

# TeamBandit

## Design Review 1

### Team Outlaws

Quinn Melssen

Dakota Battle

Liam Scholl

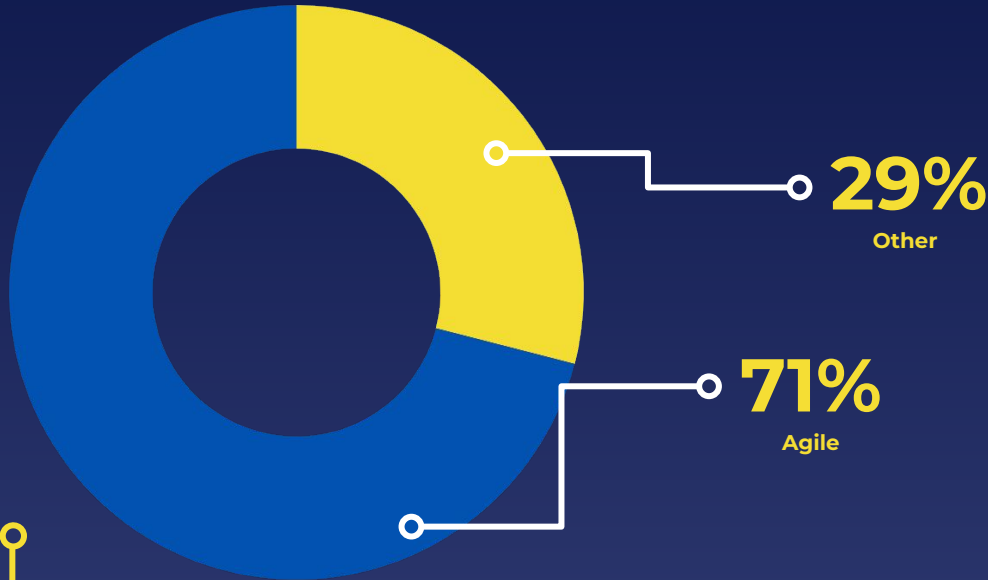
Max Mosier

### Client

Dr. Eck Doerry



# Tech Companies Use of the Agile Methodology



- Agile software development is a programming practice centered around small teams.
- According to Goremotely, over 71% of tech companies either already use, or are adopting agile methods.
- This number is rapidly increasing as agile continues to dominate the waterfall method.

# The Difficulty in Managing and Maintaining Teams

- Team based classes are difficult to organize due to how many moving parts they entail
- Keeping up with submissions and ongoing processes can be tedious
- Menial tasks take up a lot of time
- Communication between different parties is difficult to track



