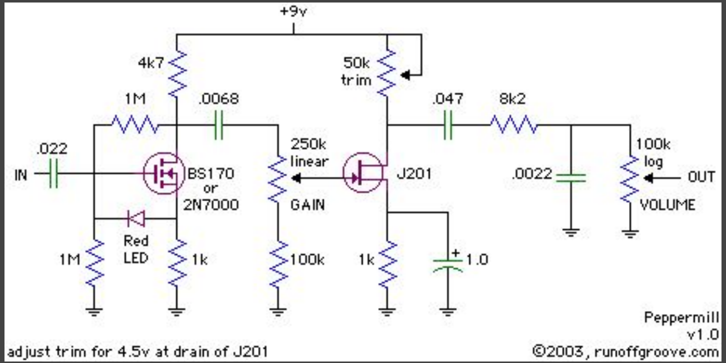
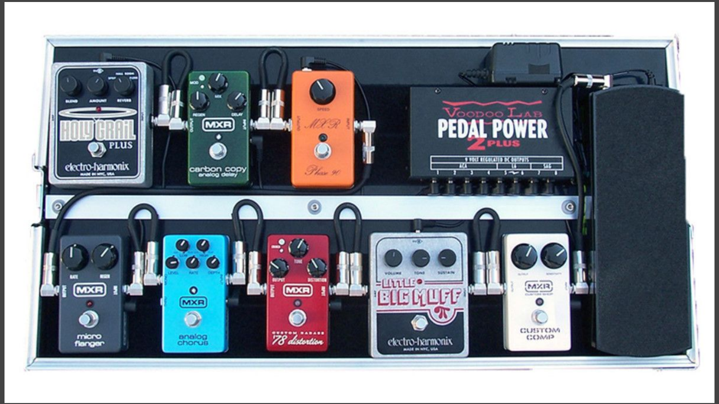
# Stairway to Hammerschlag

Joseph Kim, Stephen He, and Matthew Kasper
(Team D8)

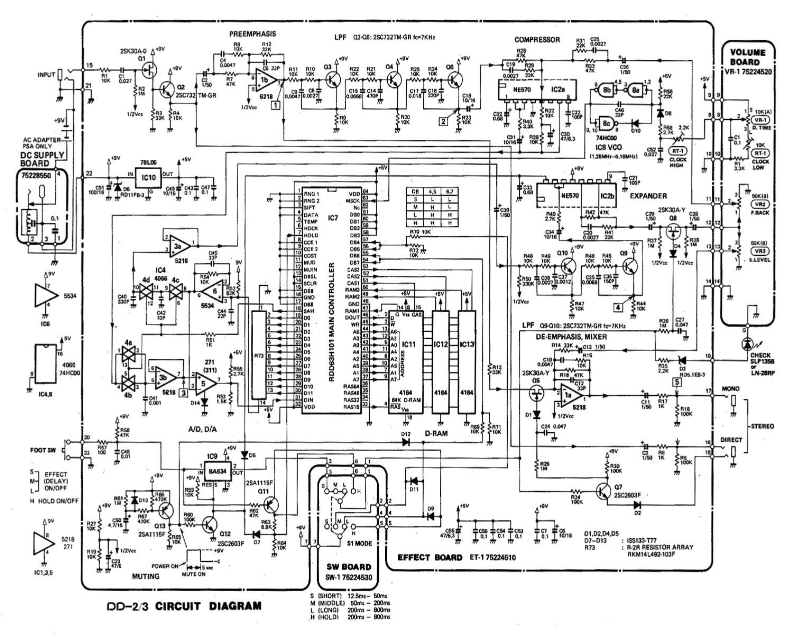
# What is a guitar pedal?



Overdrive Pedal Schematic



Pedal Board



Digital Delay Pedal

# Use Case - Improve Workflow for Guitar Pedal Designers

## **Traditional** Pedal Design Workflow

Design Pedal Circuitry

Construct Pedal Circuitry

Test and Iterate

Choose an underlying pedal circuit based on established pedal designs, or create a circuit completely from scratch.

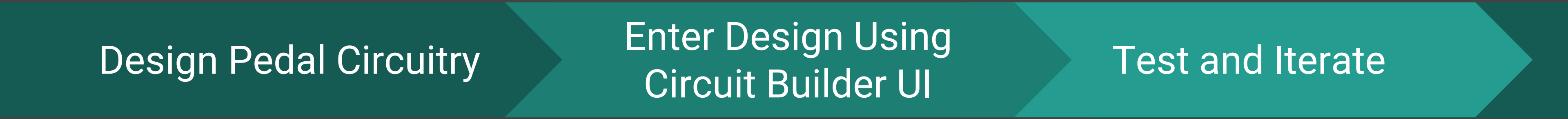
Replicate the design in hardware (PCB or breadboard).

