

# Analyse des iPad-Scanner-App Projekts

## Hintergrund und Ziel des Projekts

Das vorliegende Projekt zielt darauf ab, eine **iPad-App zur Raumakustik-Analyse** zu entwickeln. Diese App soll mit Hilfe des LiDAR-Sensors eines iPad Pro Räume dreidimensional erfassen und **akustische Kennwerte** wie Nachhallzeit (RT30/RT60) automatisch berechnen. Zusätzlich sollen **Materialempfehlungen** zur akustischen Optimierung des Raums gegeben werden. Die Motivation dahinter ist klar: Gute Raumakustik verbessert Sprachverständlichkeit, Konzentration und Wohlbefinden, während lange Nachhallzeiten Unbehagen, Produktivitätsverlust und Stress verursachen <sup>1</sup>. Traditionelle Akustikmessungen sind allerdings oft aufwendig und teuer. Ziel ist daher eine **einfache, mobile Lösung** für die Raumakustikanalyse, die **zeit- und kosteneffizient** ist <sup>2</sup>. Dieses Konzept steht im Einklang mit der Philosophie der MSH-Audio-Gruppe, die in ihrem Vertriebshandbuch betont, dass Akustik nicht nur Normerfüllung ist, sondern Räume **hörbar verbessert** werden sollen <sup>3</sup>. <sup>4</sup>.

## Geschäftsmodell und Marktanalyse

**Zielgruppen:** Die geplante App richtet sich primär an Fachleute in der **Bau- und Audio-Branche**, z.B. Ingenieurbüros für Akustik und AV, Architekturbüros sowie Facility Manager. Diese Gruppen stehen häufig vor dem Problem, akustische Messungen und Optimierungen in Räumen durchführen zu müssen, ohne dafür aufwendige Gutachten beauftragen zu wollen. Wie im Konzept festgehalten, zählen insbesondere **AV-Systemintegratoren**, Architekturbüros mit Planungs- oder Forschungsfokus und **Bauherren/Betreiber** zu den potenziellen Nutzern <sup>5</sup>. Sie alle haben einen klaren **Engpass**: Entweder fehlt akustisches Fachwissen, die Zeit für manuelle Modellierung, oder es mangelt an digitalen Bestandsdaten bei Sanierungen <sup>6</sup>. Hier bietet die App einen direkten Mehrwert, indem sie **schnell digitale Akustik-Daten** liefert und Entscheidungshilfen gibt.

**Wertversprechen:** Die App soll eine **schnelle Vor-Ort-Analyse** ermöglichen, inklusive normgerechter Nachhallzeit-Berechnung nach DIN 18041, dokumentiert durch automatische Protokolle mit Diagrammen und Empfehlungen <sup>7</sup> <sup>8</sup>. Damit können Nutzer **Kosten sparen** ("Wir holen Sie früh ins Boot – dann sparen Sie später viel Geld") und fundierte Entscheidungen treffen. Zudem kann die App helfen, komplexe Technikinstallationen (z.B. AV-Anlagen) besser abzustimmen, da **die Technik nur so gut ist wie die Akustik im Raum**, wie es im Vertriebshandbuch formuliert wird <sup>4</sup>. Insgesamt wird ein **Mehrwert durch schnellere Analyse**, Einhaltung von Normen und geringere Kosten gegenüber herkömmlichen Verfahren geschaffen <sup>9</sup>.

**Geschäftsmodell:** Vorgeschlagen wird ein hybrides Modell aus **App-Lizenz und Consulting-Dienstleistungen** <sup>9</sup>. Die Basis-App könnte als **Jahreslizenz (z.B. ~1.500 € pro Jahr)** pro Kunde vertrieben werden, was sich an Preisen ähnlicher spezialisierter Software orientiert. Zusätzlich kann die MSH-Audio-Gruppe begleitende Beratung anbieten (z.B. detaillierte Gutachten, maßgeschneiderte Lösungen), was zusätzlichen Umsatz generiert. Bemerkenswert ist, dass ein **Open-Source-Kern** angedacht ist <sup>9</sup>: Das könnte bedeuten, dass zentrale Algorithmen oder Libraries offen verfügbar sind, um Vertrauen zu schaffen und eine Entwickler-Community einzubinden. Die Monetarisierung würde dann über die App-Integration, Support und Zusatzservices erfolgen. Dieses Modell fördert **Community-Kollaboration**, wovon die Weiterentwicklung profitieren kann <sup>10</sup>.

**Marktbedarf und Trends:** Der Markttrend spricht für eine solche Lösung. Ein großer Teil der Bauprojekte betrifft heute **Bestandsgebäude**, die saniert oder umgebaut werden müssen – Schätzungen zufolge rund 70% der Bauinvestitionen in vielen Regionen <sup>11</sup>. In den USA z.B. haben Renovierungen 2022 erstmals die Neubauten in den Architekten-Billing-Indizes überholt <sup>12</sup>, was den globalen Trend zum Bauen im Bestand widerspiegelt. Zugleich steigt der **Digitalisierungsdruck** im Bauwesen, verstärkt durch Initiativen wie den EU AI Act, der ab 2025 höhere Anforderungen und Chancen für KI-Lösungen schafft <sup>13</sup>. KI-gestützte Verfahren versprechen deutliche Effizienzgewinne – so zeigen Scan-to-BIM-Lösungen wie Aurivus, dass sich durch KI bei der Modellierung von Gebäuden **bis zu 50% der Zeit einsparen** lassen <sup>14</sup>. Dies ist ein Hinweis, dass auch in der Akustikplanung erhebliches Zeitsparpotenzial besteht. Ein weiterer Trend ist der **PropTech-Boom**: Immer mehr Startups kombinieren Bauwesen mit Tech (z.B. generative Planungshelfer, automatisierte Simulationen). Der Markt ist jedoch noch **fragmentiert** – bisher gibt es kein Produkt, das LiDAR-Erfassung, akustische Simulation, Bewertung und Empfehlung in einer Lösung vereint <sup>15</sup>. Bestehende Tools decken meist nur Teilaspekte ab, wie **Scan-to-BIM** (Geometrie erstellen, z.B. aurivus), **CAFM** (Facility Management mit Bestandsdaten) oder **raumakustische Simulation** am PC. Diese Fragmentierung bietet Raum für eine integrierte Lösung, die mehrere Schritte kombiniert <sup>16</sup>. Zudem steigt der **Sanierungsbedarf** (z.B. ein Drittel aller Büroflächen gelten als sanierungsbedürftig laut Branchenreports) – was den Bedarf an schnellen akustischen Bewertungen erhöht. Insgesamt zeigt die Marktanalyse, dass eine mobile Akustik-App auf fruchtbaren Boden trifft, sofern sie zuverlässig funktioniert.

