MICHIGAN STATE UNIVERSITY

Project Plan

Integration & Testing Suite for ADAS Sensors

The Capstone Experience

Team Bosch

Jana Holderbaugh
Jesse Mcclay
Evan Martin
Wei Li
Nick Grenn

Department of Computer Science and Engineering
Michigan State University

Fall 2019

From Students…
…to Professionals

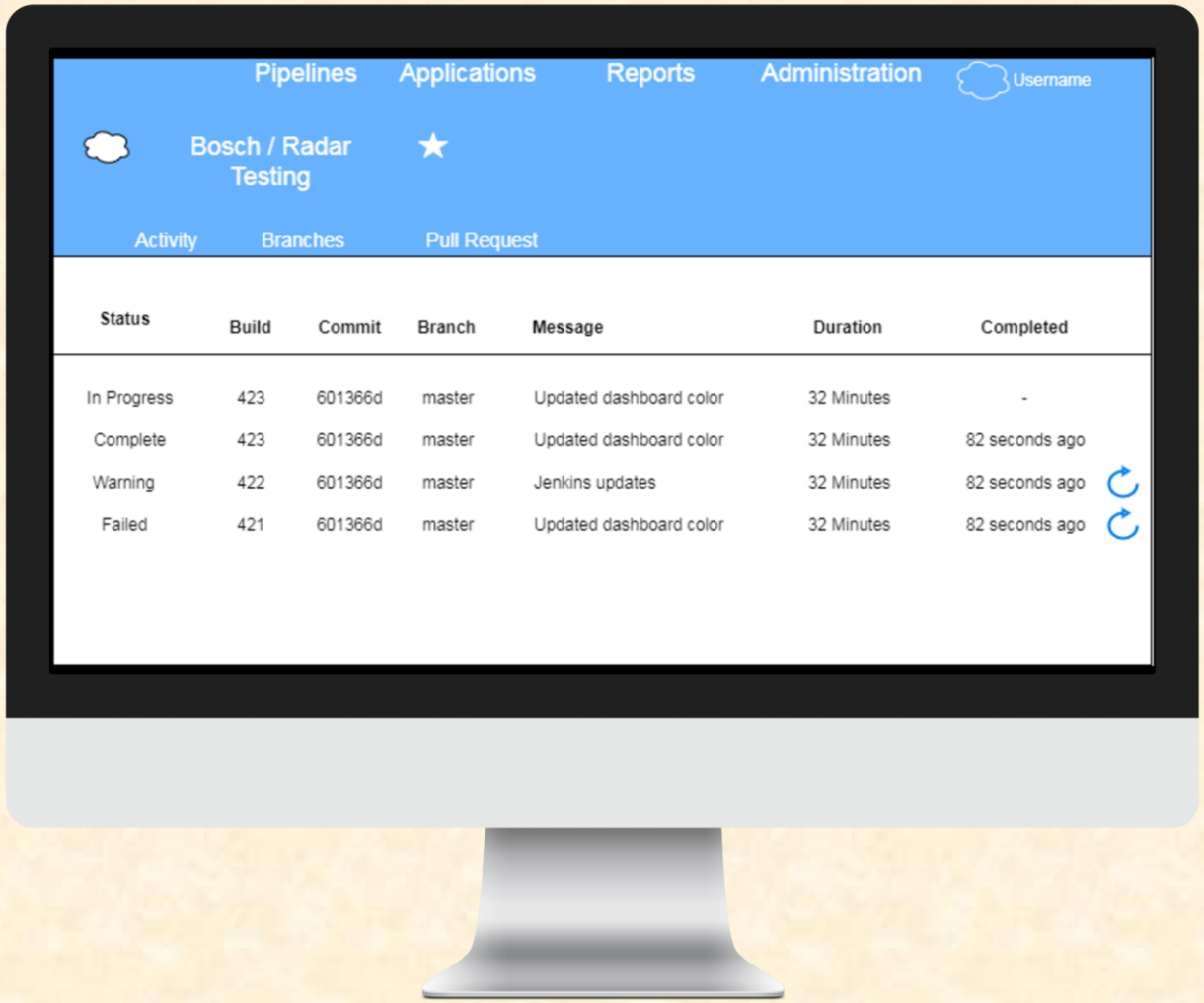
Functional Specifications

- Developers writing code for ADAS Sensors in version control system
  - Push to get ADAS systems to market quickly is higher than ever
- Current testing involves a compile test and some manual functionality tests
  - Manual functionality tests slow down process greatly
- CICT suite automates manual functionality tests
  - Automation of sensor tests can speed up testing process by days
- Failing functionality tests stops developer from merging code to higher path
  - Catches errors in testing instead of in vehicle

