01/18: Team Status Reports

The Capstone Experience

Dr. Wayne Dyksen
Department of Computer Science and Engineering
Michigan State University
Spring 2018
AMAP: Automated Malware Analysis Platform

• Project Overview
  ▪ Having 268 million malware samples and growing at 300k per day.
  ▪ Categorize based on types, detection signatures byte patterns etc.
  ▪ Storing relevant information of samples in database.
  ▪ Results could be more samples of IP addresses and domains targeted.

• Project Plan Document
  ▪ Getting forward on technical specifications with client this week.
  ▪ Have a general outline.
  ▪ Most of the functional specifications just figured out.
  ▪ Overall progress: ~10%.
Team Accenture

Status Report

AMAP: Automated Malware Analysis Platform

• Server Systems / Software
  ▪ Windows 10 VM
  ▪ Ubuntu VM (Waiting to receive from client)
  ▪ Amazon AWS (Waiting to receive from client)

• Development Systems / Software
  ▪ iDefence IntelGraph (API access)
  ▪ iDefence Malware repository.
  ▪ Database (MongoDB, MySQL)
Team Accenture

Status Report

AMAP: Automated Malware Analysis Platform

• Client Contact
  ▪ Contacted with client last Thursday (in person).
  ▪ Daily communication on Slack.

• Team Meetings
  ▪ Had 4 team meetings
  ▪ Plan to meet twice a week.

• Team Organization
  ▪ Client contact Andrew.
  ▪ Rotate team lead.
  ▪ Every team member is considered a developer.
AMAP: Automated Malware Analysis Platform

Risks

• Risk 1
  ▪ Can we actually identify malware?
  ▪ Looking up how malware stored in the different file types.

• Risk 2
  ▪ Can we actually find how one malware sample connected to another?
  ▪ Learning how the iDefence tools can help detect patterns and malware samples.

• Risk 3
  ▪ Can we properly use iDefence tools?
  ▪ Using documentation from client.

• Risk 4
  ▪ How to process in multi-threaded environment?
  ▪ Looking tutorials on multi-thread programs.
AMPERED

• Project Overview
  ▪ Loyal, well-educated listeners
  ▪ Episode sponsors provide the main source of revenue
  ▪ Primary goal: recommend relevant Amazon products based on podcast audio content
  ▪ Content producers will receive commission on recommended item revenue
  ▪ Machine learning model and robust API are top priority
  ▪ Front end and UI are secondary

• Project Plan Document
  ▪ Functional specifications well defined by client
  ▪ Primary features identified
  ▪ System components identified
  ▪ System architecture diagram outlined
  ▪ Risks/mitigation identified
  ▪ ≈ 40% Complete
Team Amazon

Status Report

AMPED

• Server Systems / Software
  ▪ AWS EC2, RDS, Elasticache, S3, Beanstalk (Infrastructure)
  ▪ AWS Lambda, API Gateway, Route 53 (Front-end facility)
  ▪ AWS Transcribe, Comprehend, SQS, SNS (Middle-ware)

• Development Systems / Software
  ▪ Ubuntu Server, GitLab.MSU, Google Drive, Slack, Trello
  ▪ JavaScript, Python, AWS Services and APIs
  ▪ Web application first, with portability in mind
Team Amazon

Status Report

AMPED

• Client Contact
  ▪ Remote conference call, 9:00 AM every Friday
  ▪ In-person meetings are planned

• Team Meetings
  ▪ Three set meeting times per week
  ▪ 1 Hour meetings

• Team Organization
  ▪ Front-end focused workgroup
  ▪ Back-end focused, and research workgroup
Team Amazon

Status Report

AMPED

Risks

• Risk 1
  ▪ Recommending an item which has been negatively described
  ▪ Machine learning and sentiment analysis (feature of AWS Comprehend)

• Risk 2
  ▪ Controversial or inappropriate subjects
  ▪ Blacklist specific terms, utilize ‘IsAdultProduct’ attribute included in API

• Risk 3
  ▪ Portability, deployment, scalability
  ▪ Separation between back-end (APIs) and front-end (UIs)

• Risk 4
  ▪ Statistics and revenue reports for podcasters (stretch goal)
  ▪ Utilize Amazon affiliate program for purchase tracking
Team Aptiv

Status Report

Cyber Security Management System

• Project Overview
  ▪ Interdepartmental communications/information sharing suite
    o Information sharing between developer and security teams
  ▪ Automation of the entire cybersecurity process
    o Assessment request
    o Threat Analysis & Risk Assessment (TARA)
    o Vulnerability/Penetration Assessment
    o Vulnerability Remediation
    o Final Report

• Project Plan Document
  ▪ Not started
    o Client changed the full project proposal several times
Team Aptiv

Status Report

Cyber Security Management System

• Server Systems / Software
  ▪ WebApp
    o Javascript
    o HTML,CSS
    o PHP
  ▪ Database
    o MySQL
  ▪ Vmware ESXi – Hypervisor (VM Platform)

• Development Systems / Software
  ▪ WebStorm/PHPStorm
  ▪ Git repo
Team Aptiv
Status Report

Cyber Security Management System
• Client Contact
  ▪ Weekly Conference Calls (x2)
    ❖ Tuesdays 2:00-2:45pm
• Team Meetings (x2)
  ▪ Reviewed client proposal
  ▪ Change project proposal
  ▪ Met with Dr. Enbody/Pranshu Bajpai
  ▪ Weekly Team Meetings
    ❖ Monday 4:30-6pm
• Team Organization
  ▪ Developing the Cyber Security Management System
Cyber Security Management System

Risks

• Application security
  ▪ Software/Database will hold all of Aptiv’s data (schematics, software, vulnerabilities, etc.) for all of their products
  ▪ Project advisors (Pranshu/Enbody) have commercial web app pen-testing experience, and we will be implementing best practice security measures as we develop the system

• Database Implementation
  ▪ Best way to design a database to accommodate fast retrieval, and manage information for user-base with a wide range of specific permissions
  ▪ Focus initial development on strong database design instead of focusing on other things like the UI

• Knowledge of client procedure
  ▪ No knowledge of client complex process from start to finish vulnerability testing
  ▪ Reviewing documentation, close contact with client while constant prototyping