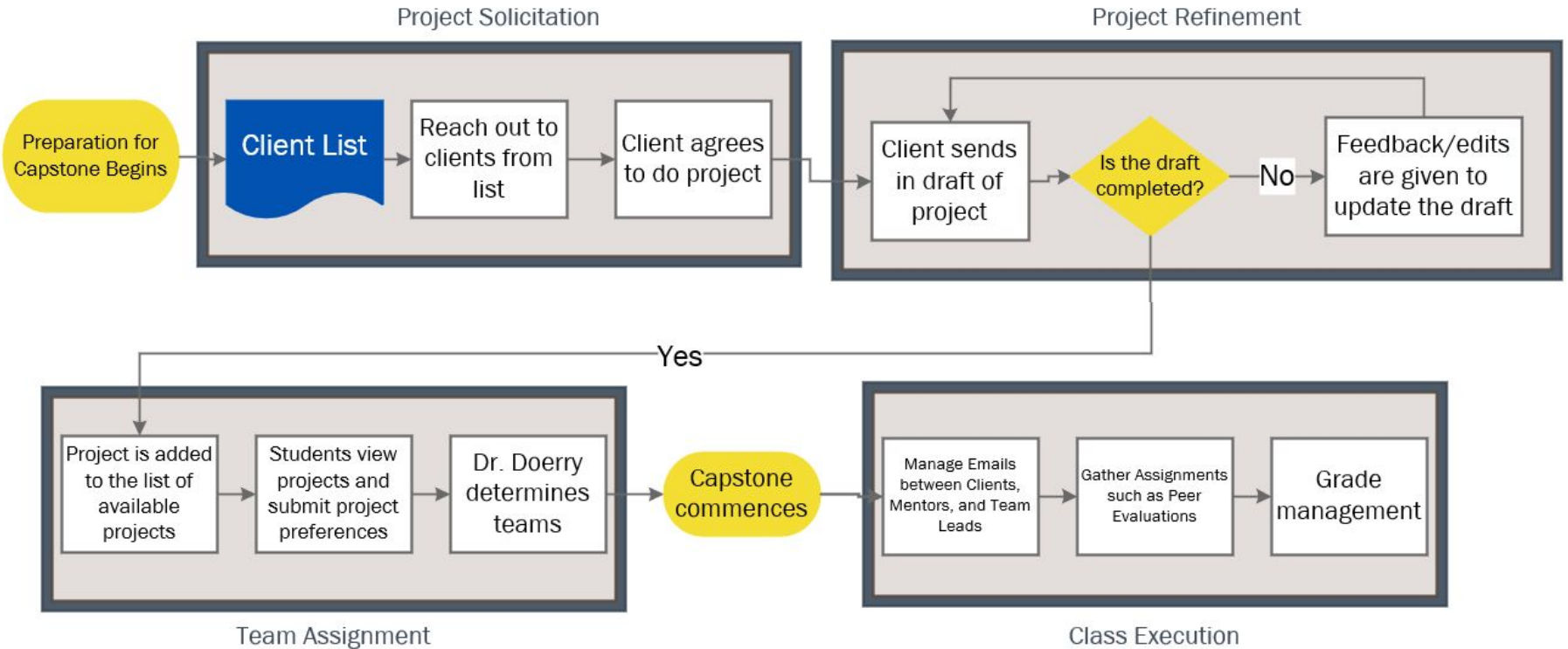
# Our Client

- Computer Science Professor at Northern Arizona University
- Has lead CS capstone project for 15+ years
- Is one of many faculty members that understands the moving parts in organizing a team-based course

