 315-318, doi: 10.1109/JCSSE.2015.7219816.

[40] C. S. Yusof, T. S. Low, A. W. Ismail, and M. S. Sunar, "Collaborative Augmented Reality for Chess Game in Handheld Devices," in *2019 IEEE Conference on Graphics and Media (GAME)*, 19-21 Nov. 2019 2019, pp. 32-37, doi: 10.1109/GAME47560.2019.8980979.

[41] C. Puspitasari, G. Pangestu, A. Rahayu, and B. I. Al-Abdillah, "AR - Mart: The Implementation of Augmented Reality as a Smart Self-Service Cashier in the Pandemic Era," in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, 28-28 Oct. 2021 2021, vol. 1, pp. 413-417, doi: 10.1109/ICCSAI53272.2021.9609740.

[42] V. Turchenko, I. Kit, O. Osolinskyi, D. Zahorodnia, P. Bykovyy, and A. Sachenko, "IoT Based Modular Grow Box System Using the AR," in *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, 6-9 Oct. 2020 2020, pp. 741-746, doi: 10.1109/PICST51311.2020.9467949.

[43] E. H. Shaltout, A. Afifi, and K. M. Amin, "Augmented Reality Based Learning Environment for Children with Special Needs," in *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, 15-16 Dec. 2020 2020, pp. 1-7, doi: 10.1109/ICCES51560.2020.9334571.

[44] C. Flinton, P. Anderson, H. P. H. Shum, and E. S. L. Ho, "NETIVAR: NETwork Information Visualization based on Augmented Reality," in *2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, 3-5 Dec. 2018 2018, pp. 1-8, doi: 10.1109/SKIMA.2018.8631530.

[45] L. Gao, Z. Gao, F. Wu, S. Liu, and L. Liu, "Research on Visual Monitoring and Auxiliary Maintenance Technology of Equipment Based on Augmented Reality," in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 24-25 Aug. 2019 2019, vol. 1, pp. 93-97, doi: 10.1109/IHMSC.2019.00030.

[46] X. Li, "Mobile Augmented Reality Technology in Museum Cultural and Creative Products Design," in *2021 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 27-28 March 2021 2021, pp. 422-425, doi: 10.1109/ICITBS53129.2021.00110.

[47] G. Botilias, G. Papantzikos, A. Christopoulos, J. Besarat, and C. Stylios, "Designing a Cultural Augmented Reality Application For Educational Purposes," in *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 24-26 Sept. 2021 2021, pp. 1-5, doi: 10.1109/SEEDA-CECNSM53056.2021.9566218.

[48] H. R. Rahman, D. Herumurti, I. Kuswardayan, A. Yuniarti, W. N. Khotimah, and N. B. Fauzan, "Location based augmented reality game using Kudan SDK," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, 31-31 Oct. 2017 2017, pp. 307-310, doi: 10.1109/ICTS.2017.8265689.

[49] C. Zhou and S. Li, "Application of children Artificial Intelligence science popularization books based on Augmented Reality technology," in *2021 International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*, 21-23 May 2021 2021, pp. 22-26, doi: 10.1109/ISAIAM53259.2021.00012.

[50] A. Yulianti, B. P. Andika, and A. Labellapansa, "Application of Batu Belah Batu Bertangkup Folklore In Riau Province With Augmented Reality," in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, 30 Oct.-1 Nov. 2019 2019, pp. 60-64, doi: 10.1109/ICSEC47112.2019.8974761.

[51] M. T. Abhishek, P. S. Aswin, N. C. Akhil, A. Souban, S. K. Muhammedali, and A. Vial, "Virtual Lab Using Markerless Augmented Reality," in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 4-7 Dec. 2018 2018, pp. 1150-1153, doi: 10.1109/TALE.2018.8615245.

[52] Z. Xiong, L. Zhi, and J. Jiang, "The Research of Developing Virtual Jewelry Worn System Based on AR Toolkit," in *2018 13th International Conference on Computer Science & Education (ICCSE)*, 8-11 Aug. 2018 2018, pp. 1-3, doi: 10.1109/ICCSE.2018.8468680.

[53] C. Geiger, L. Oppermann, and C. Reimann, "3D-registered interaction-surfaces in augmented reality space," in *2003 IEEE International Augmented Reality Toolkit Workshop*, 7-7 Oct. 2003 2003, pp. 5-13, doi: 10.1109/ART.2003.1320417.

