Process Specification

Blindsight / Theia
Version 1.0

Prepared by Charlie Wong, Alec Yeasting, Riley Hunter, Zach Gherman
Team CARZ - WSU CPT_S 484
12/12/2021

1. Team

Project Manager: Alec Yeasting

Responsibilities:

- Creation of a dummy navigation system for prototyping using trigonometry to convert a sequence of points into human usable navigation instructions.
- KAOS model brainstorming
- General document collaboration
- Vision Document collaboration

Developer: Charlie Wong

Responsibilities:

- User Manual Lead
- Menu UI Mockup collaboration
- General document collaboration
- Vision Document collaboration

Developer: Riley Hunter

Responsibilities:

- Menu UI Mockup Lead
- Vision Document Lead
- User Manual collaboration
- General document collaboration

Developer: Zach Gherman

Responsibilities:

- KAOS model brainstorming/creation
- IDEF0 model creation
- Process specification document finalization
- General document collaboration

2. Iterations

Initial
Initially, the team met to discuss general ideas for how to tackle the issues of navigating a blind person in-doors. We culminated with the ideas of using Bluetooth, WiFi, LiDAR, Magnetometer, Accelerometer, Camera (with DenseNets) to determine the Users position from specific objects or objectives for navigating the blind user. To better represent the blind community, Zach reached out to Rodney Moag, a Professor Emeritus at the University of Texas who is blind and became our subject matter expert (SME). The biggest take-away from Rodney Moag is that he has difficulties noticing objects at hip to head level which his cane cannot detect.

Generate initial prototype and documents
From the culmination of our ideas of how to navigate, we each worked on producing a prototype of each base component: GUI, sound, and sensors. Riley worked on creating a basic GUI for the Register and Log-In screens of the application while Zach worked on figuring out how to play sounds for the User (beeps/blips, sound files) and Alec worked on identifying which sensors the target platform (mobile device / phone) are available. Charlie worked on generating a user manual for how to navigate the Register and Log-In screen as well as a general idea of how to use the application. After we had a basic understanding of our components, we worked on updating the WRS document by filling out the Functional/ Non-Functional Requirements of the Application to help further hone our focus.

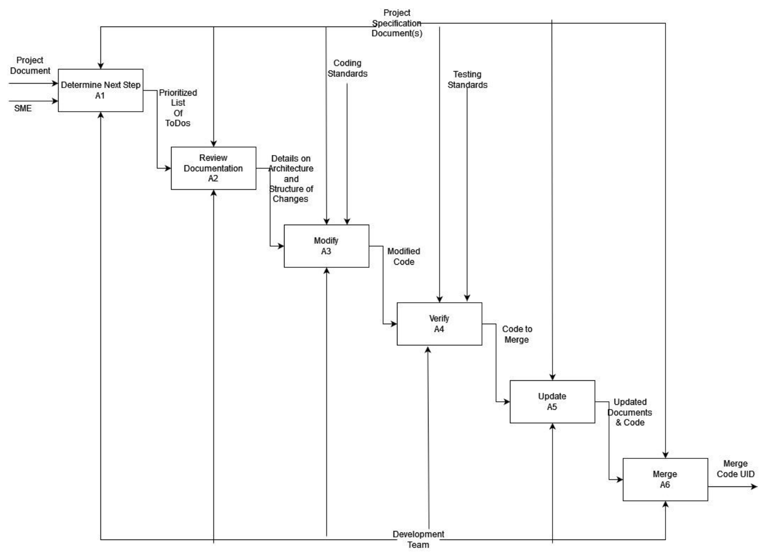
Modify existing model, documents and prototype
After Phase 1 was completed, the team needed to move quickly to generate documents and models for the additional requirements set forth by the professor (Bolong Zeng). Zach got to work by developing the IDEF0 diagrams (shown in Section 3 of this document), making modifications to the WRS Document, filling out our process specification document, and brainstorming KAOS models with Alec. Riley and Charlie went to work on the Vision document and further update the WRS document. Alec focused his attention on developing the text-to-speech code for our project. He also went to work on the WRS document, the Vision document, initializing the process specification document and KAOS modeling with Zach.

Further Prototyping
This is mainly about Alec and his contributions. Alec took the initiative to develop the navigation process for the project. He built a virtual user to follow the prompts set-forth by the hard-coded navigation system. Using some basic trigonometry and knowledge of the accelerometer, he was able to prototype a system which has a virtual user follow a specified path. Then he merged the Text-to-Speech portion of the code with the direction outputs to the screen for the users to follow so that blind users have a way of getting information about the next step in being guided.