• Scalability issues with users
  ▪ How well the database will handle multiple users doing the same operations
    ○ Aptiv has 147,000 employees
  ▪ Use a cloud service (Amazon) instead of in-house servers to handle workload
Team Auto-Owners

Status Report

IMAGINE: IMAGe INtake Experience

• Project Overview
  ▪ Object recognition and classification of physical environments for insurance purposes
  ▪ Users load 360° images into web application
  ▪ Web-viewable database for inventory of identified objects
  ▪ Unity VR application to fully experience the environment

• Project Plan Document
  ▪ Have begun rough draft of Project Plan Document
  ▪ Currently reviewing project requirements, calculating risks and mitigation strategies, and designing overall system architecture
  ▪ Confirming overall plan with client on 1/19
  ▪ 20% Complete
Team Auto-Owners

Status Report

IMAGINE: IMAGe INtake Experience

• Server Systems / Software
  ▪ Capstone Server with Ubuntu Server – Up but Firewall Issues
  ▪ Apache Web Server with PHP – Up and Configured
  ▪ MariaDB – Up and Configured but potential Firewall Issues

• Development Systems / Software
  ▪ Unity Game Development Studio- Configuring
  ▪ Windows Machine with Oculus Rift and Controllers – Acquiring (Oculus Rift and Controllers have been acquired)
  ▪ OpenCV and TensorFlow - Configuring
Team Auto-Owners

Status Report

IMAGINE: IMAGe INtake Experience

• Client Contact
  ▪ Met with client at their headquarters in Lansing, MI on Monday 1/15/2018 from 11:30am to 1:30pm.
  ▪ Plan to have weekly conference calls on Fridays at 11:30am

• Team Meetings
  ▪ Have had three team meetings thus far
  ▪ Official team meetings are planned for Thursday after class

• Team Organization
  ▪ Tasks will be specialized between members
  ▪ 3 groups – Classifier Systems, Virtual Reality, Web Interface
Team Auto-Owners

Status Report

IMAGINE: IMAGe INtake Experience

Risks

• Object recognition in spherical images
  ▪ 3D images have distorted pixel densities and will make classifying difficult
  ▪ Normalize 3D images to 2D or include warped images when training our classifier

• Inability to classify an environment
  ▪ Environments should be classified based on types of objects found (i.e. bedroom, office, etc.)
  ▪ Train negative classifiers to drop incompatible environments

• Multiple concurrent users
  ▪ Multiple separate workflows will need to be able to be accessed by all users
  ▪ Manage interactions with a user system using transactions to enforce ACID

• Server Access Limited by MSU Firewall
  ▪ MSU has unknown firewall rules that prevent some outside communications
  ▪ Determine what is prohibited and pipeline traffic through approved channels
Team Dow

Status Report

Virtual Reality Simulation for Railcar Loading

• Project Overview
  ▪ Teach How to Load a Railcar Safely
  ▪ Achieved Through First Person Virtual Reality

• Project Plan Document
  ▪ Table of Contents Finished
  ▪ Executive Summary Started
  ▪ Risk Analysis Finished
  ▪ Schedule Drafted
  ▪ 20% Complete
Team Dow

Status Report

Virtual Reality Simulation for Railcar Loading

- Server Systems / Software
  - No Servers

- Development Systems / Software
  - Computer with GTX 1060 or better (obtained)
  - HTC Vive Headset (obtained)
  - Unity Game Engine (installed)
  - Maya 3D (installed)
  - Photoshop CS6 (installed)
  - Audacity (installed)
Team Dow

Status Report

Virtual Reality Simulation for Railcar Loading

• Client Contact
  ▪ Emailed twice and had conference call
  ▪ Weekly conference calls Friday at 12:30pm

• Team Meetings
  ▪ Weekly meetings on Tuesdays at 4:30pm
  ▪ Weekly meetings with Johnny on Thursdays at 4:40pm

• Team Organization
  ▪ Using GroupMe group chat for instant communication
  ▪ Using Trello for project role/tasks organization
Virtual Reality Simulation for Railcar Loading

Risks

• Unity Game Engine
  ▪ Description: Understand development with Unity
  ▪ Mitigation: Follow online tutorials through Unity, websites, YouTube, etc.

• Vive VR Development
  ▪ Description: Understand how HTC Vive works with Unity and what works in a VR environment
  ▪ Mitigation: Follow online tutorials, build test scenes for basic VR interaction

• Project Assets
  ▪ Description: Acquiring realistic models and sounds for development
  ▪ Mitigation: Search Unity Asset store and royalty free websites for assets

• Accurate Simulation
  ▪ Description: Accurately replicating scenario of loading railcars
  ▪ Mitigation: Watch/analyze videos of proper railcar loading/filling
Team DRIVEN-4

Status Report

2020 Business in a Box

• Project Overview
  ▪ Base on Internet of Things (IoT) architecture
  ▪ Showcase future business environment with 2020 as target
  ▪ Model a connected product utilizing Wi-Fi for connectivity
  ▪ Develop manufacturing processes and artifacts
  ▪ Demonstrate capabilities for collaboration and integration

• Project Plan Document
  ▪ Have not started
Team DRIVEN-4

Status Report

2020 Business in a Box

• Server Systems / Software
  ▪ No server needed

• Development Systems / Software
  ▪ CAD – Siemens NX, PTC Creo
  ▪ PLM – Siemens Teamcenter, PTC Windchill
  ▪ IoT Platform – PTC Thingworx, Siemens MindSphere
  ▪ AR/VR – PTC Thingworx Studio
  ▪ Factory Floor Simulation – Siemens Tecnomatrix
  ▪ Waiting on access
Team DRIVEN-4
Status Report

2020 Business in a Box
• Client Contact
  ▪ Conference calls scheduled for Fridays at 1pm
    ◦ One so far
  ▪ In-person meeting scheduled for Thursday 11/18
• Team Meetings
  ▪ Two so far
  ▪ Weekly meetings on Wednesdays
• Team Organization
  ▪ Client Contact – Sam Coffey
  ▪ Technical roles not defined at this time
Team DRIVEN-4

Status Report

2020 Business in a Box

Risks

• Embedded Software Design Experience
  ▪ No team members have experience with embedded software design
  ▪ Individual research and training from DRIVEN-4