[54] J. Xu, L. Yang, and M. Guo, "AR Mobile Video Calling System Based on WebRTC API," in *2022 IEEE 5th International Conference on Computer and Communication Engineering Technology (CCET)*, 19-21 Aug. 2022 2022, pp. 110-114, doi: 10.1109/CCET55412.2022.9906395.

[55] I. Permozer and T. Orehovački, "Utilizing Apple’s ARKit 2.0 for Augmented Reality Application Development," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 20-24 May 2019 2019, pp. 1629-1634, doi: 10.23919/MIPRO.2019.8756928.

[56] J. Koller, D. Kemp, and F. Döpper, "Supporting Disassembly in Remanufacturing with Augmented Reality," in *2020 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, 24-27 Nov. 2020 2020, pp. 1-8, doi: 10.1109/ICTMOD49425.2020.9380599.

[57] I. J. X. Ang and K. H. Lim, "Enhancing STEM Education using Augmented Reality and Machine Learning," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, 28-30 June 2019 2019, pp. 1-5, doi: 10.1109/ICSCC.2019.8843619.

[58] Y. S. Yoon, D. M. Kim, and J. W. Suh, "Augmented Reality Services Using Smart Glasses for Great Gilt-bronze Incense Burner of Baekje," in *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, 6-9 Feb. 2022 2022, pp. 1-4, doi: 10.1109/ICEIC54506.2022.9748275.

[59] D. D. C and S. Duzgun, "An Evaluation of AR-Assisted Navigation for Search and Rescue in Underground Spaces," in *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 9-13 Nov. 2020 2020, pp. 1-2, doi: 10.1109/ISMAR-Adjunct51615.2020.00017.

[60] P. Souza-Herod and A. Hamam, "Augmented Reality Shopping Framework," in *SoutheastCon 2021*, 10-13 March 2021 2021, pp. 1-6, doi: 10.1109/SoutheastCon45413.2021.9401926.

[61] N. Yabuki, K. Miyashita, and T. Fukuda, "AR-based visibility evaluation for preserving landscapes of historical buildings," in *2010 IEEE International Symposium on Mixed and Augmented Reality*, 13-16 Oct. 2010 2010, pp. 281-282, doi: 10.1109/ISMAR.2010.5643607.

[62] B. Brinkman and S. Brinkman, "AR in the library: A pilot study of multi-target acquisition usability," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 1-4 Oct. 2013 2013, pp. 241-242, doi: 10.1109/ISMAR.2013.6671785.

[63] M. Grafe, R. Wortmann, and H. Westphal, "AR-based interactive exploration of a museum exhibit," in *The First IEEE International Workshop Agumented Reality Toolkit*, 29-29 Sept. 2002 2002, p. 5 pp., doi: 10.1109/ART.2002.1106945.

[64] C. Ganser, "AR-Model: collaborative building visualization," in *The First IEEE International Workshop Agumented Reality Toolkit*, 29-29 Sept. 2002 2002, p. 2 pp., doi: 10.1109/ART.2002.1106998.

[65] J. Gausenmeier, C. Matysczok, and R. Radkowski, "AR-based modular construction system for automobile advance development," in *2003 IEEE International Augmented Reality Toolkit Workshop*, 7-7 Oct. 2003 2003, pp. 72-73, doi: 10.1109/ART.2003.1320432.

[66] L. Hu, M. Wang, and Z. Song, "A Convenient Method of Video See-Through Augmented Reality Based on Image-Guided Surgery System," in *2013 Seventh International Conference on Internet Computing for Engineering and Science*, 20-22 Sept. 2013 2013, pp. 100-103, doi: 10.1109/ICICSE.2013.27.

[67] D. Bousnina, R. Ejbali, M. Zaied, and C. B. Amar, "A wavelet network speech recognition system to control an augmented reality object," in *2013 9th International Conference on Information Assurance and Security (IAS)*, 4-6 Dec. 2013 2013, pp. 121-124, doi: 10.1109/ISIAS.2013.6947744.

[68] ClickItTech. "Web Application Architecture: The Latest Guide 2022." https://www.clickittech.com/devops/web-application-architecture/ (accessed 16/10/2022.

[69] J. Fulton. "Web Architecture 101." https://medium.com/storyblocks-engineering/web-architecture-101-a3224e126947 (accessed.

[70] "Cloud services - IaaS, CaaS, PaaS, FaaS, SaaS." DevOps Platform. https://www.devopsplatform.co.uk/blogs/cloud-services (accessed 12/10/2022.

[71] Cisco. "What is Cloud Networking?" https://www.cisco.com/c/en/us/solutions/cloud/what-is-cloud-networking.html#~q-a (accessed.