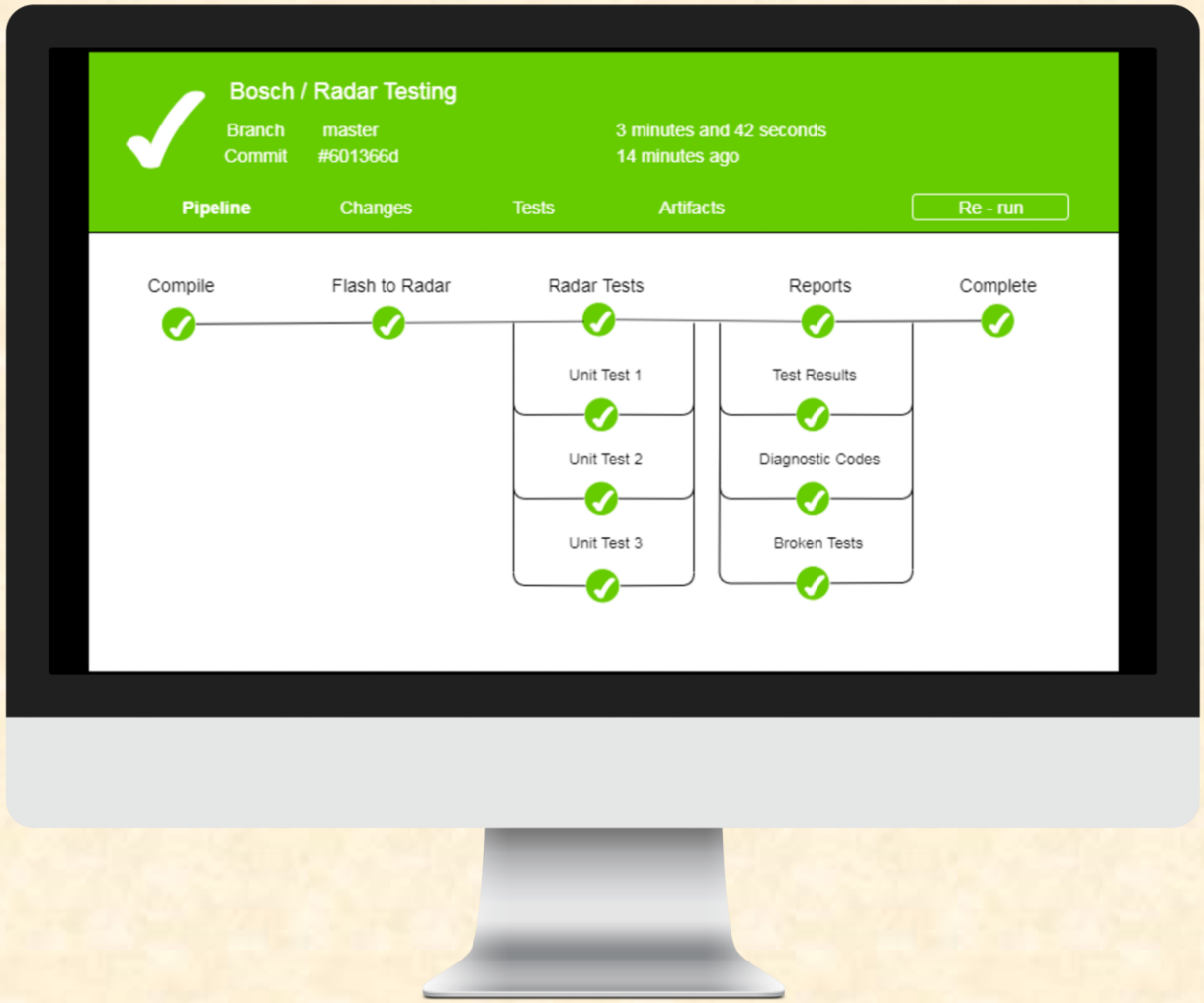
Design Specifications

- Visualized Pipelines
  - Capability to see status of each test as it runs
- Dashboard of running jobs
  - Includes info like job status, build number, commit number, branch name, commit message, duration of build and time completed
- Send email notifications when tests broken
  - If someone runs a build and that fails certain tests, an email notification will be sent to the test owner
- Powered by Blue Ocean
  - Plugin for Jenkins