• Product Use Visualization
  ▪ No team members have experience with visualization design
  ▪ View examples provided by DRIVEN-4 and research visualization design fundamentals and samples

• Hardware Familiarization
  ▪ Can’t get familiarized with hardware until received
  ▪ Meeting scheduled to receive hardware

• Hardware-Software Interaction
  ▪ Need to determine how to get devices interacting
  ▪ Get training from DRIVEN-4 once devices received
Team GM

Status Report

Plato

• Project Overview
  ▪ Artificially Intelligent Dev Bot for Microsoft Teams
  ▪ Create and Manage Virtual Machines via Bot and Web App
  ▪ Manage and Run Test Cases
  ▪ Provide Single Unified Environment for Developers

• Project Plan Document
  ▪ 20% complete
  ▪ Outline done, schedule done
  ▪ Working on database schema, system diagram, architecture
  ▪ Working on functional and design specifications
Team GM
Status Report

Plato

• Server Systems / Software
  ▪ Obtained Microsoft Teams Account from GM
  ▪ Setup Microsoft Team for testing
  ▪ SQLServer Standard 2017 getting set up

• Development Systems / Software
  ▪ VMWare and Windows 10 installed on both iMacs
  ▪ Visual Studio, Azure, and Microsoft Teams installed
  ▪ Hello World Bot written with MBF
Plato

- **Client Contact**
  - Established Weekly meeting (Tuesday 9 - 10AM)
  - Requirement specification meeting tomorrow 1-2PM
  - Have had 2 total client meetings

- **Team Meetings**
  - First Triage meeting before class at 2:20PM
  - 3 total team meetings
  - Team meetings scheduled for 4:30-5:00PM Tuesday/Thursday – more to come.

- **Team Organization**
  - Client Contact/Project Manager: Colin Coppersmith
  - Web Application Development: Tao Tao/Colin Coppersmith
  - Backend and Bot Developer: Matthew Eaton/Simeon Goolsby/Alex Lepird
Team GM

Status Report

Plato

Risks

• Controlling and managing Virtual Machines programatically
  ▪ Description: No experience with CRUD using a programming language.
  ▪ Mitigation: Use C# backend to communicate with Azure.

• Implementing acceptable language processing to ensure bot can understand commands effectively
  ▪ Description: Thousands of ways to execute each command.
  ▪ Mitigation: Use grammars and NLP to fill in the blanks.

• Integrating Microsoft TFS to automate Test case creation and testing
  ▪ Description: GM uses TFS to test applications, no experience using it.
  ▪ Mitigation: Gather unit tests from GM, try to emulate style.

• Customizing bot interaction based off team/ individual user
  ▪ Description: Need to tailor to the needs of each team/user.
  ▪ Mitigation: Using emails/IDs to determine which user against records.
AR Adjust App
• Project Overview
  ▪ Native iOS app for customers using Herman Miller adjustable office chairs
  ▪ Augmented Reality detects and identifies model of chair from camera
  ▪ AR technology highlights adjustable parts and shows tooltip descriptions of adjustments for that model

• Project Plan Document
  ▪ Writing the rough draft
  ▪ Overall ~10% complete
  ▪ Basic points on the design, functional and technical specifications
  ▪ Early screen mockups
Team Herman Miller

Status Report

AR Adjust App

• Server Systems / Software
  ▪ Windows Server – Setting up currently
  ▪ Git for source control

• Development Systems / Software
  ▪ Unity3d – Up and running
  ▪ Vuforia (for AR) – Testing, awaiting on approval for pro license
  ▪ Xcode – Up and running
AR Adjust App

• Client Contact
  ▪ Team visited office in Zeeland, Michigan and met with team / toured chair facility
  ▪ Weekly conference meetings scheduled with Herman Miller team members, planning second and final on-sites

• Team Meetings
  ▪ Tuesdays before class
  ▪ Client meetings Wednesday afternoon

• Team Organization
  ▪ Client Contact / UI Developer – Kyle Kinsey
  ▪ AR Developer – Mike Bremiller, Kevin Gaban
  ▪ UI Developer – Jacob Weber, Han Huang
Team Herman Miller

Status Report

AR Adjust App

Risks

- **Risk 1**
  - Integrating Augmented Reality into the app
  - Testing different software solutions (Vuforia, Arkit)

- **Risk 2**
  - Ability to recognize chair model via camera
  - Obtaining physical chairs and pictures to train models

- **Risk 3**
  - Learning to develop for iOS devices
  - Developing application with Unity3d in C#, making basic Swift applications

- **Risk 4**
  - Cross platform app development
  - Using Unity and Vuforia (vs. Apple Arkit), which can create both native Android and iOS apps
Team Meijer

Status Report

Personal Shopping Assistant

• Project Overview
  ▪ Simplify shopping experience (at home and in store)
  ▪ Ask app instead of a team member
    o Item locations, availability, coupons, etc.
  ▪ Create bot to answer natural language questions
  ▪ Ensure API is universal, can be consumed for other projects

• Project Plan Document
  ▪ Outline/Table of Contents finished
  ▪ Shared via OneDrive for collaboration
  ▪ 10% Complete
Team Meijer
Status Report

Personal Shopping Assistant

- Server Systems / Software
  - Microsoft Azure – have access
  - Meijer Web Services – do not have access yet, pending
  - SQL/Mongo Server(s) – not created/accessible yet

- Development Systems / Software
  - Android Studio – installed and running
  - Xcode – installed and running
  - Version Control – access pending
Team Meijer

Status Report

Personal Shopping Assistant

• Client Contact
  ▪ Had 3 Conference Calls
  ▪ Weekly 45 min. call scheduled (2:15-3 Thursdays)

• Team Meetings
  ▪ Tuesdays 2-3
  ▪ Met twice to get iMacs/VMs setup

• Team Organization
  ▪ Corporate Contact – Zach
  ▪ Android/Java – Emerson and Aaron
  ▪ iOS/Swift – Megan and Jake
  ▪ Bot Backend/C# Lead – Zach
Team Meijer

Status Report

Personal Shopping Assistant

Risks

• Bot
  ▪ Need to implement a Natural Language bot
  ▪ Zach will be dedicated lead for this section of the project but every team member will research and contribute

• Item location in each store
  ▪ We'll need to determine where the item is in the store at which the customer is shopping.
  ▪ Testing the app at several Meijer locations in the Lansing area to make sure the information is accurate.