[72] VMWare. "What is a virtual machine?" https://www.vmware.com/topics/glossary/content/virtual-machine.html (accessed 12/10/2022.

[73] I. C. Education. "Containerization." IBM Cloud Education. https://www.ibm.com/nl-en/cloud/learn/containerization (accessed 12/10/2022.

[74] S. McCarty. "A Practical Introduction to Container Terminology." Red Hat. https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction# (accessed 12/10/2022.

[75] VMWare. "What is container orchestration?" VMWare. https://www.vmware.com/topics/glossary/content/container-orchestration.html (accessed 13/10/2022.

[76] "What is runtime environment (RTE)?" underthehoodlearning.com. https://www.underthehoodlearning.com/what-is-runtime-environment/ (accessed 13/10/2022.

[77] I. C. Education. "FaaS (Function-as-a-Service)." https://www.ibm.com/nl-en/cloud/learn/faas#toc-what-is-fa-j2qiBzdQ (accessed 12/10/2022.

[78] AWS. "What is AWS Lambda?" https://docs.aws.amazon.com/lambda/latest/dg/welcome.html (accessed 12/10/2022.

[79] R. Hat. "What is Function-as-a-Service (FaaS)?" https://www.redhat.com/en/topics/cloud-native-apps/what-is-faas (accessed 12/10/2022.

[80] I. C. Education. "DBaaS (Database-as-a-Service)." IBM. https://www.ibm.com/cloud/learn/dbaas (accessed 13/10/2022.

[81] I. Baldini *et al.*, "Serverless computing: Current trends and open problems," in *Research advances in cloud computing*: Springer, 2017, pp. 1-20.

[82] AWS. "Serverless on AWS." https://aws.amazon.com/serverless/ (accessed 12/10/2022.

[83] G. Cloud. "Serverless." https://cloud.google.com/serverless (accessed 12/10/2022.

[84] M. Azure. "Serverless computing." https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-serverless-computing/ (accessed 12/10/2022.

[85] B. Berman and D. Pollack, "Strategies for the successful implementation of augmented reality," *Business Horizons,* vol. 64, no. 5, pp. 621-630, 2021/09/01/ 2021, doi: https://doi.org/10.1016/j.bushor.2021.02.027.

[86] K. Mühlan, K. A. Przybysz, F. Lindner, D. Akrmanová, D. Winkler, and S. Keil, "A Review and Implementation Framework of Industrial Augmented Reality," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 7-10 Sept. 2021 2021, pp. 01-04, doi: 10.1109/ETFA45728.2021.9613426.

[87] M. A. Livingston, "Evaluating human factors in augmented reality systems," *IEEE Computer Graphics and Applications,* vol. 25, no. 6, pp. 6-9, 2005, doi: 10.1109/MCG.2005.130.

[88] (2003). *The Department of Justice Systems Development Life Cycle Guidance Document*. [Online] Available: https://www.justice.gov/archive/jmd/irm/lifecycle/ch1.htm#para1.2

[89] "Design Thinking." Interaction Design Foundation. https://www.interaction-design.org/literature/topics/design-thinking (accessed 2022).

[90] M. Sezdi and Y. Üzcan, "Clean room classification in the operating room," in *2016 Medical Technologies National Congress (TIPTEKNO)*, 27-29 Oct. 2016 2016, pp. 1-4, doi: 10.1109/TIPTEKNO.2016.7863107.

[91] RedHat. "What is a CI/CD pipeline?" https://www.redhat.com/en/topics/devops/what-cicd-pipeline (accessed.

[92] Ova, "Augmented Reality for aircraft maintenance, remote support and training," ed, 2020, p. https://youtu.be/O7dXn9u2WEc.

[93] Wikitude. "CoCA Pop-Up (AR)t Book." https://www.wikitude.com/showcase/coca-popup-art-book/ (accessed 2022).

[94] Taqtile, "Laboratory Training with Augmented Reality using Manifest®, Magic Leap, and HoloLens," ed, 2021, p. https://youtu.be/4Vjajj7ZggA.

[95] apoQlar, "A hospital revolution: VSI HoloMedicine®," ed, 2019, p. https://youtu.be/JeqxBQERp7Q.

[96] M. HoloLens, "Microsoft HoloLens: Partner Spotlight with Black Marble," ed, 2017, pp. https://youtu.be/4UHN8G5BM-8.

[97] M. HoloLens, "Microsoft HoloLens 2: Partner spotlight with VisualLive and NOX Innovations," ed, 2020, p. https://youtu.be/ltGmB2tQyY8.

