



## Reflektionsrapport

---

TDP017 - Företagsprojekt

Dennis Abrikossov - Denab905

January 8, 2026

# 1 Beskrivning av projektet och slutprodukten

Projektet utfördes i samarbete med företaget Agricam och syftade till att utveckla en funktion för röststyrda uppgiftshantering. Målet var att implementera en funktion där användaren kan tala in instruktioner via en mikrofon, varpå systemet transkriberar talet och automatiskt skapar en relevant uppgift på en digital whiteboard.

Arbetet delades upp i tre huvudsakliga faser. Den inledande fasen, **Del 0**, utgjorde en förstudie där vi undersökte olika modeller för Speech-to-Text (STT) för att utvärdera vilken lösning som passade bäst för projektets behov, särskilt med avseende på svenska språket. I **Del 1** utvecklades därefter en fristående backend-tjänst som kunde ta emot ljudklipp, transkribera dessa till text och tolka innehållet för att skapa en uppgift i databasen. Slutligen, under **Del 2**, integrerades tjänsten med Agricams befintliga frontend-kod. Under denna fas lades arkitekturen om till en WebSocket-lösning för att möjliggöra realtidsfeedback medan användaren talar, och vi implementerade en LLM (Large Language Model) för att omvandla den transkriberade texten till strukturerade kommandon.

## 1.1 Slutprodukten

Den slutliga produkten som levererades till Agricam är en integrerad modul i deras applikation. När en användare talar in i mikrofonen visas texten i realtid på skärmen. Systemet känner automatiskt av när användaren slutar prata (alternativt kan användaren trycka på en knapp).

Efter avslutat tal läses den tolkade texten upp tillbaka till användaren via Text-to-Speech (TTS) för verifiering. Användaren kan då bekräfta om tolkningen är korrekt, varpå en uppgift med relevant titel skapas. Systemet hanterar även tidsaspekter; om användaren nämner en specifik dag tolkas detta för att placera uppgiften på rätt datum. Om användaren väljer att inte bekräfta, avbryts processen med ljudfeedbacken "AVBRÖT". Utöver integrationen levererades även en separat testklient för att kunna verifiera funktionaliteten utanför huvudapplikationen. En dag alternativt kunde användare välja att inte lägga till den uppgiften och då kommer TTS tjänsten att säga "AVBRÖT". Det fanns också en testklient som man kan använda för att testa alla dessa funktioner utanför deras existerande app.

# 2 Beskrivning av valda och använda utvecklingsmiljöer och stödverktyg

För att genomföra projektet användes en kombination av olika miljöer och verktyg. För kodskrivning och debugging användes främst Visual Studio och Visual Studio Code, medan Android Studio nyttjades för att emulera Android-enheter vid testning av mobilapplikationen.

Gällande programspråk och ramverk utvecklades backend primärt i **C#** (.NET och SignalR), medan frontend-integrationen skedde i **React Native**. Dessa tekniker valdes primärt för att underlätta integrationen i företagets befintliga system efter projektets avslut. Som molntjänst användes **Azure AI Foundry** för att hantera de AI-modeller som krävdes för transkribering och tolkning.

### **3 Beskrivning av min roll och mitt ansvarsområde under utvecklingen av produkten**

Gruppen valde att arbeta enligt en agil metodik utan fasta roller tilldelade från start. Istället inleddes vi varje ny del av projektet med ett gemensamt planeringsmöte där vi diskuterade nödvändiga arbetsuppgifter och enades om vad som behövde göras. Därefter fördelades uppgifterna baserat på gruppmedlemmarnas intresse och kompetens. Arbetet utfördes på plats i Agricams lokaler, vilket underlättade kommunikationen.

Även om vi inte hade formella titlar, tog jag under projektets andra hälft ett utökat ansvar för planering och struktur genom att driva veckomöten och samordna schemaläggningen. Jag kartlade gruppens tillgänglighet och identifierade alla gemensamma luckor i våra scheman. Utifrån dessa alternativ beslutade vi sedan gemensamt i gruppen vilka tider vi skulle nyttja för arbete. Detta arbetssätt säkerställde att vi effektivt kunde boka in arbetspass och hålla tempot uppe mot slutet av projektet. Jag tog även ansvar för att sammanställa och skicka in en av veckorapporterna.

### **4 Beskrivning av vilka delar av produkten som jag jobbat med**

Jag arbetade primärt genom parprogrammering tillsammans med en annan gruppmedlem (Elvin Holst). Mina bidrag fördelade sig över projektets delar enligt följande:

#### **4.1 Del 0: Förstudie och Testning**

Under förstudien fokuserade jag på att samla in och utvärdera träningsdata. Jag letade fram ljudklipp med olika svenska dialekter för att testa robustheten i de modeller vi utvärderade. Jag genomförde även tester av lokala STT-modeller för att se hur de presterade jämfört med molnbaserade alternativ.

#### **4.2 Del 1: Backend-struktur**

I den första utvecklingsfasen låg mitt fokus på backend-logiken. Jag arbetade med att bygga den första controllern i C# och implementerade valideringslogik för att säkerställa att inkommende filer till tjänsten var av korrekt format och typ innan bearbetning.

#### **4.3 Del 2: WebSocket och Integration**

Den största delen av mitt arbete under Del 2, ansvarade jag för att bygga om kommunikationen från traditionella HTTP-anrop till **WebSockets** för att stödja realtidstranskribering. Konkret innebar detta att jag byggde WebSocket-endpointen med dess tillhörande funktioner och events, samt implementerade logiken för live-transkribering mot STT-tjänsten i Azure AI Foundry.

För att kunna testa datafödet och kommunikationen isolerat utvecklade jag en separat testklient för WebSockets. Slutligen arbetade jag med att koppla ihop frontend-applikationen (React Native) med vår backend-tjänst via den nya WebSocket-lösningen.

## 5 Diskussion

Ett område där jag ser tydlig förbättringspotential rör vår tidsplanering och prioritering under projektets inledande fas (Del 0). Vi spenderade en oproportionerligt stor del av tiden på att brett testa och utvärdera diverse olika STT-lösningar.

I slutändan landade vi dock i att använda Azure AI Foundry, som är en samling av LLM- och STT-tjänster. Det visade sig att de modeller vi hade lagt ner omfattande tid på att testa inte fanns tillgängliga eller var relevanta i den Azure-miljö vi skulle använda.

Om jag fick göra om projektet skulle jag förändra metodiken kring förstudien. Jag skulle i ett mycket tidigare skede föra en dialog med uppdragsgivaren för att fastställa vilka tekniska plattformar och molntjänster som är acceptabla eller föredragna. Hade vi tidigt vetat att vi skulle begränsa oss till Azures ekosystem hade vi kunnat rikta vår förstudie specifikt mot deras tjänsteutbud. Detta hade effektiviserat arbetet avsevärt och frigjort tid som istället kunde ha lagts på att ytterligare förfina funktionaliteten i Del 1 och Del 2.