**Wettbewerbs- und Umfeldanalyse:** Im direkten Wettbewerb gibt es einige Referenzpunkte: - **Raum-Scan-Apps:** Es existieren bereits LiDAR-Scan-Apps wie **Polycam** oder **RoomScan**, und speziell in der Akustik z.B. **CadnaR.scan** von Datakustik. CadnaR.scan ermöglicht es, per iPhone/iPad einen Raum zu scannen und das Modell in die professionelle Akustiksoftware CadnaR zu importieren <sup>17</sup> <sup>18</sup>. Dies bestätigt die technische Machbarkeit und dass Bedarf gesehen wird, jedoch erfordert die Auswertung weiterhin eine Desktop-Software. Kein mir bekanntes Produkt jedoch führt direkt auf dem Tablet eine vollständige **Akustikanalyse samt Empfehlungen** durch. - **Akustik-Mess-Apps:** Es gibt Apps wie **Studio Six "AudioTools"** oder **NTi Audio Acoustilyzer**, die Nachhallzeitmessungen und andere Akustik-Messungen mobil durchführen können. Diese sind allerdings oft nur Tools zur Messung, ohne die Integration von 3D-Raumdaten. - **Simulationstools & Plugins:** In der Forschung und Open-Source-Community existieren Libraries wie *Pyroomacoustics* oder *Room Acoustics C++*, die raumakustische Simulationen (z.B. Image-Source-Methoden, Raytracing oder FDTD) durchführen können. Solche Tools sind allerdings **für Endnutzer schwer zugänglich** und erfordern einiges an Expertise <sup>19</sup>. Hier setzt unser Ansatz an, indem diese wissenschaftlichen Bibliotheken in einer nutzerfreundlichen App verborgen im Hintergrund arbeiten.

Das Alleinstellungsmerkmal (USP) der geplanten App wäre also eine **integrierte End-to-End-Lösung**: Vom Raumscan über akustische Berechnung bis zur konkreten Handlungsempfehlung, alles in *einer* mobilen Anwendung. Dies spart nicht nur Zeit, sondern auch Schnittstellenprobleme zwischen verschiedenen Tools. Laut den ausgearbeiteten Konzeptfolien existiert derzeit **kein Produkt**, das LiDAR-Erfassung, akustische Simulation, Bewertung und Materialempfehlung in einem bietet <sup>20</sup> – eine klare Marktlücke und Chance.

**Marktpotenzial:** Betrachtet man den adressierbaren Markt (TAM/SAM), so gibt es allein in Deutschland über 30.000 Architektur- und Ingenieurbüros <sup>21</sup>, von denen ein relevanter Prozentsatz regelmäßig mit Raumakustik zu tun hat (Schätzung: z.B. 30% = SAM). Selbst wenn nur 5% davon early adopters wären (SOM), wären das einige hundert potenzielle Kunden in DACH. Bei einem Lizenzpreis von z.B. 1.500 € jährlich ergibt sich ein substantieller Umsatzspielraum. Zusätzlich könnte das Angebot auf **internationalen Markt** ausgedehnt werden, da das Problem (schnelle Akustikanalyse) weltweit besteht und die Software (bis auf die Sprache und Normen) übertragbar wäre.

**Strategie:** Empfohlen wird, mit einem klar umrissenen Kern-Feature zu starten (**Bottom-up-Ansatz**) und dann modular zu erweitern <sup>22</sup>. Das heißt, zunächst die **Kerntätigkeit – Nachhallzeit messen und vergleichen** – perfekt umzusetzen, da dies den dringendsten Engpass vieler Nutzer adressiert (schnelle Bestandsaufnahme der Raumakustik). Ist dieser Kern erfolgreich, kann iterativ um weitere Funktionen (Materialdatenbank, Simulation, AR-Visualisierung) ausgebaut werden. Diese Fokussierung folgt der sogenannten **EKS-Strategie** (Engpasskonzentrierte Strategie): Erst das drängendste Problem einer spitzen Zielgruppe lösen, dann auf dieser Basis skalieren <sup>23</sup>. Visionär gedacht, könnte das System mit jedem Projekt dazulernen (Machine Learning) und so die Prognosegenauigkeit immer weiter steigern <sup>24</sup>.

## Produktfunktionalität und technische Umsetzung

Die bereits entwickelte **Prototyp-App** zeigt grundlegende Funktionen einer **Scanner-App** auf dem iPad. Im Folgenden werden die wichtigsten *funktionalen Anforderungen* (Was soll die App leisten?) und *nicht-funktionalen Anforderungen* (Qualitätskriterien) skizziert – quasi ein Lastenheft-Entwurf für die Swift-Entwicklung. Ebenso betrachten wir eingesetzte Technologien und den bisherigen Entwicklungsstand aus technischer Sicht.