Connect hardware to an instrument to sees how it sounds.

Repeat process until satisfied with sound.

# Use Case - Improve Workflow for Guitar Pedal Designers

## **Stairway to Hammerschlag** Pedal Design Workflow



Design Pedal Circuitry

Choose an underlying pedal circuit based on established pedal designs, or create a circuit completely from scratch.

Enter Design Using Circuit Builder UI

Enter a description of the pedal circuitry into the **Stairway to Hammerschlag** application. This interface will feature drag and drop circuit building capabilities.

Test and Iterate

Simulate the effects of chosen design on real-time or recorded audio.

Repeat process until satisfied with sound.

ECE Areas Covered: Software Systems, Signals, Analog Circuits

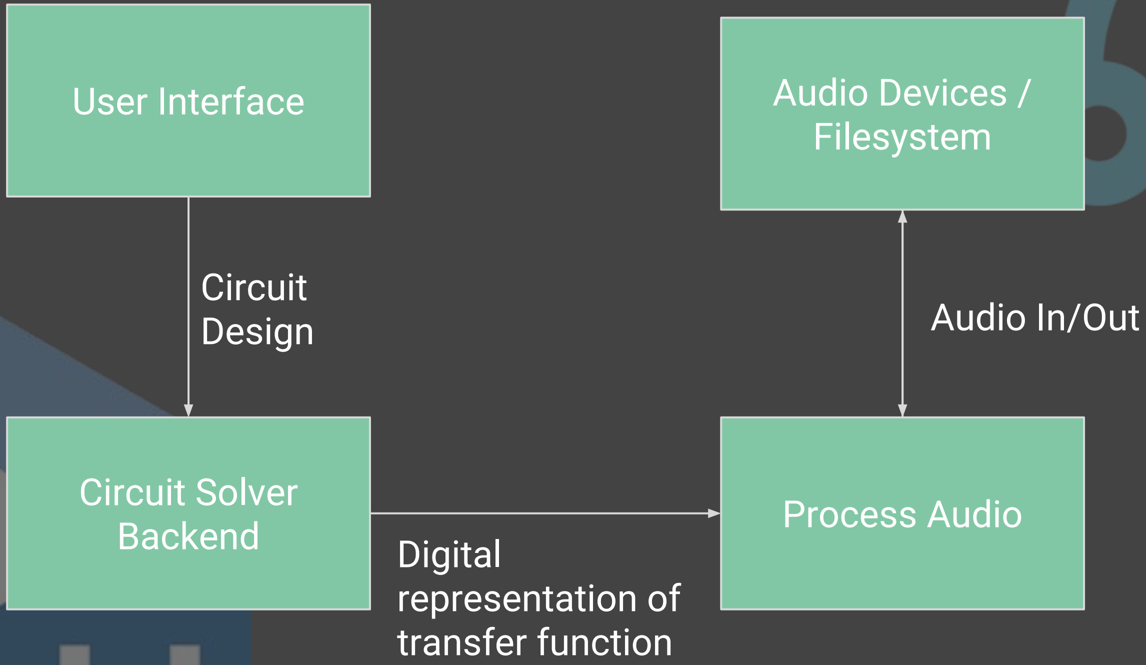
# Requirements

- Simulate the behavior of resistors, capacitors, diodes, transistors, and possibly op-amps in software to derive transfer function of circuits
- Be able to process any arbitrary audio file
- Be able to connect an instrument and do processing in pseudo real time (latency less than 15 ms)
- User friendly graphical interface

# Challenges

- Deriving accurate transfer functions from underlying circuits
- Achieving low latency required for real-time audio processing
- Integration of real-time audio processing and circuit simulation components
- Digital representation of various transfer functions

# Solution Approach



User Interface

Circuit Design

Circuit Solver Backend

Digital representation of transfer function

Process Audio

Audio In/Out

Audio Devices / Filesystem

# Solution Approach - Considerations

| Component or Subsystem | Technology Options |
|---|---|
| User Interface | ElectronJS / Proton Native / NW.js |
| Circuit Simulator | C++ / C / Java / Python |
| Audio Processing | JUCE (C++) / Synthesis ToolKit (C++) / SndObj (C++) |
| Hardware Platform | 2017 Macbook Pro / Raspberry Pi 3 / BeagleBoard-X15 |

# Testing, Verification and Metrics

- Have a suite of test-circuits with manually derived transfer functions
- Have a suite of audio files to test our system
- Use an oscilloscope to measure the phase difference between signal in and signal out
- Do a user-study on our UI

