MICHIGAN STATE UNIVERSITY

01/25: Team Status Reports

The Capstone Experience

Dr. Wayne Dyksen

Department of Computer Science and Engineering
Michigan State University

Spring 2016

From Students…
…to Professionals

Team Amazon

Status Report

(1 of 4)

Automated Laundry Manager

- Project Description
  - Minimize user's time input (one mass load and unload)
  - Use AWS IoT to communicate between smart machines
  - Application for user input and customization
  - Use machine learning to optimize system schedule
- Project Plan Document
  - Executive Summary and Functional Specs written.
  - Android and iOS apps 25% done and basic UI's created.
  - GPS functional and location services working.
  - Document is 33% done.

Team Amazon

Status Report (2 of 4)

Automated Laundry Manager

- Server Systems / Software
  - Supplied server running Ubuntu 12.04LTS
  - Usage as a git server, and a staging point for software
  - AWS account provided by client.
- Development Systems / Software
  - XCode & Android Studio for mobile app development.
  - Amazon Web Services (AWS) for communicating between devices and doing machine learning.
  - Raspberry Pi 2 Model B ordered for simulating IoT smart devices.

Team Amazon

Status Report (3 of 4)

Automated Laundry Manager

- Client Contact
  - Weekly conference call: Thursdays at 4:00pm
  - In-person meetings to take place at presentations and as needed
- Team Meetings
  - After triage meetings, Monday 6:00pm
  - Before and after weekly conference calls, Thursdays 3:00pm
- Team Organization
  - 2 people working on mobile apps.
  - 2 people working on AWS, Raspberry Pi, 1 on Server

Team Amazon

Status Report (4 of 4)

Automated Laundry Manager

Risks

- Risk 1
  - Various obstacles in achieving MQTT communication
  - Mitigation: Prioritize MQTT communication as high difficulty and importance
- Risk 2
  - AWS IoT - no prior experience and lots of content to learn
  - Mitigation: Use Amazon online tutorials and help each other learn
- Risk 3
  - No prior experience developing for Raspberry Pi/Embedded C
  - Mitigation: Research both topics, learn as we go
- Risk 4
  - Storing code versions on local server
  - Mitigation: Daily pulls, centralized backup

Team Auto-Owners

Status Report (1 of 4)

Catastrophic Claims Unit Mobilization

- Project Description
  - Catastrophic claim web page and mobile app
  - Predicting possible claims using NOAA weather information
  - Helping to allocate resources upon statistical data
  - Functionable mobile app with no service
- Project Plan Document
  - Project Plan has been started
  - Presented mock up designs to Auto-Owners for input
  - 40% completed
  - 5 pages written

Team Auto-Owners

Status Report (2 of 4)

Catastrophic Claims Unit Mobilization

- Server Systems / Software
  - Windows Server 2012 R2; Installed, running and tested
  - MySQL, PHPmyAdmin and Apache 2.4; Installed
  - Glassfish; Potentially later on
- Development Systems / Software
  - PHPstorm, Netbeans, XCode; Installed and running
  - VM, Windows 10, Sublime, PHP, Office
  - Trello; Web registered and started

Team Auto-Owners

Status Report (3 of 4)

Catastrophic Claims Unit Mobilization

- Client Contact
  - 1 formal meeting in person with with clients
  - Scheduled weekly conference calls Monday's at 2pm
    Starting January 25th, 2016
- Team Meetings
  - 5 team meetings
  - Weekly meetings on Wednesday 2pm
- Team Organization
  - Main contact person – Nicole Lawrence
  - Jason, Nick, and Matt – Tech Support

Team Auto-Owners

Status Report (4 of 4)

Catastrophic Claims Unit Mobilization

Risks

- Switching PHP to Java
  - Experience in PHP webpages. Limited knowledge on Java
  - Formatting PHP to communicate with Auto-Owners server
- Map overlays from NOAA onto Google Maps
  - Finding and pulling current NOAA maps and merging to google maps
  - Research overlays and map synchronization
- iOS swift fundamentals
  - No knowledge base on development
  - Research iOS development
- Synching database to mobile app and able to run offline
  - Creating a mobile app that can run without cell service
  - Researching self contained databases

Team GE

Status Report (1 of 4)

Cloud Management Portal

- Project Description
  - Cloud Asset Management Portal
  - Exposing Basic CRUD features
  - Users will be non-technical
- Project Plan Document
  - Functional Spec(90%) - drafted, awaiting client review
  - Design Spec(65%) - First interface prototype and use cases submitted for client review. Data Flow and Organization need to be written
  - Technical Spec(15%) – Section scaffolded and required technologies researched

Team GE

Status Report

(2 of 4)

Cloud Management Portal

- Cloud Systems / Software
  - AWS cloud instances tested and live
  - Scalr tested on VM, working on getting it live on AWS
  - Scalr API tested, started development of a wrapper
- Development Systems / Software
  - Lab Macs configured with VMWare and MSOffice
  - Private Github repo established and in use
  - Personal laptops configured with Git and AWS connections tested

Team GE

Status Report (3 of 4)