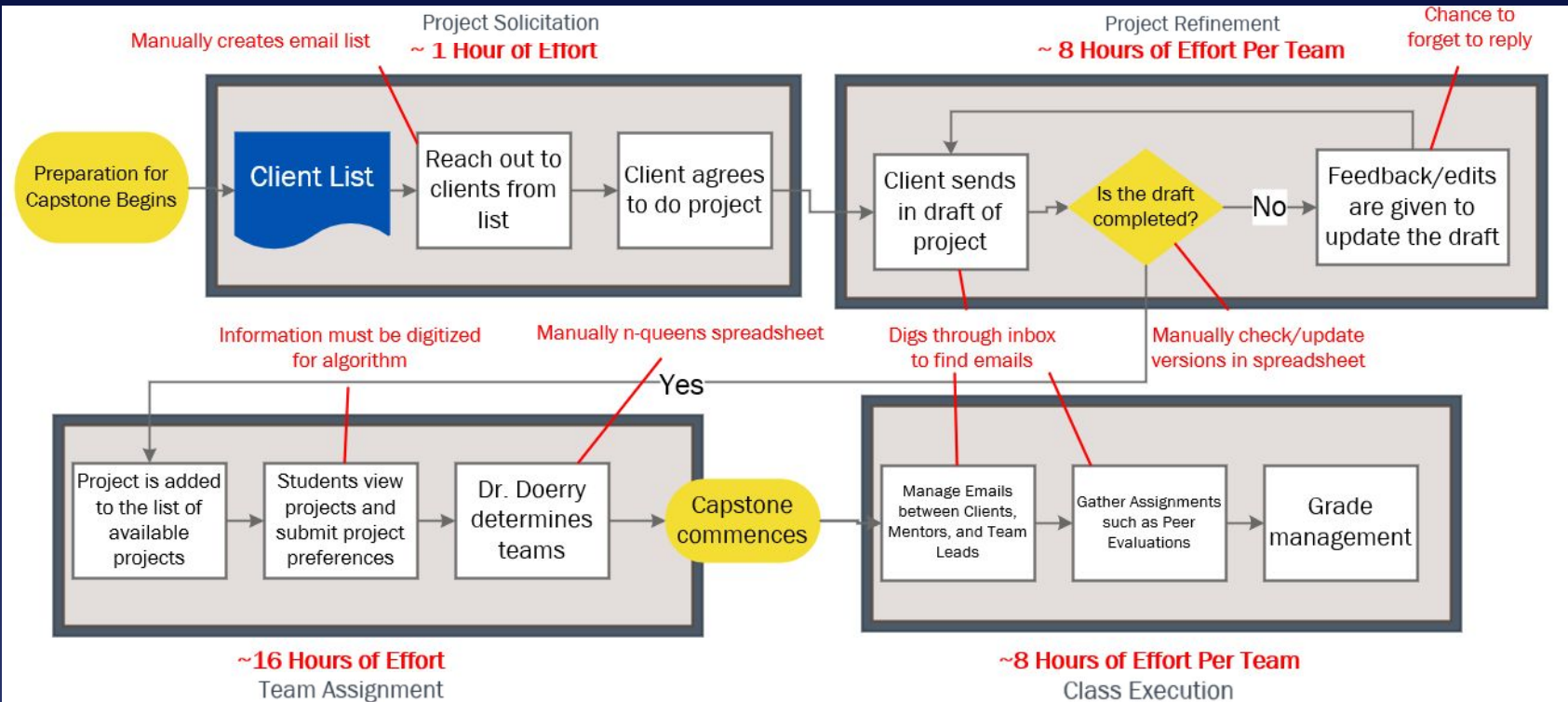


Dr. Eck Doerry

# Dr. Doerry's Current Workflow



# Problems with the Workflow



Based on 2021-2022 Capstone, Total Hours of Effort: 401 Hours of Effort Per Capstone