The process can be easily digested from the IDEF0 diagrams, and descriptions about how the work is broken down, provided in section 3 below.

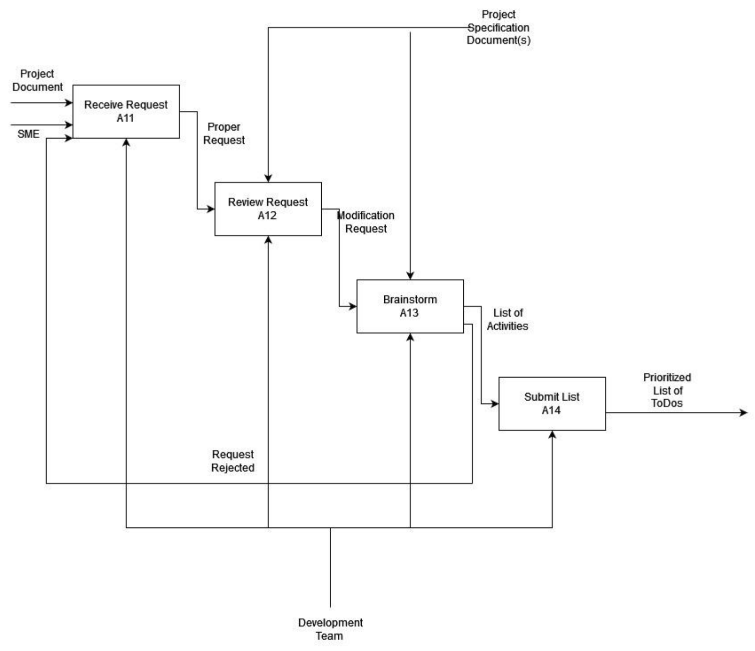
3. IDEF0 Models

Here we will show the IDEF0 Models which will help to further understand our process.



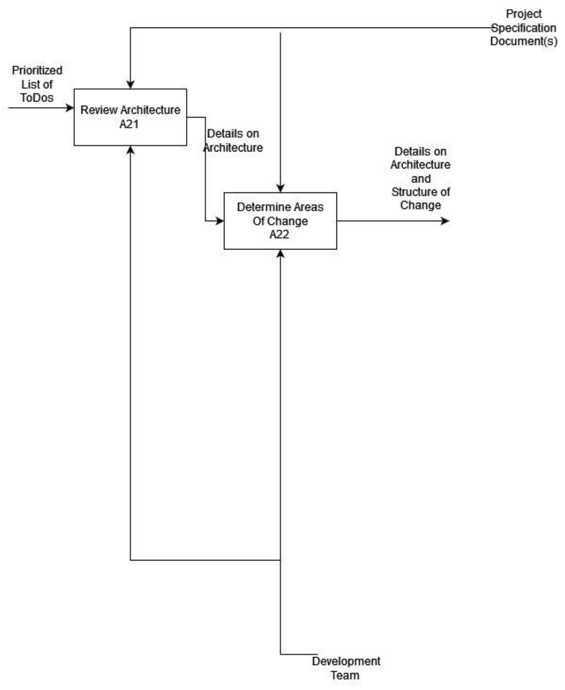
A0

The IDEF0 Diagram A0 shows our overall process. We will break this down with our other diagrams, explaining what we are doing at each phase.