• UI
  ▪ Meijer has not settled on an app design (pure chatbot vs menus vs ?) and will require a large variety of screen mocks to make a decision
  ▪ Will create lots of screen mockups and get feedback as often as possible

• Bluebird Integration
  ▪ A stretch goal of this project is to incorporate team member assistance via bluebird devices.
  ▪ Work with Meijer to get a device and communicate with team at Meijer who work with or develop for the devices. May be able to contact the vendor directly.
Student Engagement App

• Project Overview
  - Expand learning inside and outside of the classroom
  - Create a universal classroom response tool
  - Allow students to use mobile devices to engage
  - Streamline and simplify attendance

• Project Plan Document
  - Outline of document in place
  - Risks have been identified
  - Initial UI mockups have been created
  - Created development process
Student Engagement App

- **Server Systems / Software**
  - Set up Amazon Web Services with Flask and Python
  - Explored database and storage options (Dynamo, SQL, etc.)
  - Prototyped entity relationship diagram

- **Development Systems / Software**
  - Set up Android Development Environment and initial project
  - Set up iOS foundation and initial project
  - Started VUE project for web application
Team Michigan State University

Status Report

Student Engagement App

• Client Contact
  ▪ Met in person at MSU, set up reoccurring meetings
  ▪ Gathered initial requirements and resources

• Team Meetings
  ▪ Plan to meet Tuesdays and Fridays to work together
  ▪ Paired programming development

• Team Organization
  ▪ Set up Slack for communication
  ▪ Created Git group, and using Trello for progress tracking
Student Engagement

Risks

• iBeacon compatibility with Android
  ▪ iBeacon technology was developed by Apple for iOS
  ▪ Research existing Android libraries and create a basic app that can connect

• Create a positive experience for students AND faculty
  ▪ Students and faculty have different priorities in classroom involvement
  ▪ Conduct iterative user testing with both groups throughout development

• Align Amazon Web Services with required technology
  ▪ Technology recommended by Amazon does not fulfill requirements of the app
  ▪ Create a basic lab that utilizes all software

• Data input has to be scalable
  ▪ Interaction from students will come in large portions at a time
  ▪ Develop with scalability in mind and conduct rigorous testing with high volume.
Mozilla

• Project Overview
  ▪ Expand Firefox Theming API to allow for theming of previously unthemable browser aspects, like bookmarks and scroll bars.
  ▪ Expand theming API to allow for Google Chrome extensions to be easily transitioned over as Firefox extensions
  ▪ Create new themes for the Firefox browser
  ▪ Resolve existing bugs and issues with the theming API

• Project Plan Document
  ▪ Divided up work between team members
  ▪ Started writing schedule of project milestones
  ▪ Expect to have a first draft 1/26
  ▪ 5% done
Team Mozilla

Status Report

Mozilla

• Server Systems / Software
  ▪ All team members have the Firefox build environment downloaded and compiling on their systems.
  ▪ All team members have received level 1 access to the Firefox codebase
  ▪ All team members have created Bugzilla accounts and set up IRC chat.

• Development Systems / Software
  ▪ Both iMacs have the Firefox codebase downloaded and compiling
  ▪ Both iMacs have a Windows virtual machine running
  ▪ Both iMacs have a Ubuntu Linux virtual machine running
Team Mozilla

Status Report

Mozilla

• Client Contact
  ▪ We have emailed and met with our client
  ▪ Weekly video conference call scheduled for 3:00 P.M. on Fridays
  ▪ Hacking weekend with Mozilla on February 10 - 11

• Team Meetings
  ▪ Tues/Thurs: 4:30pm; Mon: 5:30pm
  ▪ Triage: Monday 2:20pm
  ▪ Our team has met 6 times so far

• Team Organization
  ▪ Vivek Dhingra is the client contact
  ▪ Assigned tasks to each team member through Bugzilla ticketing system
  ▪ Weekly team code review on Thursdays
Mozilla

Risks

• Large Codebase
  ▪ Firefox codebase is over 35 million lines code, finding a place to start is challenging
  ▪ Using the searchfox.com web tool to locate files of interest, rather than grep.

• Platform Testing
  ▪ Need to efficiently code for all platform without breaking compatibility. Limited team experience with testing suites
  ▪ Research and write basic tests in Mozilla’s testing suite.

• API Experience
  ▪ Unsure of the type of API (REST, SOAP, RPC). Limited team experience with API development.
  ▪ Building a basic API once Mozilla’s Theme API type is determined

• Theme Transitions
  ▪ Need to ensure compatibility when transitioning themes from Google Chrome
  ▪ Review resources to get a comprehensive understanding of Google Chrome themes. Additionally, build themes to further understanding
Team MSUFCU

Status Report

Digital Assistant and Personal Financial Coach

• Project Overview
  ▪ Digital assistant and financial coach for MSUFCU members
  ▪ Answers questions about member’s financial situation
  ▪ Gives members the ability to compare spending habits in same demographic
  ▪ Members can take action on their account to request/transfer funds

• Project Plan Document
  ▪ Created basic outline
  ▪ Completed system architecture mockup
  ▪ Started executive summary
Team MSUFCU

Status Report

Digital Assistant and Personal Financial Coach

• Server Systems / Software
  ▪ Will receive necessary hardware from MSUFCU
  ▪ Will receive access to database containing dummy accounts from MSUFCU
  ▪ This database is for testing only, and is not connected to their main database

• Development Systems / Software
  ▪ Installed necessary software on iMacs
  ▪ Will receive previous source code from MSUFCU
  ▪ Tested each program installed to verify they are working correctly
Team MSUFCU

Status Report

Digital Assistant and Personal Financial Coach

• Client Contact
  ▪ Met with client and signed an IP agreement and Non-Disclosure Agreement
  ▪ Discussed the resources that we will be using and the devices the client will be providing

• Team Meetings
  ▪ Installed Android Studio, Xcode, PHPStorm, and VMWare on iMac
  ▪ Planned weekly meetings at 5 PM on Tuesday and Thursday

• Team Organization
  ▪ Client Contact and Project Manager (Rachel)
  ▪ Machine Learning (Patrick)
  ▪ Database (Dallas)
  ▪ Web (Dane)
  ▪ Mobile Apps (Michael)
Team MSUFCU

Status Report

Digital Assistant and Personal Financial Coach

Risks

• Building off of Previous Code
  ▪ This project is building off of work done by 2 previous capstone teams
  ▪ We will work with clients to ensure our code works well with previous code; we will also contact old team members if necessary

• Working with Voice Recognition Software
  ▪ No experience with voice recognition or speech-to-text
  ▪ We will research best practices and use previous code to develop our knowledge

• Using Machine Learning to Make Predictions
  ▪ Making comparisons between members of similar demographics requires machine learning techniques
  ▪ We will research best methods for this type of data analysis and we will rely on clients to assist us.