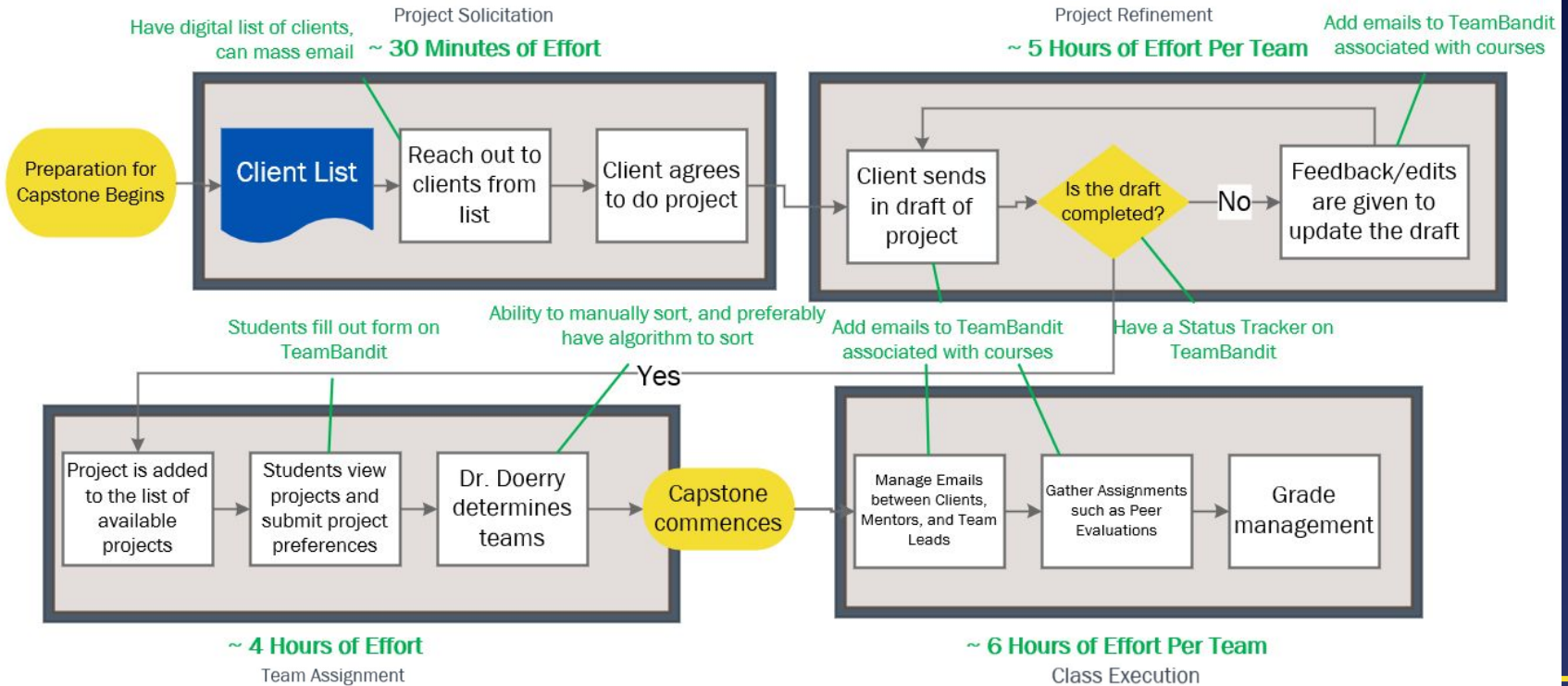
# Solution Overview

Solution - Develop a web application that allows a user to:

- Store information from emails associated with TeamBandit
- Have a digital list of clients to email in one place
- Students can fill out specific deliverables to store information automatically
- Tool for manually/algorithmically sorting students into teams



# Improved Workflow



Based on 2021-2022 Capstone, Total Hours of Effort: 269 Hours of Effort Per Capstone, ~ 132 Hours of Effort Saved



# Requirements Acquisition

- The team and the client have met weekly
  - Some form of requirements have been mentioned in every meeting
- Throughout the week, the team would consult together on which requirements were made by the client
  - New requirements were making themselves apparent every week
- After completing our tech feasibility document, we have worked with our chosen technologies
  - New requirements and solutions were identified as we worked with our technologies

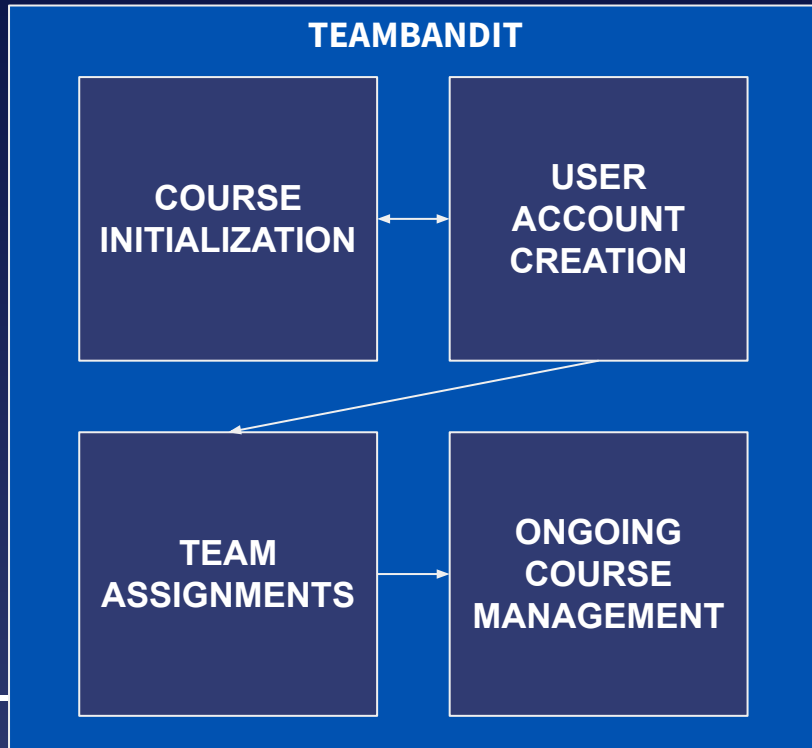
# Domain Requirements

The main requirements that this web application should fulfill:

- Help alleviate our client's task of keeping track of a mass amount of emails
- Allow users to create an account that gives them access to their assigned courses and projects
- Provide tools for creating projects and assigning students to them
- Provide tools for managing and editing projects and user information



# Requirement Separation



## Course Initialization:

Number of Functional Requirements: 24  
Number of Performance Requirements: 4

## User Account Creation:

Number of Functional Requirements: 17  
Number of Performance Requirements: 6

## Team Assignments:

Number of Functional Requirements: 25  
Number of Performance Requirements: 8

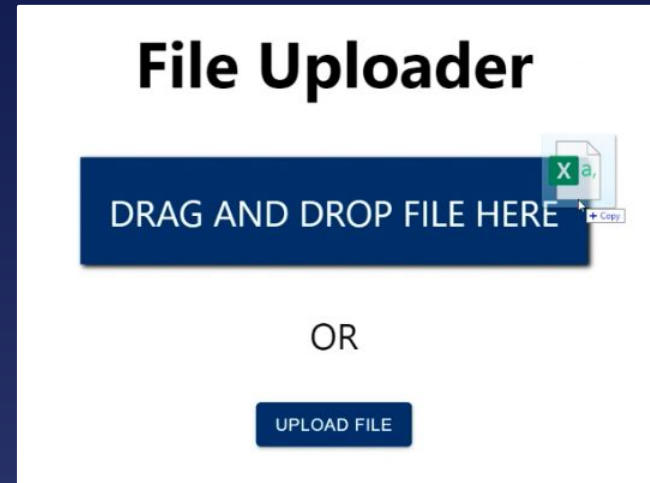
## Ongoing Course Management:

Number of Functional Requirements: 16  
Number of Performance Requirements: 7

There were no environmental constraints imposed on this project

# Functional Requirements

1. Faculty can edit project objects to contain the correct information when team assignments are made.
  - a. Faculty has the ability to associate....
    - i. Faculty will be able to select from...
    - ii. Ability for faculty to view/download...
  - b. Ability to upload project descriptions...
    - i. Supports individual upload...
      1. ...
      2. ...
    - ii. Supports bulk upload...
      1. ...
      2. ...
  - c. Can add or remove clients from...
    - i. Faculty can access a page with...
      1. ...



# Performance Requirements

1. 85% of faculty can begin editing a project's details in 2 minutes after two attempts.
2. 90% of faculty will have the ability to associate a single student with a project in 30 seconds after one attempt.
3. 80% of faculty can upload a PDF for a single project description in 30 seconds after three attempts.
4. 90% of faculty can add or remove a client from a project in 1 minute after two attempts.



RISKS	LIKELIHOOD	SEVERITY	MITIGATION
SERVER GOES DOWN	LOW	MEDIUM	Monitor AWS server
COMPETITOR COMES OUT WITH SIMILAR PRODUCT	LOW	MEDIUM	Competition would require us to innovate new features to bring clients to our product
MEMBER TEMPORARILY UNABLE TO CONTRIBUTE	MEDIUM	MEDIUM	Cross training and documentation to ensure no task is only capable of being completed by one member
DATABASE CORRUPTION	LOW	HIGH	Save information on separate devices as a backup
TEAMBANDIT EMAIL NOT CARBON COPIED (CC)	MEDIUM	MEDIUM	When user error is committed, the email can be manually forwarded to TeamBandit by the system administrator