### Funktionale Kernfunktionen der App

- **LiDAR-Raumscan:** Die App nutzt die LiDAR-Kamera des iPad Pro, um **automatisch Geometrie und Volumen des Raumes** zu erfassen <sup>7</sup>. Apple bietet hierfür mit **ARKit/RoomPlan** bereits eine leistungsfähige API, die das Kamerabild und den LiDAR-Scanner kombiniert, um einen 3D-Grundriss inklusive Dimensionen und erkannten Objekten zu erzeugen <sup>25</sup>. Dieser Scan liefert die notwendigen Raumparameter (Raumvolumen, Flächen der reflektierenden Flächen wie Wände/Decke/Boden). Die Prototype-App sollte überprüfen, ob sie bereits RoomPlan einbindet – dies wäre empfehlenswert, um genaue und schnelle Scans zu erzielen. CadnaR.scan zeigt, dass solche Scans auf dem iPad zu vereinfachten **3D-Raumdaten** führen, inkl. Wänden und großen Objekten <sup>26</sup> <sup>27</sup>. In der App sollte nach dem Scan eine Visualisierung (z.B. Drahtgittermodell oder "Dollhouse"-Ansicht) dem Nutzer sofort Rückmeldung geben, ob der Raum vollständig erfasst ist (ähnlich wie RoomPlan's Coaching UI mit Echtzeit-Feedback <sup>28</sup>).
- **Nachhallzeit-Berechnung (Modell):** Aus den erfassten Raumdaten berechnet die App zunächst eine **theoretische Nachhallzeit** mittels bekannter akustischer Formeln (Sabine oder Eyring). Hierzu wird eine **Absorptionsdatenbank** hinterlegt, die für typische Materialien (z.B. Betonwand, Gipsdecke, Teppichboden) die Schallabsorptionsgrade pro Frequenzband enthält <sup>29</sup>. Basierend auf der Norm **DIN 18041** („Hörsamkeit in Räumen“) kann so je nach Raumvolumen und Nutzung eine Soll- und Ist-Nachhallzeit ermittelt werden. Die Sabine-Formel liefert einen ersten Anhaltswert für RT60 basierend auf Gesamtvolumen und äquivalentem Absorptionsflächenmaß <sup>30</sup>. Diese Berechnung erfolgt **offline auf dem Gerät** – dafür bietet sich z.B. die Integration einer Open-Source-Bibliothek wie *Pyroomacoustics* an, die Algorithmen zur RT60-Berechnung bereitstellt. (Anmerkung: Da Pyroomacoustics in Python implementiert ist, müsste man entweder die Formel direkt in Swift nachimplementieren oder über C/C++-Bindings arbeiten. Alternativ sind die Sabine-Formel und einfache Iterationen auch manuell in Swift codebar, zumal es primär um Summation von  $A = \alpha \cdot S$  geht.)
- **Reale Messung via Mikrofon:** Ein Alleinstellungsmerkmal ist die **tatsächliche akustische Messung** im Raum als Kalibrierung. Die App soll ein **Messmikrofon** (idealerweise ein kalibriertes externes Mikrofon über USB/Lightning) unterstützen <sup>8</sup>. Der Nutzer würde im Raum z.B. einen **Impuls** erzeugen (Knall, Klatschen oder besser ein definierter Lautsprechersweep). Die App

zeichnet die **Impulsantwort** des Raumes auf und berechnet daraus die **Nachhallzeit T30/T60** nach Standardverfahren (Abfall der Schallpegelkurve um 30 dB extrapoliert) <sup>31</sup>. Wichtig ist eine Kalibrierungsroutine: z.B. erst Umgebungsgeräusch-Pegel messen, dann sicherstellen, dass der Impuls laut genug ist, ggf. **Mikrofon-Kalibrierung** (Empfindlichkeit) zulassen <sup>31</sup>. Diese gemessene RT30 kann mit der berechneten verglichen werden, um Abweichungen festzustellen (z.B. deutliche Abweichungen deuten auf komplexe Geometrie oder Frequenzphänomene hin, die Sabine nicht abbildet). Für die Umsetzung kann Apple's **AVAudioEngine** oder **AudioKit** verwendet werden, um das Mikrofon-Signal in hoher Samplerate aufzuzeichnen und die Impulsantwort zu berechnen. FFT und Schroeder-Integration für Nachhallzeit sind nötig; hier gibt es teils Beispielcode oder man greift auf bewährte Formeln zurück <sup>32</sup> (nach DIN 3382 erfordert das in der Praxis meist einen Sweep-Sinus als Anregung und Obertöne-Analyse, aber für einen Prototyp reicht evtl. auch ein einfacher Hand Clap in kleineren Räumen).