• Integration of Android, Alexa, iOS, and Administrative Web App
  ▪ Making these systems communicate with each other may prove to be difficult
  ▪ We plan to use centralized database to maintain consistency between all different platforms
Customer Service System with Chatbot

• Project Overview
  - Customer service team references paper manuals
  - Digitize manuals using tablet camera
  - Browse and manage manuals in ebook format
  - Chatbot to answer customer questions

• Project Plan Document
  - 10% of final plan complete
  - Functional specification first draft
  - System architecture first draft
  - Identified major risks
Team Phoenix Group
Status Report

Customer Service System with Chatbot

• Server Systems / Software
  ▪ Linux installed on server
  ▪ Built toy client/server application on local machine
  ▪ Remote server access pending

• Development Systems / Software
  ▪ Installed Windows and Visual Studio 2017
  ▪ Tested C# hello world application
  ▪ Created toy chat bot with Microsoft Bot Framework
Team Phoenix Group

Status Report

Customer Service System with Chatbot

• Client Contact
  ▪ Weekly meetings on Fridays
  ▪ Upcoming functional specification draft review

• Team Meetings
  ▪ Weekly meetings on Tuesdays

• Team Organization
  ▪ Server application: Fatema, Amanuel
  ▪ Client application: Sarah, James
  ▪ Web chat bot: Dan
Team Phoenix Group
Status Report

Customer Service System with Chatbot
Risks

• Use of Docker when shipping application
  ▪ Lack of experience and compatibility issues
  ▪ Mitigation: Containerize a toy application

• Image quality when composing eBook and running OCR
  ▪ OCR system may require high image quality
  ▪ Mitigation: test OCR performance on images taken with tablet

• Dataset size and system scaling
  ▪ eBook size may be large, impacting performance
  ▪ Mitigation: OCR speed tests, client-server data transfer speed tests

• Chat bot embedding in client website
  ▪ Lack of experience
  ▪ Mitigation: Embed prototype chat bot on WordPress site
Next Generation Malware Detection

• Project Overview
  ▪ Reduce number of malware an analyst has to examine manually
  ▪ Provide malware analysis dashboard for analysts
  ▪ Produce real time signatures for malware undergoing dynamic analysis
  ▪ Schedule malware analysis efficiently

• Project Plan Document
  ▪ Estimated all but schedule and some technical specifications, under review by client
  ▪ Project Plan outline is 75% done.
  ▪ Wireframe for front end sketched
  ▪ System architecture sketched
Team ProofPoint

Status Report

Next Generation Malware Detection

• Server Systems / Software
  ▪ Web server – not set up awaiting client confirmation
  ▪ SQL Server – not set up awaiting client confirmation

• Development Systems / Software
  ▪ YARA, Cuckoo, Python – configured and tested
  ▪ Suricata, ClamAV – Not yet
  ▪ Linux, Windows, MAC OS – installed and configured
Next Generation Malware Detection

- Client Contact
  - Video conference call on Fridays starting January 12th
  - In person meeting scheduled on January 19th

- Team Meetings
  - Met 4 times so far
  - Weekly meetings – Wednesdays

- Team Organization
  - Brad is the client contact
  - Crystal is the project manager
Next Generation Malware Detection

Risks

• Clustering Malware
  ▪ What metric(s) do we use to cluster similar malware.
  ▪ Talk with analysts and/or client and research how malware can be clustered

• Scalability and Speed
  ▪ How our program can efficiently analyze malware provided
  ▪ Test speed of software and determine probability of dynamic analysis

• Processing Output of Software
  ▪ Analyzing the output of the detection software that we are using
  ▪ Prototype output parsing tools

• Constructing an API
  ▪ Give a way for the Web App to interact with the analysis tool via an API
  ▪ Research common ways to make an API and create a simple prototype API
Team Quicken Loans

Status Report

Fundamenta

• Project Overview
  ▪ Web Application for construction of a house
  ▪ Blockchain-based
  ▪ Allows interactivity between builder, buyer, and contractors
  ▪ Visual of workflow and transactions stored in the blockchain

• Project Plan Document
  ▪ Sections assigned to each team member
  ▪ Initial screen mock-ups completed and given to client
  ▪ Functional specifications have been discussed with the client
  ▪ Outlined and sections 20% complete
Team Quicken Loans

Status Report

Fundamenta

- Server Systems / Software
  - Azure
  - .NET (C#) backend
  - Private Ethereum blockchain initialized

- Development Systems / Software
  - React
  - Python (for blockchain)
  - Multiple “hello world” applications have been created
Team Quicken Loans

Status Report

Fundamenta

• Client Contact
  ▪ Scheduled weekly meetings on Wednesday afternoons
  ▪ On-site meeting scheduled January 31st

• Team Meetings
  ▪ 6 team meetings so far
  ▪ Scheduled weekly team meetings on Monday at 1 PM

• Team Organization
  ▪ Frontend / UX (Erin and Turner)
  ▪ Backend / Blockchain (Riley, Jaiwant, Vishal)
Team Quicken Loans

Status Report

Fundamenta

Risks

• Blockchain
  ▪ Applicability of proof-of-work and mining
  ▪ Create own blockchain and utilize in-house experts at Quicken Loans
• Ethereum-specific Challenges
  ▪ Usage of smart contracts for this project
  ▪ Have questions prepared for Friday call with client
• Setting Up Development Workflow
  ▪ Getting all of the technologies up and running cohesively will be a challenge
  ▪ Starting early, doing research, and asking questions
• Interaction with Database
  ▪ Setting up the blockchain to interact with SQL Server
  ▪ Small-scale testing with simple queries
Team Rook