# Feasibility Results

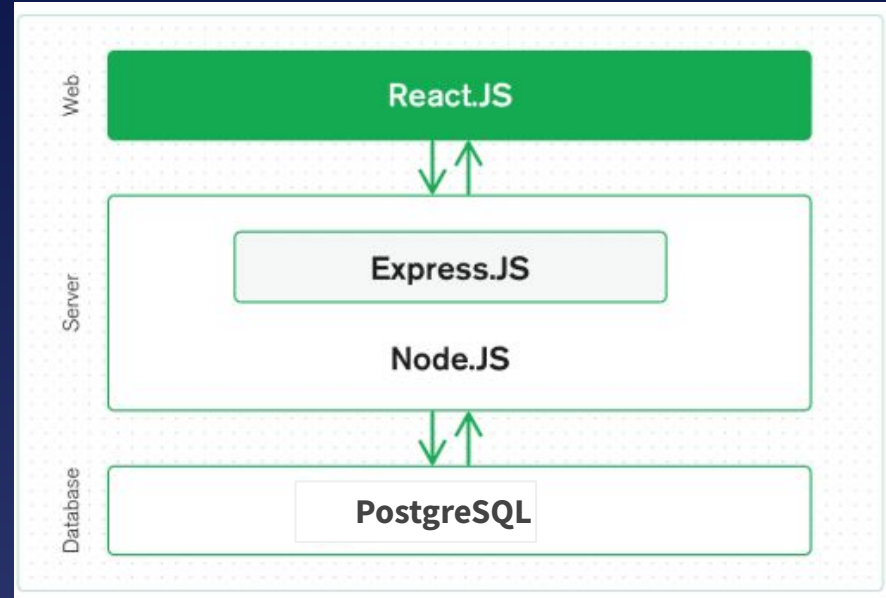
**Database Management System:** PostgreSQL

**Front-end Framework:** React

**Email Parsing Tool:** ~~Zapier~~ Built-in Python Library

## PERN Stack

- PostgreSQL used for backend stability that supports complex operations
- Express and Node used to run server
- React as a front-end framework



# Schedule

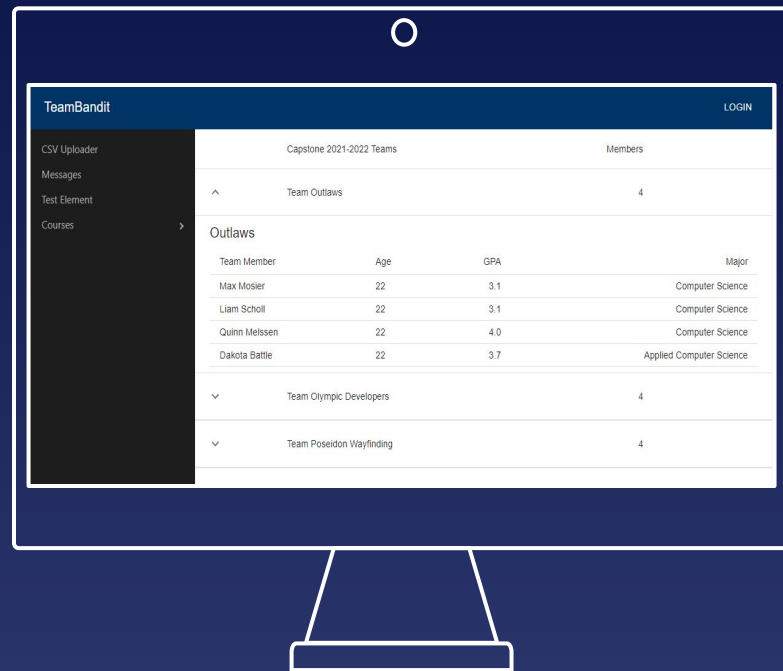
## TeamBandit Portal Development Plan





# Conclusion

- Small teams are increasingly gaining popularity in the real world
- Managing a team-based course is an arduous but important task
- TeamBandit will streamline the process of managing team-based courses



# Questions?

## Team Outlaws

*Click on our pictures for more info!*



**Quinn Melssen**

Team Leader  
qjm7@nau.edu



**Liam Scholl**

Release Manager  
las589@nau.edu



**Max Mosier**

Recorder  
mlm886@nau.edu



**Dakota Battle**

Architect  
db2354@nau.edu

<https://ceias.nau.edu/capstone/projects/CS/2022/Outlaws/>