- **Ergebnis-Report & Dokumentation:** Die App generiert **automatisierte Berichte** mit allen relevanten Ergebnissen <sup>8</sup>. Dazu gehören: Graphen der Nachhallkurven (Abklingkurve), berechnete vs. gemessene Nachhallzeit pro Frequenzband, identifizierte Problemfrequenzen, und Fotos oder Screenshots des erfassten Raums (z.B. der 3D-Scan, oder vom iPad aufgenommene Fotos der Wände). Dieser Report sollte exportierbar sein, etwa als **PDF** oder als HTML. Für Business-Kunden wichtig: das Protokoll sollte **normgerecht** sein, um es z.B. in Dokumentationen einzufügen. Evtl. könnte man DIN-18041-Grenzwerte grafisch mit darstellen (z.B. Sollwert-Bereich der RT60 je nach Raumvolumen). Der **Prototyp** hat vermutlich rudimentäre Berichtsfunktionen; diese gilt es auszubauen, etwa durch SwiftUI PDF rendering oder serverseitig (wenn Cloud vorgesehen, was aber aus Datenschutzgründen eher vermieden werden soll – daher lieber on-device PDF).
- **Empfehlung von Maßnahmen:** Aus der Differenz zwischen Soll- und Ist-Nachhallzeit leitet die App **Materialempfehlungen** ab <sup>33</sup>. Konkret: Wenn z.B. die Nachhallzeit zu hoch ist, schlägt die App Materialien vor, um die Absorption zu erhöhen – z.B. Deckensegel, Wandabsorber, Teppich – inklusive einer Abschätzung, wie viel Fläche davon nötig wäre, um die Nachhallzeit in den Zielbereich zu bringen. Die Materialdatenbank (Absorptionskoeffizienten) liefert hierzu die Grundlage. Der Nutzer könnte interaktiv angeben: "*Raumtyp: Klassenzimmer*" – und die App wählt passende Empfehlungen (z.B. schallabsorbierende Deckenplatten, Wandpaneele aus PET-Fasern etc.). Eine **visuelle Komponente** wäre wünschenswert: eventuell kann die App im gescannten 3D-Modell zeigen, an welchen Flächen die Maßnahmen angebracht werden könnten (Augmented Reality Vorschau). Dies ist eher ein Erweiterungsfeature, aber sehr **innovativ und verkaufsfördernd**. Im Prototyp-Lastenheft ist die Empfehlungskomponente erwähnt <sup>34</sup>, also sollte sie von Anfang an konzipiert werden, auch wenn die MVP-Version evtl. nur textuelle Empfehlungen ausgibt.
- **Simulation und Auralisation:** Ein visionäres Feature ist die **Echtzeit-Simulation** akustischer Maßnahmen. In Zukunft könnte die App mittels eingebauter KI und DSP **Hörbeispiele (Auralisation)** erzeugen <sup>24</sup>. Das heißt, der Nutzer könnte über Kopfhörer hören, wie der Raum nach einer Maßnahme (z.B. mit X m<sup>2</sup> Absorbern) klingen würde. Technisch geht das, indem man die Impulsantwort des Raums modifiziert (verkürzt) und einen Audiobeispiel (Sprache, Musik) falten (Faltungshall). Sogar eine Integration als **VST-Plugin** wurde angedacht <sup>35</sup>, um in Audio-Workstations den Raumklang zu simulieren. Für den MVP ist Auralisation optional, aber die Architektur sollte dafür offen sein (z.B. modulare Audio-Engine). Die Nennung dieses Features zeigt aber den **hohen Innovationsanspruch** und kann im Business-Plan als Alleinstellungsmerkmal kommuniziert werden.

- **Datenmanagement und Export:** Alle erfassten Daten (Scans, Messungen, Berechnungen) sollten speicher- und abrufbar sein. Eine Projektübersicht (Liste von Scans, ähnlich CadnaR.scan<sup>36</sup>) erleichtert das **Verwalten mehrerer Räume**. Exportformate können neben PDF-Bericht auch **3D-Modell-Exporte** umfassen (z.B. USDZ, um den Raum in CAD weiterzubearbeiten, oder das akustische Modell für andere Software). CadnaR.scan exportiert z.B. ein .EOX-Format zur direkten Weiterverarbeitung in CadnaR<sup>37</sup>. Unsere App könnte ggf. einen **Wave-File-Export** der gemessenen Impulsantwort anbieten (für akustische Fachleute zur weiteren Analyse in z.B. MATLAB oder ODEON).

Die obigen Funktionen entsprechen weitgehend dem in den Konzeptfolien skizzierten **Lastenheft-Auszug**<sup>34</sup>. Zusammengefasst müssen also folgende Kern-Features umgesetzt werden: *LiDAR-Scan, theoretische Nachhallberechnung, Messung mit Mikrofon, Reporting, Empfehlungen, (später Simulation)*. Dies deckt sich mit den vorhandenen Plänen.

## Nicht-funktionale Anforderungen

Neben den Funktionen sind Qualitätskriterien entscheidend, damit die App in der Praxis akzeptiert wird:

- **Genauigkeit:** Die geometrische Erfassung muss ausreichend präzise sein. Apple gibt an, dass RoomPlan sehr genaue Abmessungen liefern kann (innerhalb weniger Zentimeter)<sup>25</sup>. Für akustische Zwecke sollte die Volumengenauigkeit im Bereich <5% Abweichung liegen, da sonst RT60-Berechnungen leiden. Die Messung der Nachhallzeit sollte möglichst **normgerecht** sein (DIN 3382), also genug Dynamik (>35 dB Abklingbereich) und im mittleren Frequenzbereich gemittelt. Eine Herausforderung: **Kalibrierung** – da unterschiedliche Mikros und Geräuschpegel die Messung beeinflussen, muss die App hier Robustheit bieten (evtl. Messwiederholungen erlauben, Warnung bei Störgeräuschen usw.).
- **Performance:** Die App muss auf iPad Hardware flüssig laufen. LiDAR-Scanning und Echtzeit-Mapping sind rechenintensiv, werden aber von ARKit gut optimiert auf dem Neural Engine Chip durchgeführt. Die akustische Berechnung (Sabine-Formel) ist trivial schnell; die Auswertung der Impulsantwort erfordert FFT und kann im Millisekundenbereich erledigt werden. Solange keine vollwertige 3D-Simulation (Raytracing) on-device gemacht wird, bleibt das im Rahmen. Dennoch sollte darauf geachtet werden, effiziente Algorithmen zu nutzen und große Daten (z.B. Punktwolken) nicht unnötig lange vorzuhalten. **Speicherverwaltung** ist wichtig, damit auch komplexere Räume nicht zum Absturz führen.
- **Usability (Benutzerfreundlichkeit):** Die Zielnutzer sind Ingenieure, Architekten, Facility Manager – meist keine IT-Experten, aber auch keine Laien. Die App sollte eine **intuitive UI** haben, idealerweise mit **SwiftUI** gestaltet (modern, flexibel). Ein klarer Scan-Workflow (ggf. mit Schritt-für-Schritt-Assistent), deutliche Darstellung der Ergebnisse (Graphen, Ampel-Indikator ob Nachhall im grünen Bereich) und einfache Navigation sind entscheidend. Eine lokalisierte Oberfläche (Deutsch/Englisch zumindest) wäre für den Markt wichtig. Auch sollten **Fachbegriffe** erklärt oder mit Hilfetext versehen sein, damit z.B. ein Architekt die Audio-Fachterminologie versteht.
- **Datensicherheit:** Da Raumdaten u.U. sensibel sind (Pläne von Gebäuden) und Audioaufnahmen gemacht werden, ist Datenschutz relevant. Im Idealfall funktioniert alles **offline** auf dem Gerät<sup>38</sup> – das Konzept sieht ausdrücklich einen Offline-Modus vor, was positiv hervorzuheben ist. Cloud-Optionen (etwa um schwere Simulationen auszulagern) könnten optional sein, aber die Kernfunktion sollte ohne Internet nutzbar sein, um z.B. in abgeschirmten Gebäuden oder

datensensitiven Umgebungen eingesetzt werden zu können. Wenn Cloud genutzt, dann nur mit Zustimmung und Verschlüsselung.

- **Kompatibilität und Erweiterbarkeit:** Die App wird für **iPadOS** entwickelt, wahrscheinlich mit Swift/SwiftUI. Es sollte darauf geachtet werden, dass sie auf möglichst vielen iPad-Pro-Modellen mit LiDAR läuft (also ab iPad Pro 2020 und neuer). Eine **Portierung** auf iPhone Pro (LiDAR-fähig) wäre denkbar, aber aufgrund der Nutzungsszenarien (Profi-Messung) ist iPad mit großem Screen sinnvoller. Perspektivisch könnte man überlegen, einen Teil der Logik plattformunabhängig zu halten (z.B. Kernalgorithmen in C++), um evtl. eine Desktop-App oder Android-Version (für Geräte mit Time-of-Flight-Sensor) zu ermöglichen. Auch ist die Wahl eines **Open-Source-Stacks** im Konzept verankert <sup>38</sup> – also Libraries mit passenden Lizenzen (MIT/BSD) zu nutzen, damit rechtlich alles sauber ist.
- **Zuverlässigkeit und Fehlerbehandlung:** In einer professionellen Anwendung ist es wichtig, klare Fehlermeldungen zu haben (z.B. *"Raum-Scan unvollständig, bitte alle Wände abgehen"* oder *"Messgeräusch zu leise, Wiederholung nötig"*). Der Prototyp sollte darauf getestet werden, wie robust die Pipeline funktioniert. Abstürze oder Inkonsistenzen wären Show-Stopper in Präsentationen beim Kunden.

## Technologischer Stand und Code-Bewertung

Der Projekt-Ordner enthält vermutlich bereits erste **Swift-Code-Dateien** der Prototyp-App. Mangels direkter Einsicht in den Code können wir nur allgemein bewerten, welche Architektur sinnvoll ist:

- Vermutlich gibt es einen `ViewController` oder in SwiftUI eine `ContentView` für den Scan-Vorgang, der ARKit initialisiert. Hier sollte geprüft werden, ob statt einer Eigenimplementierung auf **ARKit's RoomPlan** gesetzt wird, was die Code-Komplexität enorm reduziert (Apple liefert mit `ARSession` und `RoomBuilder` quasi alles Nötige). Falls der Prototyp vor RoomPlan (WWDC22) begonnen wurde, könnte er selbst Punktwolken triangulieren – das wäre fehleranfällig und kann nun ersetzt werden.
- Die Messfunktion wird auf AVFoundation/AudioToolbox basieren. Hier ist kritisch, dass **AudioSession** korrekt konfiguriert ist (auf `AVAudioSessionCategoryPlayAndRecord` mit Measurement mode, falls ein Sweep abgespielt und aufgenommen wird). Code-seitig sollte die Berechnung der RT30 ordentlich kommentiert und nachvollziehbar sein, da dies das Kernstück ist. Eventuell existiert ein Code für die **Schroeder-Integration** der Impulsantwort; dieser sollte mathematisch validiert werden.
- Struktur: Für eine saubere Architektur wäre zu empfehlen, die App in Schichten zu teilen (z.B. Model-View-ViewModel), aber oft entwickeln Ingenieure prototypisch erst mal alles in einer Klasse. Hier sollte man langfristig refaktorisieren: z.B. eine `RoomModel`-Klasse/Struct, die alle raumakustischen Parameter hält, eine `AcousticsCalculator`-Klasse für alle Formeln, und separate Views für Scan vs. Ergebnis.
- **Code-Qualität:** Ohne konkreten Code ist schwer zu sagen, aber da **Git Repos, VSCode und GitHub Copilot** genutzt werden, wurde vermutlich viel KI-generierter Code eingesetzt. Das kann zu gelegentlich unsauberem Swift führen (falsche Optionals-Behandlung, etc.). Ein kritischer Blick eines erfahrenen Swift-Entwicklers ist ratsam, besonders hinsichtlich **Memory Leaks** (ARKit kann viele Daten pushen – sicherstellen, dass alte Scans freigegeben werden), Threading (Audio

sollte auf separatem Thread verarbeitet werden um UI nicht zu blockieren) und **Apple Guidelines** (z.B. Nutzungsanfragen für Kamera/Mikrofon sind Pflicht, sonst Absturz).