Status Report

Endpoint Data Monitoring and Analysis Agent

• Project Overview
  ▪ Agent captures event logs on end point hosts
  ▪ Create web interface to configure agent
  ▪ Analyze health metrics based on logs
  ▪ Cross-platform compatible

• Project Plan Document
  ▪ 20% Completed
  ▪ Skeleton File Created
  ▪ Sections split up among team members
  ▪ Rough draft of system architecture created
  ▪ Initial mock user interface created
Team Rook

Status Report

Endpoint Data Monitoring and Analysis Agent

• Server Systems / Software
  ▪ Amazon S3 (access pending)
  ▪ RESTful Endpoint
  ▪ Ubuntu 16.04 Back-End API Server

• Development Systems / Software
  ▪ Go
  ▪ Django + Python
  ▪ React/Redux JS
Team Rook

Status Report

Endpoint Data Monitoring and Analysis Agent

- Client Contact
  - Weekly meeting set up for 2PM on Mondays
  - Constant Communication via Company HipChat Channel

- Team Meetings
  - Weekly Conference Call: Monday 2pm
  - Weekly Triage Meeting: Thursday 4:50pm
  - Weekly Group Meeting: Wednesday 6pm

- Team Organization
  - Client Contact: Drew Gilbertson
  - Team Structure: Web App group, Agent group, Database group
Team Rook

Status Report

Endpoint Data Monitoring and Analysis Agent

Risks

• Develop agent software that is cross-compatible
  ▪ Creating background process that captures event logs for all OS
  ▪ Understanding different OS event logs and how to capture them

• Developing health metrics to analyze captured logs
  ▪ Determining the thresholds for analyzing event logs in real time
  ▪ Research event log details and priorities, and conform to Rook standards

• Integration with existing Force Platform
  ▪ Create a web app which extends the existing platform. We need to maintain the Force Platform’s integrity.
  ▪ Develop using iterative process while working closely with Rook’s developers

• How to handle testing
  ▪ How to gain realistic data. How to evaluate security thresholds.
  ▪ Work with experts at Rook to fully understand realistic information flow
Team SpartanNash

Status Report

SpartanTrack

• Project Overview
  ▪ Track Volunteer hours
  ▪ Gamify the app, using badges/leaderboards
  ▪ Messaging from captain to the platoon
  ▪ Integrate Social media features

• Project Plan Document
  ▪ Began initial draft of the plan – 10%
  ▪ Began drafting our UML – 25%
  ▪ Began sketching out GUI – 50%
SpartanTrack

• Server Systems / Software
  ▪ MySQL is used to transfer app data from clients to server
  ▪ Using SpartanNash proprietary API’s for security

• Development Systems / Software
  ▪ React Native for dual development (Android + IOS)
  ▪ HTML/CSS/JAVASCRIPT/PHP (Web)
Team SpartanNash

Status Report

SpartanTrack

• Client Contact
  ▪ Traveled to SpartanNash HQ and met with IT leaders
  ▪ Weekly meetings - Tuesday and Friday @ 10:00 a.m.

• Team Meetings
  ▪ Team meetings – Tuesday @ 11:30 a.m., Friday @ 1:00 p.m.
  ▪ Triage meetings Friday @ 11:40 a.m.

• Team Organization
  ▪ Pair Programming
  ▪ Application Interface – Aleks + Denis
  ▪ Web Development – Tianyi + Abbott
  ▪ Data Transfer – Antonino + Abbott
Team SpartanNash

Status Report

SpartanTrack

Risks

• Scalable connection with SpartanNash DB
  ▪ May not perform well in real time
  ▪ Lots of testing, good planning

• Gold-platting
  ▪ Adding too many features may make the app difficult to use
  ▪ Constantly check with the client to make sure the features implemented are needed

• Client satisfaction with User Interface
  ▪ Interface may not be acceptable to user base
  ▪ Redesign the interface for simplicity
Team Spectrum Health

Status Report

Spectrum GO

• Project Overview
  ▪ Applications for hospital visitors to navigate
  ▪ Web interface for staff to configure paths
  ▪ Real-Time use of waypoints

• Project Plan Document
  ▪ Outlined
  ▪ Rudimentary Functional and Design Specifications Complete
  ▪ 15% complete
Spectrum GO

- **Server Systems / Software**
  - Microsoft SQL Server – TBD Friday
  - Identity Server – TBD Friday
  - GitHub/Repository – TBD Friday

- **Development Systems / Software**
  - Xcode – Up and Running
  - Android Studio – Up and Running
  - PHP Storm – Awaiting Client Approval
Team Spectrum Health

Status Report

Spectrum GO

• Client Contact
  ▪ Spoken with client once, weekly conference call time to come
  ▪ Site meeting scheduled for Friday, 01/19

• Team Meetings
  ▪ Team has met 4 times
  ▪ Weekly meetings scheduled two/three times a week, as needed

• Team Organization
  ▪ Mobile & Web Development
  ▪ Database Management
  ▪ Customer Liaison
Spectrum GO

Risks

• Spectrum Health Repository
  ▪ Obtaining sample code and repository from Spectrum on Friday
  ▪ Familiarize with the code as quickly as possible

• Route Data
  ▪ Can route data be compact enough for a reasonable phone download?
  ▪ If not, create options for downloading specific site data

• Real-time OCR
  ▪ Using Google Optical Character Recognition
  ▪ Follow tutorials and read documentation on the API

• Managing Waypoint and Routes
  ▪ How to add/delete waypoints and update routes
  ▪ Get everyone familiar with databases (SQL server)
Team Symantec

Status Report

Detecting Security Threats from User Authentication Patterns

• Project Overview
  § Build an application for analyzing VIP login data
  § Use the data to detect security threats in near real-time
  § Make dashboards to visualize the login data

• Project Plan Document
  § We started working on the outline
  § Table of contents is laid out
  § Started dividing up the work and discussing it at a meeting
  § 10% is complete
Team Symantec