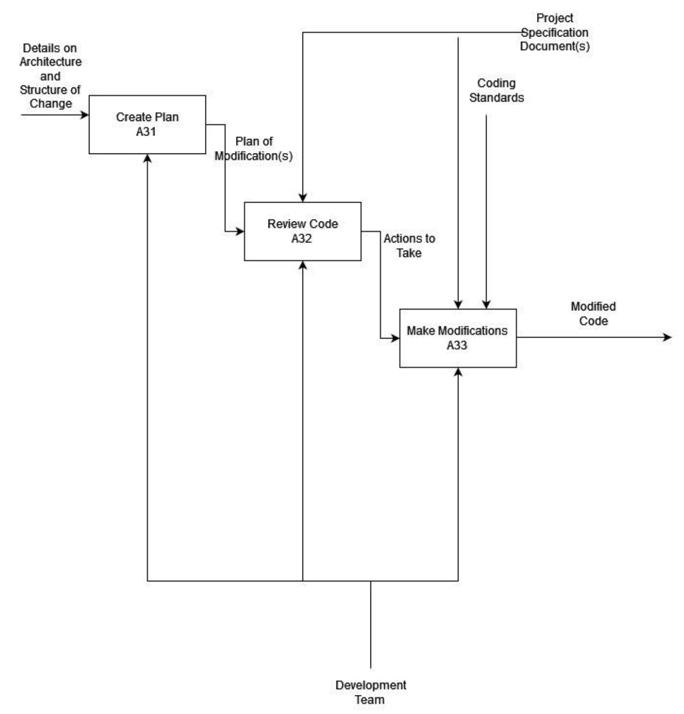
A1

The IDEF0 Diagram A1 shows the broken down process of how we determine what to do for the next step in our prototyping/ process. It starts by our SME (Rodney Moag) relaying a message to Zach about a neat idea. Zach then takes the request from the SME and relays it to the team in the form of a request (more of a statement of what our SME would like with a question and description of its feasibility to the team). Whoever has the available time then looks at the request and reviews the documentation to ensure that the request falls within the scope of our project. The next step involves brainstorming possible solutions to the request. If the team can determine a solution, that solution is recorded and split into a list of activities. If the team can not determine a solution, the request is sent back to our SME explaining that we cannot with our current knowledge complete that request, and the request is rejected. With this list of activities, we simply post it to the group for anyone to do as a prioritized list of ToDos.


Project Specification Document(s)
Prioritized List of ToDos
Review Architecture A21
Details on Architecture
Determine Areas Of Change A22
Details on Architecture and Structure of Change
Development Team

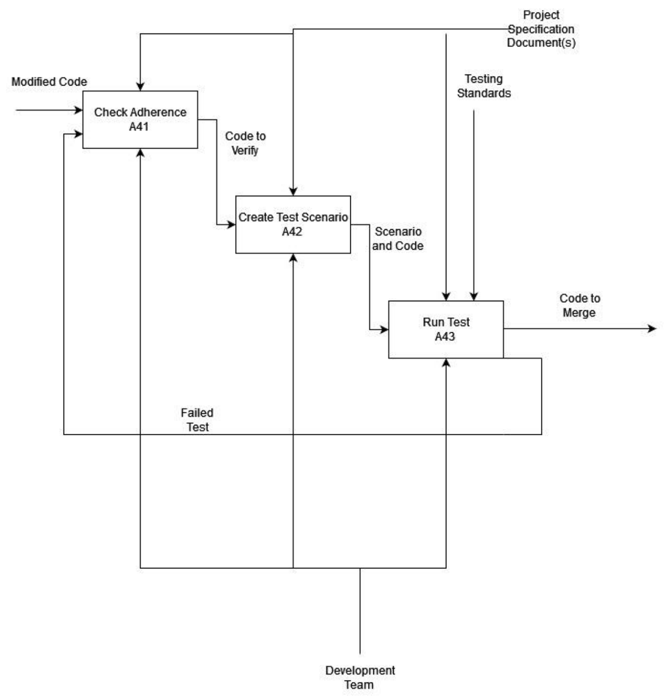
A2

The IDEF0 diagram A2 shows the broken down process of how we review the request (prioritized list of ToDos) in relation to the most up-to-date documents regarding the architecture of the project. We start by taking the prioritized list of ToDos and checking the Architecture to see where/how the request would fit in the project. Next, after we have the details of how it would fit (architecture-wise) we then determine what areas in the code and in the documents will need to be updated because of this request. This gives us all the information we need to quickly and easily modify our documents in later stages (this is equivalent to brainstorming / getting ideas on paper).