- **Externe Libraries:** Es ist unklar, ob bereits Open-Source-Bibliotheken integriert sind. Mögliche Kandidaten: *Accelerate* (Apples DSP-Framework) für FFT; vielleicht *ARKit* (ja) und *SceneKit* or *RealityKit* für 3D Darstellung; *Charts* library für Graphen? Wenn Copilot genutzt wurde, könnten ähnliche Projekte inspiriert haben. Eine Code-Review sollte sicherstellen, dass Lizenzrechte gewahrt bleiben (Copilot könnte Code-Snippets eingebracht haben; man sollte Lizenzen prüfen wenn größerer externer Codeabschnitt identifizierbar ist).

Insgesamt scheint der Prototyp die Machbarkeit bereits bestätigt zu haben – wesentliche Bausteine (Scan, Berechnung, Messung) sind heutzutage gut umsetzbar. Der nächste Schritt ist, aus dem Prototyp einen robusten **Produkt-Prototyp** zu formen, der intern getestet werden kann.

## Visionäre Erweiterungsmöglichkeiten

Neben den Kernfunktionen bietet das Konzept auch **visionäre Ansätze**, die das Produkt in Zukunft ausbauen könnten:

- **KI-gestützte Optimierung:** Durch Machine Learning könnte die App mit der Zeit lernen, bessere Prognosen zu stellen. Beispielsweise könnten die Differenzen zwischen berechneter und gemessener Nachhallzeit durch einen ML-Algorithmus ausgewertet werden, um das Rechenmodell kontinuierlich anzupassen <sup>24</sup>. Denkbar ist ein systematisches Anlernen: Die App sammelt (mit Einwilligung) anonymisierte Projekte und bildet daraus ein verbessertes Modell zur RT60-Vorhersage, das z.B. komplexe Geometrien berücksichtigt (über einen **neuronalen Netzwerk-Ansatz** neben den klassischen Formeln).
- **Community-Integration & Open Source Hub:** Wenn der Kern Open Source ist, kann man eine Entwickler- und Experten-Community anlocken. Ein Portal (ähnlich *CHORAS* auf ResearchGate, das als Open-Source-Hub erwähnt wird <sup>39</sup> <sup>40</sup>) könnte entstehen, wo Akustiker selbst Materialdaten beitragen oder Plugins entwickeln. Visionär wäre eine **Plattform**, auf der Anwender eigene Materialien einpflegen und teilen, oder Messdaten in eine gemeinsame Datenbank laden, um die Grundlage für alle zu verbessern. Diese offene Strategie könnte das Projekt beschleunigen und es als *Standard-Tool* in der Branche etablieren, während das Unternehmen weiterhin über Premium-Features oder Services Einnahmen erzielt.
- **AR/VR und Immersive Planung:** Die Integration von **Virtual Reality / Augmented Reality** ist ein spannender Ausblick <sup>41</sup>. Man könnte z.B. mit einer AR-Brille (oder dem kommenden Apple Vision Pro) den Raum virtuell mit akustischen Elementen ausstatten und *visuell & akustisch* erleben, bevor man real umbaut. Die App könnte AR-Marker an Wänden anzeigen, wo Absorber platziert werden sollten, oder per VR-Simulation Architekten ermöglichen, den Klang verschiedener Designs zu vergleichen. Dies wäre bahnbrechend in der **Designkommunikation** – Architekten könnten ihren Kunden hörbar machen, warum Akustikmaßnahmen wichtig sind.
- **Erweiterung auf andere Akustikbereiche:** Bisher liegt der Fokus auf **Raumakustik** (Innenräume, Nachhall). Zukünftig könnte das Konzept auch auf **Gebäudeschall** (Schallschutz, z.B. Messung von Nachhall in Treppenhäusern, oder Schalldämmung) erweitert werden. Oder **Außenlärm-Scanning** – z.B. Lärmkartierung eines Außenbereichs mittels Drohne + Akustikkamera. Zwar ist das ein anderes Feld, aber die Idee einer schnellen Bestandsaufnahme + KI-Auswertung lässt sich übertragen.

- **Schnittstellen zu BIM und Planungstools:** In der Vision würde die App integraler Bestandteil digitaler Workflows. Eine **API-Anbindung** an gängige BIM-Software (Autodesk Revit, Archicad) könnte entwickelt werden, sodass Architekten direkt aus ihren 3D-Modellen eine Akustiksimulation anstoßen können. Ebenso könnte die App als **Plugin** für CAD- oder CAFM-Systeme fungieren. Die Folien erwähnen z.B. *Autodesk Forma* (ehemals Spacemaker) <sup>16</sup> – dort fließen KI-Simulationen in frühe Planungsphasen ein (z.B. Wind, Licht). Akustik könnte hier ergänzt werden.
- **Globale Ausrichtung und Forschung:** Um führend zu bleiben, sollte man sich an neuester **Forschung** orientieren. Offene Projekte wie *Pyroomacoustics*, *PFFDTD*, *I-Simpa* und *AcousticCamera* werden genannt – diese decken unterschiedlichste Simulationsmethoden ab (von statistisch bis wellenbasiert) <sup>42</sup>. Eine regelmäßige Überprüfung, welche Methoden praxisreif werden (z.B. wenn Rechenleistung steigt, könnte man irgendwann eine vereinfachte Wellensimulation auf dem Tablet laufen lassen), gehört zur Vision. Zusammenarbeit mit Hochschulen oder Instituten (Fraunhofer etc., was im Vertriebshandbuch sogar als Ziel erwähnt wird <sup>43</sup>) könnte dabei Win-Win schaffen: Die App als **Anwendungsplattform** für neue Akustik-Algorithmen, und die Forschung liefert die Modelle.