Status Report

Detecting Security Threats from User Authentication Patterns

• Server Systems / Software
  ▪ Amazon Web Services: Need to get it up and running
  ▪ VIP Reporting Service Client: Waiting to get it from the client

• Development Systems / Software
  ▪ Splunk: Installed on iMacs and became familiar with it
  ▪ Elastic Search, Logstash, Kibana (ELK): Learning stack, doing basic tutorials
Detecting Security Threats from User Authentication Patterns

• Client Contact
  ▪ Had a conference call with client and met with local contact
  ▪ Weekly conference call scheduled Wednesday at 5:00 PM

• Team Meetings
  ▪ Our team has met 5 times
  ▪ Weekly team meetings Tuesday/Thursday at 4:20

• Team Organization
  ▪ Assign 2 people for Splunk and 3 people for ELK/AWS
  ▪ Re-distribute responsibilities halfway for pattern recognition and data analytics
Team Symantec

Status Report

Detecting Security Threats from User Authentication Patterns

Risks

• Ability to Detect suspicious patterns
  ▪ There is a wide range of threats to detect and want to avoid false flags
  ▪ Consult with experienced security advisor and identify possible threats

• Test Data
  ▪ Real VIP data is necessary to identify accurate threat patterns
  ▪ Get MSU’s VIP data

• Consistency between Splunk and ELK
  ▪ Making sure that functionality is consistent between both platforms
  ▪ Develop both applications concurrently

• AWS Servers
  ▪ The possibility of deploying the ELK applications on the AWS server
  ▪ Use AWS documentation and use online resources
Team TechSmith

Status Report

Snagit and Camtasia Output Extensibility

• Project Overview
  ▪ Simplify Sharing of Media Produced by TechSmith Products
  ▪ Extend Output Capabilities of Snagit and Camtasia
    o TechSmith Video Review
    o Wistia
    o Student Choice: Imgur

• Project Plan Document
  ▪ Outline complete
  ▪ Overall document is ~10% written
Team TechSmith
Status Report

Snagit and Camtasia Output Extensibility

- Development Systems / Software
  - Windows 10 virtual machine set up in Capstone lab
    - Installed and tested Visual Studio 2017 with C# .NET
  - Access to relevant TechSmith GitHub repositories established, along with communication through Flowdock and Slack
  - Currently examining documentation of various APIs needed to accomplish our output extensibility features
Team TechSmith

Status Report

Snagit and Camtasia Output Extensibility

• Client Contact
  ▪ Met with client in-person Friday 1/12 (Free Lunch!)
  ▪ Weekly Google Hangouts call on Fridays at 1 PM

• Team Meetings
  ▪ Team has met 3 times excluding All-Hands Meetings
  ▪ Weekly meetings on Mondays at noon

• Team Organization
  ▪ Quality Assurance: Logan Arent
  ▪ Client Liaison: Carter Chamberlain
  ▪ Technical Lead: Collin Dillinger
  ▪ Project Manager: Ryan Schiller
Team TechSmith

Status Report

Snagit and Camtasia Output Extensibility

Risks

• Risk 1: Reduced Team Size
  ▪ Description: Our team started with one fewer member than was originally intended
  ▪ Mitigation: More rigid team organization and adherence to deadlines

• Risk 2: API Credential Management
  ▪ Description: Our team requires APIs from three different applications
  ▪ Mitigation: Coordination with TechSmith and establishing a timeline to have all credentials in place

• Risk 3: API Uniformity
  ▪ Description: APIs used in this project may not present information uniformly
  ▪ Mitigation: Use abstraction provided by the TechSmith Extensibility Framework

• Risk 4: UI Design
  ▪ Description: GUI design is not a major skillset of our team
  ▪ Mitigation: Using WPF will reduce the difficulty of making a unified design for our plugin interfaces
Team Union Pacific

Status Report

“ALEXA – what’s my work schedule look like?”

• Project Overview
  - Trainmen, Yardmen and Enginemen (TY&E) employees operate trains for Union Pacific.
  - TY&E employees are on-call 24/7 and have constantly changing schedules.
  - Schedules are currently viewable online or in mobile app.
  - Integrate employees schedules’ into voice assistants such as Amazon Alexa, Google Home, or Siri.

• Project Plan Document
  - Skeleton created and uploaded to Google Team Drive.
  - Early database schema plans created.
  - Different use cases discussed but not added yet.
  - Next team meeting plan to divide sections to each member.
Team Union Pacific

Status Report

“ALEXA – what’s my work schedule look like?”

• Server Systems / Software
  ▪ Server assigned and early set up began, not finished.
  ▪ MySQL downloaded but not installed.
  ▪ Early database schema created.

• Development Systems / Software
  ▪ X-Code downloaded on Mac to begin iOS development.
  ▪ Alexa Skills Kit development online.
  ▪ Windows10 VM installed if needed.
“ALEXA – what’s my work schedule look like?”

- **Team Meetings**
  - January 10th 6PM-7PM
    - First meeting with teammates
    - Introduce each other and share each person’s schedule
    - Talked about project briefly
  - January 11th 2PM-3PM
    - Overview about Client meeting
    - Talked about how we can approach the project
  - January 16th 1:30PM-2PM & 4PM-5PM
    - Ready for Client meeting
    - Shared technical idea for project
    - Re-summarized Client meeting
    - Planned for each week detail schedule

- **Client Contact**
  - January 11th 1PM-2PM
    - First Client meeting with conference call
    - Overview for project tasks
    - Talked about Client request detail
  - January 16th 2PM-3PM
    - Second meeting with Client
    - Talked about the plan for project
    - Talked about risk and difficulties
    - Shared UX mock up design
    - Talked brief schedule for each week

- **Team Organization**
  - Client contact: Jared McMillan
  - Scheduling: M Kim
  - IOS & Siri & Alexa: M Kim, Daniel Agbay, Austin McGee
  - Database & Server: Jared McMillan, Daniel Agbay, David Hubble
Risks

• Risk 1 : Verification
  ▪ We will be handling sensitive information to the company and want to make sure this data is secured and not a security threat.
  ▪ We are exploring authentication channels through Google and Apple to minimize this risk.