[98] D. Mascareñas *et al.*, "Augmented Reality for Enabling Smart Nuclear Infrastructure," *Frontiers in Built Environment,* vol. 5, 06/25 2019, doi: 10.3389/fbuil.2019.00082.
[99] F. Lardinois. "Google Maps gets improved Live View AR directions." https://techcrunch.com/2020/10/01/google-maps-gets-improved-live-view-ar-directions/ (accessed.
[100] M. HoloLens, "Microsoft HoloLens: Collaborate with Microsoft Remote Assist to solve problems faster," ed, 2018, p. https://youtu.be/UpmolMrf5HQ.
[101] M. Clothier and M. Bailey, "Overcoming Augmented Reality Tracking Difficulties In Changing Lighting Conditions," 2004.
[102] I. Akyildiz and H. Guo, "Wireless communication research challenges for Extended Reality (XR)," vol. 3, pp. ITU Journal on Future and Evolving Technologies (ITU J-FET), 04/05 2022, doi: 10.52953/QGKV1321.
[103] A. Rud. "Why and How Netflix, Amazon, and Uber Migrated to Microservices: Learn from Their Experience." https://www.hys-enterprise.com/blog/why-and-how-netflix-amazon-and-uber-migrated-to-microservices-learn-from-their-experience/ (accessed.
[104] C. HARRIS. "Microservices vs. monolithic architecture." https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith (accessed.
[105] K. Eby, "Comprehensive Guide to the Agile Manifesto," 2016. [Online]. Available: https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto.
[106] Microsoft. "CI/CD for microservices architectures." https://learn.microsoft.com/en-us/azure/architecture/microservices/ci-cd (accessed.
[107] AWS. "The 6 Pillars of the AWS Well-Architected Framework." https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/ (accessed.
[108] Google. "Google Cloud Architecture Framework: Cost optimization." https://cloud.google.com/architecture/framework/cost-optimization (accessed.
[109] Azure. "Cost optimization design principles." https://learn.microsoft.com/en-us/azure/architecture/framework/cost/principles (accessed.
[110] AWS. "Design principles." https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/design-principles.html (accessed.
[111] Google. "Performance optimization process." https://cloud.google.com/architecture/framework/performance-optimization/process (accessed.
[112] Microsoft. "Performance efficiency principles." https://learn.microsoft.com/en-us/azure/architecture/framework/scalability/principles (accessed.
[113] AWS. "Security foundations." https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/security.html (accessed.
[114] Google. "Security principles." https://cloud.google.com/architecture/framework/security/security-principles (accessed.
[115] Microsoft. "Security design principles." https://learn.microsoft.com/en-us/azure/architecture/framework/security/security-principles (accessed.
[116] Jeff-Coon. "Assisted Reality: What it Is, and How it Will Improve Service Productivity." https://www.ptc.com/en/blogs/service/what-is-assisted-reality (accessed 2022).
[117] S. Kechiche. "5G in Europe: Reflecting on the Progress So Far and Mapping the Future." https://www.ookla.com/articles/5g-europe-mapping-the-future-q1-2022 (accessed.
[118] V. E. K. Aditya and A. K. Dash, "Real-life Applications of integration of Augmented Reality and Internet of Things," in *2022 Second International Conference on Artificial*

Intelligence and Smart Energy (ICAIS), 23-25 Feb. 2022 2022, pp. 1268-1273, doi: 10.1109/ICAIS53314.2022.9742851.

[119] H. Subakti and J. R. Jiang, "Indoor Augmented Reality Using Deep Learning for Industry 4.0 Smart Factories," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 23-27 July 2018 2018, vol. 02, pp. 63-68, doi: 10.1109/COMPSAC.2018.10204.

[120] G. AR, "The future of construction with augmented reality and BIM," ed, 2020, p. https://youtu.be/DzFctc7bkCM.

[121] M. HoloLens, "Microsoft HoloLens: Design spaces in real-world content with Microsoft Layout," ed, 2018, p. https://youtu.be/rK7l6Gq16WA.

[122] Capgemini, "Augmented reality training for BMW assembly line workers," ed, 2019, p. https://youtu.be/V0WrCf8PbEk.

[123] DPCcars, "BMW Group Augmented Reality aka AR Prototype Engineering," ed, 2020, p. https://youtu.be/gYkApnv8QI0.

[124] M. Cloud, "Reimagine bridge inspection with Azure Remote Rendering and HoloLens," ed, 2021, p. https://youtu.be/Ayz9BSAGO1k.