## Kritische Betrachtung und Herausforderungen

Bei aller Begeisterung sollen auch die **kritischen Aspekte** nicht übersehen werden:

- **Technische Machbarkeit komplexer Features:** Einige visionäre Punkte (z.B. Echtzeit-Auralisation auf dem iPad) sind anspruchsvoll. DSP in Echtzeit erfordert Optimierung, und AR-Visualisierung von Absorbern ist nicht trivial (Objekterkennung, Verankerung an Wänden). Man muss aufpassen, sich nicht zu verzetteln – ein MVP sollte sich auf Kernfunktionen konzentrieren. Die Gefahr besteht, zu viele Features auf einmal umzusetzen und dabei in keinem richtig auszureifen. Hier ist Priorisierung wichtig.
- **Genauigkeit vs. Einfachheit:** Die Sabine-basierte Berechnung ist simpel, vernachlässigt aber Faktoren wie Raumform oder Frequenzabhängigkeiten im Detail. Profis könnten die Aussagekraft infrage stellen, wenn die App-Ergebnisse stark von realen Messungen abweichen. Es gilt also, die **Limitationen transparent zu machen**. Eventuell muss man Hinweise geben wie "*bei unregelmäßigen Raumformen kann es zu Abweichungen kommen*". Der Erfolg des Produkts hängt davon ab, dass es **hinreichend genau** ist für praktische Zwecke. Hier wäre es kritisch, Beta-Tests in verschiedenen Räumen zu machen (Büro, Klassenzimmer, Halle etc.) und die Abweichungen zu messen. Gegebenenfalls muss das Rechenmodell verfeinert werden (z.B. Owens-Corning-Formeln oder Integration empirischer Korrekturfaktoren).
- **User Acceptance und Change Management:** Architekten und Bauingenieure sind nicht alle sofort von KI und neuen Tools zu überzeugen. Einige könnten skeptisch sein, ob eine App ihren Akustiker ersetzen kann. Daher muss das Produkt als **Hilfsmittel** positioniert werden, nicht als Konkurrenz zu etablierten Planern. Der Beratungsanteil der MSH-Gruppe bleibt wichtig – die App soll diese Arbeit **ergänzen, nicht ersetzen**. Kritisch ist auch, dass Nutzer bereit sind, ein iPad als Messwerkzeug zu verwenden – in konservativen Branchen braucht es oft Referenzen und Normnachweise, damit ein digitales Tool akzeptiert wird. Ein strategischer Schritt könnte sein, die App in **Normen oder Richtlinien** zu verankern oder zumindest deren Anforderungen zu erfüllen, um Glaubwürdigkeit zu haben.



- **Wettbewerbsrisiko:** Sollte das Konzept erfolgreich erste Kunden gewinnen, werden mit Sicherheit größere Softwarehäuser aufmerksam. Zum Beispiel könnte Autodesk auf die Idee kommen, Akustik in ihr *Forma* zu integrieren, oder bestehende Akustiksoftware-Anbieter wie **ODEON** oder **EASE** könnten mobile Scanner entwickeln. Auch Apple selbst erweitert ARKit ständig – wenn Apple eines Tages akustische Eigenschaften mit scannt (Vision Pro hat z.B. Raumsound-Funktionen), könnte Konkurrenz entstehen. Es ist also wichtig, **schnell einen Vorsprung** aufzubauen und ggf. Partnerschaften einzugehen, bevor Wettbewerber aufholen. Der offene Ansatz (Open-Source-Kern) kann zwar Community-Power bringen, birgt aber das Risiko, dass andere den Code aufgreifen. Hier muss man sich über die **Schutzrechte** Gedanken machen – evtl. gewisse Teile proprietär halten oder patentieren lassen (z.B. eine spezielle Kombination der Verfahren).
- **Investitionsbedarf:** Die Entwicklung einer solch komplexen App erfordert sowohl **Zeit als auch Geld**. Momentan wird intern mit vorhandenem Personal (Toningenieure, Key Accounter mit Copilot) entwickelt. Das ist ein guter Start, aber mittelfristig könnte professionelle Entwicklungsunterstützung nötig sein, besonders für fortgeschrittene iOS-Themen (ARKit, Metal Optimierungen, etc.). Es stellt sich also die Frage der Finanzierung: Soll das Projekt *Bootstrapped* bleiben, oder zieht man Förderung/Investoren hinzu? Diese Entscheidung beeinflusst Tempo und geistige Eigentumsrechte. Kritisch zu beachten: Die **Softwarewartung** – iOS Updates, neue Gerätetypen (Vision Pro), etc., all das erfordert dauerhafte Kapazitäten. Ein unterschätzter Aufwand in vielen Tech-Projekten.

Trotz dieser Herausforderungen überwiegt das Potenzial des Projekts. Durch vorausschauende Planung – sowohl geschäftlich (Fokus auf Engpass, Kunden einbeziehen) als auch technisch (saubere Architektur, Tests, iteratives Vorgehen) – können die Risiken gemindert werden.