• Risk 2 : Scalability
  ▪ Worried that a database focused information flow for the app will be difficult to scale to many users.
  ▪ Attempting to minimize database communication by storing preferences locally on app and requiring verification only once.

• Risk 3 : Schedule Format
  ▪ Union Pacific has many child companies that will use this app and each may have a different schedule format.
  ▪ Working with client to develop a standardized schedule format such as CSV or XML.

• Risk 4 : Assistant Development
  ▪ Unsure how to develop verification and settings options on voice assistants that do not have a mobile app such as Alexa and Google Home.
  ▪ Looking through Alexa documentation and working to set up very basic Alexa function – a Hello World program. Also exploring other Alexa apps to see how their verification systems work.
Team Urban Science

Status Report

Mobile Maestro

• Project Overview
  ▪ Control Maestro Exoskeletal Arm
  ▪ Use Mobile App with Voice Input
  ▪ Auto-Leveler
  ▪ Additional Safety Features

• Project Plan Document
  ▪ Outline created
  ▪ Question list for client in progress
  ▪ 20% complete
Mobile Maestro

• Server Systems / Software
  ▪ Azure server
  ▪ .NET API Endpoint
  ▪ SQL Server

• Development Systems / Software
  ▪ Cordova / Ionic setup
  ▪ SDK’s installed
  ▪ GIT Setup
  ▪ VM Setup
Team Urban Science

Status Report

Mobile Maestro

• Client Contact
  ▪ In-Person meeting
  ▪ Weekly meetings scheduled

• Team Meetings
  ▪ Multiple held so far
  ▪ Bi-Weekly meetings scheduled

• Team Organization
  ▪ Rotating project manager
  ▪ Roles assigned
Team Urban Science

Status Report

Mobile Maestro

Risks

• Bluetooth
  ▪ Setting up BLE connection to Arms
  ▪ Cordova BLE plugins

• Auto-Balancing
  ▪ Automatically balancing the Arms when on an incline
  ▪ Using phones Accelerometer / Gyroscope to level the system

• Voice activation
  ▪ Using vocal commands to control arms at all times
  ▪ Using Siri and Google Assistant

• User Experience
  ▪ UI needs to be simple and accessible
  ▪ Follow accessibility guidelines
Team USAA

Status Report

LIMElight: Life Insurance Made Easy

• Project Overview
  ▪ Generate accurate life insurance quote using machine learning
  ▪ Improve experience of receiving a insurance quote by creating a responsive mobile-friendly web application
  ▪ Utilize Ethereum blockchain to maintain and secure health records

• Project Plan Document
  ▪ System architecture diagram mockup is complete
  ▪ Screen mockups drafted
Team USAA

Status Report

LIMElight: Life Insurance Made Easy

• Server Systems / Software
  ▪ Deployed an Ethereum blockchain consortium on a Microsoft Azure server
  ▪ Deployed additional Azure server to host web application
  ▪ Hosting “Hello, world” applications for testing

• Development Systems / Software
  ▪ Installed Homebrew for package management
  ▪ Set up Windows virtual machines
  ▪ Configured Gitlab, Dropbox, Trello, and Slack services
  ▪ Installed Anaconda Python distribution
Team USAA

Status Report

LIMElight: Life Insurance Made Easy

• Client Contact
  ▪ Conference call project kickoff (introductions, project overview, expectations)
  ▪ Scheduled recurring weekly conference call Fridays at 4:00PM EST

• Team Meetings
  ▪ The team has met four times thus far
  ▪ USAA employees will be visiting from San Antonio at least once during the semester

• Team Organization
  ▪ Mike: Machine learning
  ▪ Xingchi, Dong, Nate: Web application (front and back-end)
  ▪ Abe: Project manager, utility player
LIMElight: Life Insurance Made Easy

Risks

• Blockchain Implementation
  ▪ Inexperience with the technology, difficult to see its use case
  ▪ Enrolled in Ethereum Udemy course, discussing relevance with client

• Accurate Life Insurance Quotes
  ▪ Model may struggle to produce an accurate life insurance quote with minimal applicant input
  ▪ Review academic research about most significant factors affecting an applicant's riskiness

• Poor Dataset
  ▪ Possibility of too few samples, inaccurate metrics, and irrelevant features
  ▪ Utilize third party datasets and generate our own data (e.g. location-based)

• Lack of Subject Matter Expertise
  ▪ Group does not have any experience with life insurance industry
  ▪ Connect with underwriters and actuaries at USAA
Sentiment and Emotional Analysis of Video Interviews

• Project Overview
  ▪ Build web app to facilitate recording and playback of pre-recorded and live video interviews
  ▪ Sentiment Analysis and Emotion Detection on audio/video
  ▪ Storing and querying of video interviews and sentiment/emotion results

• Project Plan Document
  ▪ Drafted cover page and table of contents
  ▪ Functional specs and mockup in progress
  ▪ Approximately 10% complete
**Team Yello**

**Status Report**

**Sentiment and Emotional Analysis of Video Interviews**

- **Server Systems / Software**
  - Heroku Server
  - Ruby on Rails
  - PostgreSQL

- **Development Systems / Software**
  - Scikit for sentiment analysis
  - Azure Emotion API for emotion detection
  - GitLab set-up
Team Yello

Status Report

Sentiment and Emotional Analysis of Video Interviews

• Client Contact
  ▪ Slack chat, e-mail, Google Hangouts
  ▪ Weekly conference calls on Fridays, 11:00 a.m.

• Team Meetings
  ▪ 4 meetings thus far
  ▪ Weekly meetings on Tuesdays, 2:00 p.m.

• Team Organization
  ▪ Trello for task organization
  ▪ Slack for quick communication
  ▪ GitLab for code collaboration
Team Yello

Status Report

Sentiment and Emotional Analysis of Video Interviews

Risks

• Risk 1
  ▪ Functionality and integration of APIs with Ruby on Rails
  ▪ Use our own input videos on the APIs

• Risk 2
  ▪ Managing Candidate vs. Staff privileges and views
  ▪ Design user model to identify user status

• Risk 3
  ▪ Capturing live video
  ▪ Inquire into TechSmith’s similar 2016 capstone project and contact team members for advice

• Risk 4
  ▪ Transcribing audio for sentiment analysis
  ▪ Research speech recognition methods/ APIs and use our sample input