[125] U. Engine, "Coachella and Flume go all in on live concert AR | Spotlight | Unreal Engine," ed, 2022, p. https://youtu.be/DUwkup2C7b0.

[126] D. R. d. Cruz, J. S. A. Sevilla, J. W. D. S. Gabriel, A. J. P. D. Cruz, and E. J. S. Caselis, "Design and Development of Augmented Reality (AR) Mobile Application for Malolos' Kameztizuhan (Malolos Heritage Town, Philippines)," in *2018 IEEE Games, Entertainment, Media Conference (GEM)*, 15-17 Aug. 2018 2018, pp. 1-9, doi: 10.1109/GEM.2018.8516272.

[127] H. N. Hizbullah, M. F. I. Masyhuri, N. A. Santoso, F. I. Maulana, M. Ramadhani, and F. Permana, "Design and Development Learning Media Application of Presidential Car History at a Museum in Indonesia using Augmented Reality Technology," in *2021 International Conference on Information Management and Technology (ICIMTech)*, 19-20 Aug. 2021 2021, vol. 1, pp. 551-555, doi: 10.1109/ICIMTech53080.2021.9534948.

[128] A. Ioannidi, D. Gavalas, and V. Kasapakis, "Flaneur: Augmented exploration of the architectural urbanscape," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 3-6 July 2017 2017, pp. 529-533, doi: 10.1109/ISCC.2017.8024582.

[129] Artivive, "Augmented Reality Exhibition REALITIES – Vienna Design Week," ed, 2018.

[130] A. D. Pieterse, B. P. Hierck, P. G. M. d. Jong, J. Kroese, L. N. A. Willems, and M. E. J. Reinders, "Design and Implementation of “AugMedicine: Lung Cases,” an Augmented Reality Application for the Medical Curriculum on the Presentation of Dyspnea," in *Frontiers in Virtual Reality*, 2020.

[131] N. S. Rosni, Z. A. Kadir, M. N. M. M. Noor, Z. H. A. Rahman, and N. A. Bakar, "Development of mobile markerless augmented reality for cardiovascular system in anatomy and physiology courses in physiotherapy education," in *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 3-5 Jan. 2020 2020, pp. 1-5, doi: 10.1109/IMCOM48794.2020.9001692.

[132] L. U. M. Centrum, "Demonstration of AugMedicine: Lung cases," ed, 2020, p. https://youtu.be/GGiigVMKPTA.

[133] D. Agrawal, S. B. Mane, A. Pacharne, and S. Tiwari, "IoT Based Augmented Reality System of Human Heart: An Android Application," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 11-12 May 2018 2018, pp. 899-902, doi: 10.1109/ICOEI.2018.8553807.

[134] T. Collins *et al.*, "Augmented Reality Guided Laparoscopic Surgery of the Uterus," *IEEE Transactions on Medical Imaging,* vol. 40, no. 1, pp. 371-380, 2021, doi: 10.1109/TMI.2020.3027442.

[135] S. Sendari *et al.*, "Integrating Robo-PEM with AR Application for Introducing Fuel Cell Implementation," in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, 2-2 Oct. 2021 2021, pp. 145-150, doi: 10.1109/ICEEIE52663.2021.9616683.

[136] L. Husniah, Y. B. S. Nugraha, A. S. Kholimi, U. L. Yuhana, E. M. Yuniarno, and M. H. Purnomo, "GemAR: Geometry Augmented Reality Application for Elementary School Students," in *2020 IEEE Graphics and Multimedia (GAME)*, 17-19 Nov. 2020 2020, pp. 25-30, doi: 10.1109/GAME50158.2020.9315086.

[137] Taqtile, "Remote Support with Augmented Reality US Army Tank Inspection," ed, 2022, pp. https://youtu.be/Fe-L1c8kP-0.

[138] A. D. a. Space, "Holographic Tactical Sandbox: Augmented Reality for Mission Preparation," ed, 2019, p. https://youtu.be/TfCVkmAmTqA.

[139] A. Kulpy and G. Bekaroo, "Fruitify: Nutritionally augmenting fruits through markerless-based augmented reality," in *2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 23-24 Nov. 2017 2017, pp. 149-153, doi: 10.1109/ISCMI.2017.8279616.

[140] S. A, D. Gangadhar, J. Mohan, N. T. Mathew, and T. Sreenath, "AR CARD: Interactive Cards using Augmented Reality," in *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 29-30 Oct. 2021 2021, pp. 1-6, doi: 10.1109/SMARTGENCON51891.2021.9645911.