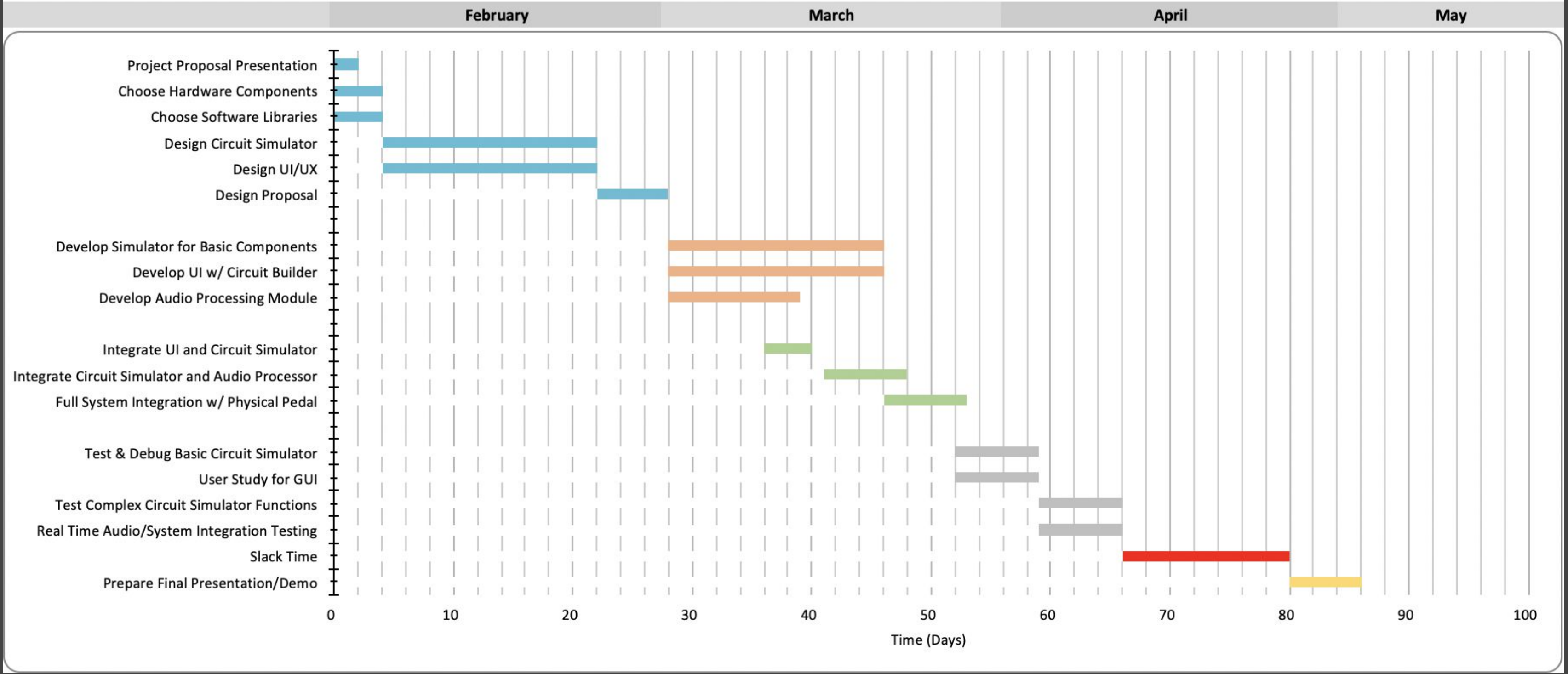
# Normalized Cross Correlation

- Metric for comparing signal similarity
- Agnostic to differences in amplitude
- Ranges from [-1, 1]
- Goal: Achieve score of 0.8 or greater when compared to reference audio for known pedal design

# Tasks and Division of Labor

| | |
|---|---|
| Matt Kasper | ● Research and Design<br>● Circuit Simulator |
| Joseph Kim | ● Research and Design<br>● Audio Processing |
| Stephen He | ● Research and Design<br>● User Interface |

# Schedule


February
March
April
May
Project Proposal Presentation
Choose Hardware Components
Choose Software Libraries
Design Circuit Simulator
Design UI/UX
Design Proposal
Develop Simulator for Basic Components
Develop UI w/ Circuit Builder
Develop Audio Processing Module
Integrate UI and Circuit Simulator
Integrate Circuit Simulator and Audio Processor
Full System Integration w/ Physical Pedal
Test & Debug Basic Circuit Simulator
User Study for GUI
Test Complex Circuit Simulator Functions
Real Time Audio/System Integration Testing
Slack Time
Prepare Final Presentation/Demo
0
10
20
30
40
50
60
70
80
90
100
Time (Days)