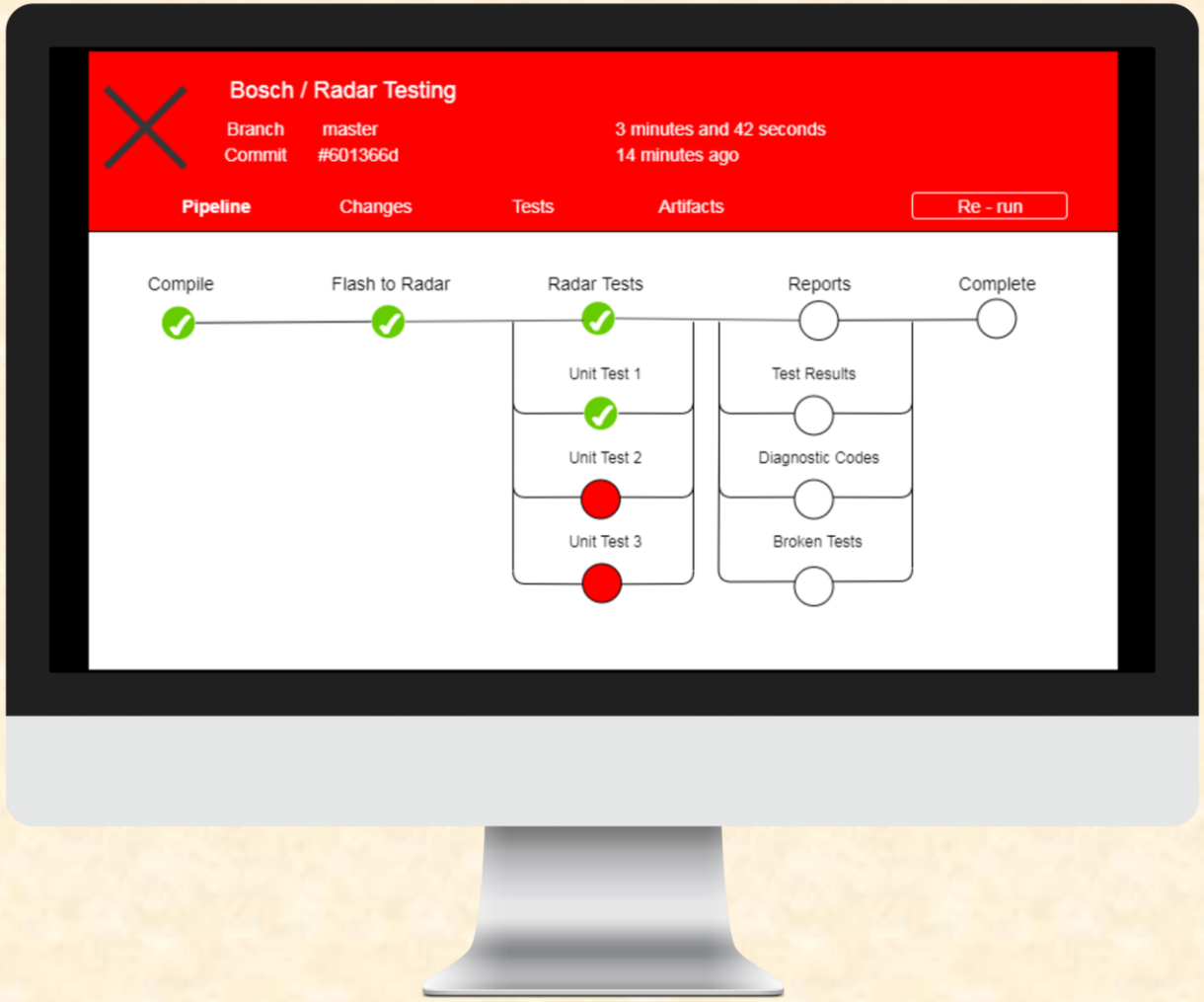
Screen Mockup: Main Dashboard



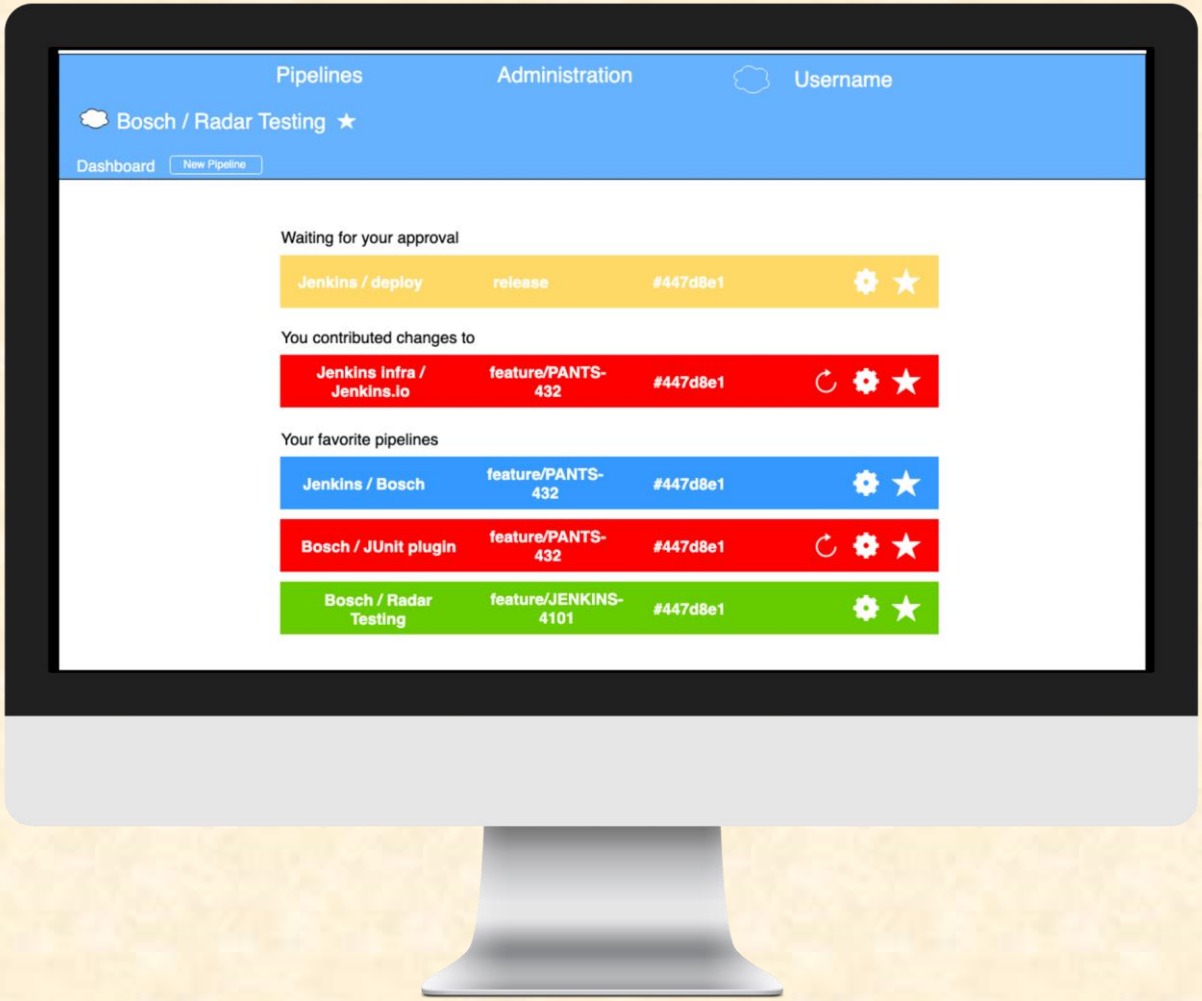
Screen Mockup: Successful Build



Screen Mockup: Failed Build



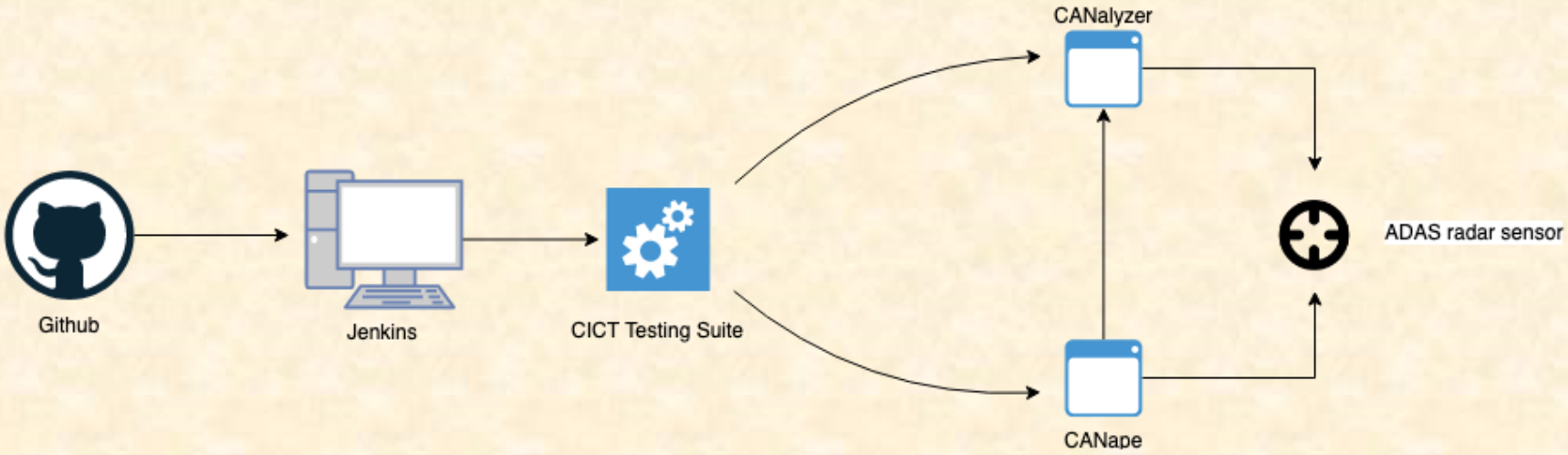
Screen Mockup: Personalized Dash.



Technical Specifications

- Git
  - Version Control system for code
  - Any git repository can be used, we use GitHub
- Jenkins
  - Continuous Integration software used in conjunction with a git repository
  - Used for building, deploying, and testing new code
- CANape
  - Used for calibration of radar sensor
  - Reports and records different values captured by radar sensor
- CANalyzer
  - Data analysis software that receives frames from CANape

System Architecture



System Components

- Hardware Platforms
  - ADAS radar sensor
- Software Platforms / Technologies
  - Git
  - Jenkins
  - CANape
  - CANalyzer

Risks

- Job Weight Distribution Optimization
  - Optimization of running processes requested; multiple machines if need be
  - Find the average amount of time it takes Bosch to finish a job, as well as the job load, run tests and simulations to find a formula that distributes tasks well
- Using CAN
  - No one on team very familiar with CAN protocol
  - Research CAN and software given to us, keep up constant contact with client for guidance
- Exporting CAN values
  - Accessing values from CAN software in external tests
  - Research documentation on python scripting and sending values out
- Radar Hardware and Software Limitations
  - Do not have readable code for radar to look at
  - Gather as much info about radar as possible, work with what we do know

Questions?