## Fazit und Empfehlungen

Abschließend lässt sich sagen, dass die Kombination aus **unternehmerischer Vision** und **technischer Innovation** in diesem Projekt äußerst vielversprechend ist. Die Analyse der bereitgestellten Unterlagen zeigt ein durchdachtes Konzept, das sowohl einen echten Bedarf im Markt adressiert als auch die vorhandene Expertise der MSH-Audio-Gruppe nutzt. Die **Prototyp-Phase** hat vermutlich bereits gezeigt, dass Kernkomponenten funktionstüchtig sind – jetzt gilt es, darauf aufzubauen.

### Empfehlungen im Überblick:

- **Kernfeatures priorisieren:** Konzentrieren Sie sich kurzfristig darauf, den **Raumscan, die RT60-Messung und den Bericht** einwandfrei zum Laufen zu bringen. Diese drei Funktionen sind der Nutzwert, für den Kunden zahlen. Erweiterungen wie KI-Optimierung und AR-Auralisation können nachgelagert in Version 2.0 kommen, wenn das Grundprodukt stabil ist.
- **Lastenheft schärfen:** Nutzen Sie die oben formulierten Anforderungen als Grundlage für ein ausführliches Lastenheft. Ergänzen Sie konkrete Akzeptanzkriterien (z.B. *"Die Abweichung zwischen berechneter und gemessener RT60 soll in üblichen Räumen <15% liegen"*) und definieren Sie User Stories (etwa *"Als Akustiker möchte ich einen PDF-Bericht exportieren, um ihn meinem Kunden zu geben"*). Ein klar definiertes Lastenheft hilft, mit eventuellen externen Entwicklern auf derselben Linie zu bleiben.
- **Codequalität sicherstellen:** Ziehen Sie einen erfahrenen **Swift-Entwickler** oder Code-Reviewer hinzu, der den bisherigen Prototyp-Code prüft. Fokus sollte auf Stabilität, korrekter Verwendung

der Apple-APIs und Lesbarkeit liegen. Dies erhöht die Erfolgsaussicht, wenn das Projekt skaliert oder weitere Entwickler einsteigen.

- **Partnerschaften suchen:** Überlegen Sie, ob es strategische Partner gibt – z.B. **Hochschulen** (für Algorithmik), **Messgeräte-Hersteller** (für eventuell Hardware-Bundles mit einem geeigneten Messmikrofon fürs iPad) oder **Softwarefirmen** (für Integration in größere Plattformen). Eine Partnerschaft mit einem etablierten Player könnte Marktzugang erleichtern und Vertrauen schaffen.
- **Beta-Test und Feedback:** Bevor groß in den Markt, sollte die App mit **Pilotkunden** getestet werden. Vielleicht gibt es befreundete Integratoren oder Architekten (die im Vertriebshandbuch genannten Kontakte) – bieten Sie ihnen an, einen Pilot zu fahren. Deren Feedback ist Gold wert und hilft, das Produkt zu optimieren und gleichzeitig schon Testimonials zu sammeln.

In Summe hat das Projekt das Potenzial, **pionierhaft** im Bereich der Bau- und Raumakustik zu wirken – eine Domäne, die bisher vom digitalen Wandel nur wenig erfasst war. Durch die sinnvolle Verknüpfung von **moderner Sensorik (LiDAR)**, **Akustik-Know-how** und **KI-Methoden** kann ein Werkzeug entstehen, das Räume nicht nur vermisst, sondern ihnen akustische *"Intelligenz"* verleiht. Mit kritischem Blick auf die Umsetzung und gleichzeitig visionärer Weitsicht, wie sie in den vorliegenden Dokumenten deutlich wird, stehen die Chancen gut, dass dieses Vorhaben sowohl technisch gelingt als auch am Markt erfolgreich Fuß fasst.

**Quellen:** Die Analyse basiert auf den bereitgestellten Konzeptdokumenten und weiteren Recherchen. Wichtige Referenzen umfassen unter anderem DIN 18041 für raumakustische Anforderungen, Marktstudien zur Renovierungsquote im Bauwesen, sowie technische Dokumentationen (Apple RoomPlan für LiDAR-Scanning <sup>25</sup>, Aurivus KI-Scan zu BIM <sup>14</sup>) und bestehende Lösungen wie CadnaR.scan <sup>26</sup>. Diese wurden im Text an den entsprechenden Stellen zitiert, um die Nachvollziehbarkeit zu gewährleisten. Letztlich bildet diese Mischung aus internen Erkenntnissen und externen Fakten die Grundlage für die vorstehenden Empfehlungen und Bewertungen.

---

1 2 5 6 7 8 9 10 11 13 15 16 19 20 21 22 23 24 29 30 31 33 34 35 38 39 40 41 42

answer (1).pptx

file:///file-HZbWAm8qB1EdRUTk36VV1L

3 4 43 Vertriebshandbuch\_Akustik\_MSH\_ILLUSTRIERT.docx

file:///file-FnwxyJcXB21gTpgRwtymr5

12 Builders and Architects Ride a Renovation Wave in US - Bloomberg

<https://www.bloomberg.com/news/features/2022-10-19/builders-and-architects-ride-a-renovation-wave-in-us>

14 aurivus – From scanning to digital

<https://aurivus.com/>

17 18 26 27 36 37 CadnaRscan - Datakustik GmbH

<https://www.datakustik.com/products/cadnarscan>

25 28 RoomPlan Overview - Augmented Reality - Apple Developer

<https://developer.apple.com/augmented-reality/roomplan/>

32 Messung der Nachhallzeit DIN3382 - Dr-Jordan-Design

<https://www.akulap.de/joomla/index.php/de/support?catid=82&id=79&view=article>