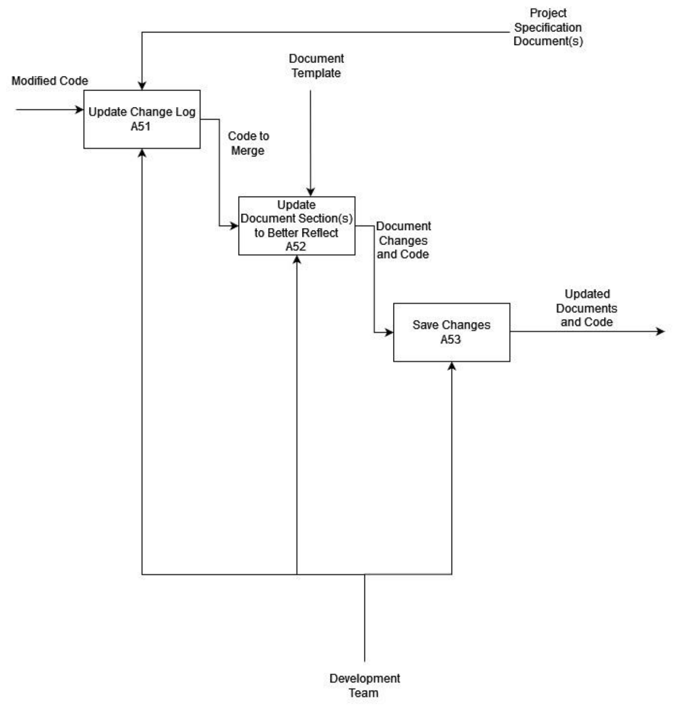
A3

The IDEF0 diagram A3 shows our process of how we go about making modifications to the code. It starts by getting the information about the architecture of the project and how the changes will affect the project as a whole. The developer in charge of making the modification then creates a plan of attack (how they will tackle the problem based on the architecture, structure of change, and the prioritized list of ToDos [not listed]). This leads the developer to have a plan of modifications they will be making. The developer then reviews the code already developed to see where their modifications would fit. During the code review, they insert ToDos into the code which are seen as actions to take. After they have inserted all the appropriate ToDos into the appropriate areas of the coe, they then generate the new code and/or update older code.



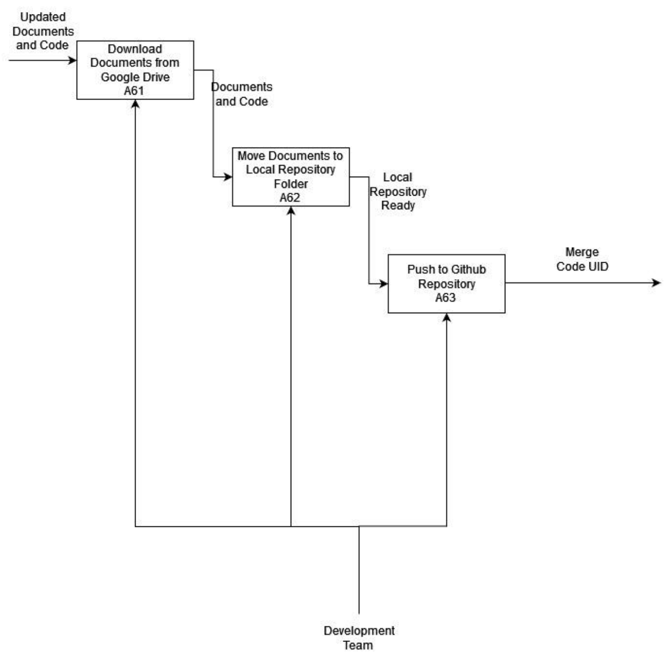
A4

The IDEF0 Diagram A4 shows how the developer who wrote the modified code goes about testing their code to ensure validity. It starts by taking the code which was recently modified and double checking that the modified code exists in the appropriate areas based on the adherence to the plan created. After code has been checked, the developer creates a test scenario of the newly modified code which adheres to the plan. After the tests have been created, the developer then runs the test. If the test fails, the developer must double check adherence to the documentation, otherwise it is passed to be pushed to the Github Repository (A6 Merge).


Project Specification Document(s)
Modified Code
Document Template
Update Change Log A51
Code to Merge
Update Document Section(s) to Better Reflect A52
Document Changes and Code
Updated Documents and Code
Save Changes A53
Development Team

A5

The IDEF0 Diagram A5 shows the process of how the developer who modified the code and ran the test(s) will go about updating the documentation with the modifications. It starts with the developer updating the changelog. Then the developer updates the appropriate sections based on the code they have created. Once the appropriate document(s) has been updated, the changes are saved (this is done automatically by Google Drive).


Updated Documents and Code
Download Documents from Google Drive A61
Documents and Code
Move Documents to Local Repository Folder A62
Local Repository Ready
Push to Github Repository A63
Merge Code UID
Development Team

A6

The IDEF0 diagram A6 shows the process of how we merge our code and documents to the Github Repository. It starts by the development team member who made the modification to the code, downloading the updated documents from Google Drive. Then that development team member moves the updated documents to their local github repository folder. That member then Pushes and merges the updates with the Github Repo. This returns a Merge Code UID which the team member can reference for prosperity.