Cloud Management Portal

- Client Contact
  - Met with GE onsite 1/15 for kickoff
  - Met with GE UX on 1/19 to discuss usage of UX framework and design resources available to the team
  - Met with primary point of contact 1/21 arranged weekly agile standup style meetings on Mondays and Wednesdays
- Team Meetings
  - Weekly in-person meetings on Wednesday
  - Tracking and assigning tasks using Trello
  - Team and Primary GE contact using GroupMe for day to day questions, status updates, and decision making
- Team Organization
  - Nick - Project Management + Client Contact
  - Vincent and Lyle – Scalr and AWS configuration
  - Will and Aaron – Scalr API Wrapper and data processing

Team GE

Status Report (4 of 4)

Cloud Management Portal

Risks

- Risk 1
  - Mismatch of Scalr API's
  - Work closely with our GE contacts to ensure feature parity
- Risk 2
  - Lack of Access to GE's network
  - Ensure our environment closely resembles the enterprise environment
- Risk 3
  - Constraints of the project are very loose
  - Prototype and request feedback early and often
- Risk 4
  - GE Login integration will be difficult/impossible outside of GE's network
  - Implement OpenID Connect the protocol GE's internal login is based on

Team GM

Status Report (1 of 4)

IT Advocate Live Help

- Project Description
  - Connect GM employees together to resolve IT issues
  - Live chat technologies (Skype for Business)
  - Full web app and additions to the existing iOS app (GSDM)
  - Ranking and expertise system for best matchmaking results
- Project Plan Document
  - Outline, overviews, and risks started
  - Mockups in second phase (first revision from original)
  - Specifications not started
  - About 25% complete

Team GM

Status Report (2 of 4)

IT Advocate Live Help

- Server Systems / Software
  - Windows Server 2012 – Installed and running
  - Microsoft SQL Server 2012 – Not yet in use
  - Microsoft Exchange Server 2013 – Not yet in use
- Development Systems / Software
  - OS X / Xcode – Tested and in use
  - PHPStorm – Tested and in use
  - InVision – Tested and in use

Team GM

Status Report (3 of 4)

IT Advocate Live Help

- Client Contact
  - Weekly meetings Mondays before class (3 times thus far)
  - Onsite visit scheduled for April 1, 2016
- Team Meetings
  - Met at least 5 times so far
  - Scheduled before and after class, Mondays & Wednesdays
- Team Organization
  - Communication and scheduling via Slack
  - Project repo via GitHub, Documents on Google Docs

Team GM

Status Report (4 of 4)

IT Advocate Live Help

Risks

- Integration into Previous App
  - Ease of integration with the previous team's code base
  - Started testing now
- Difficulty with Exchange Server
  - Connecting the server to the database and the app
  - Researching uses and currently building mock account system
- Scalability
  - Concerns regarding number of users accessing database
  - Researching and using best practices and asking for client opinion
- Skype API
  - Integrating API within the app
  - Researched and built test app utilizing Skype URI

Team MSUFCU

Status Report

(1 of 4)

Money Smash Chronicle

- Project Description
  - Educate customers about financial information
  - Appeal to wide range of ages with an online game
  - Model after Candy Crush; easy-to-learn but difficult-to-master
  - Android, iOS and Web versions with cross-platform features
- Project Plan Document
  - Project Plan started
  - 25% done
  - Skeleton and some mockups completed
  - Still need to check many design details with client before further progress

Team MSUFCU

Status Report

(2 of 4)

Money Smash Chronicle

- Server Systems / Software
  - Server set up and running
  - Ubuntu Server 14
  - SQL, PHP, nginX up and running
- Development Systems / Software
  - Github organization containing three different repositories
  - Xcode, Android Studio, PhpStorm
  - Hello World apps running for all platforms

Team MSUFCU

Status Report

(3 of 4)

Money Smash Chronicle

- Client Contact
  - First conference call Friday the 15th
  - Weekly conference calls every Friday at 3:30 PM
- Team Meetings
  - Scheduled meeting every Monday, Wednesday and Friday. Meet more often when needed.
  - Have met six times so far; also daily group messaging
- Team Organization
  - Brandon: Client Contact & Web. Bobby: iOS & Web.
  - Cory: iOS. Wyatt: Android. Amy: Android & Design.

Team MSUFCU

Status Report (4 of 4)

Money Smash Chronicle

Risks

- Gameplay Issues
  - Many gameplay aspects requested by client are incompatible with Candy Crush or are otherwise difficult to implement
  - Need to make a lot of creative game design decisions under guidance of client
- No iOS or WebGL experience
  - None of the team members have developed on iOS or with WebGL before
  - Use tutorials, references and peers for help
- Resource Issues
  - Resource-intensive project and no resources yet; no experienced designers
  - Work with client on design ideas
- Cross-platform and cross-version issues
  - Potential difficulty in making the game look and feel similar across the three platforms and the different versions of each platform
  - Need to work together closely and test thoroughly and frequently

Team Quicken Loans

Status Report

(1 of 4)

Game of Loans

- Project Description
  - Educate users on the loan process
  - Gamification of the loan process
  - Web-based single-player game
  - Use Quicken Loans properties to tie in the brand
- Project Plan Document
  - Executive Summary written
  - Design and Technical specifications not yet formalized
  - Creating testing plan and estimating schedule
  - ~20% completed

Team Quicken Loans

Status Report (2 of 4)

Game of Loans

- Server Systems / Software
  - Windows 2012 server/IIS installed/ test Website running
  - SQL Express 2014 installed and tested
  - Unity Asset server for version control will be installed upon receiving licenses
- Development Systems / Software
  - Unity 5.3/Installed on machines, pending licenses
  - MonoDevelop IDE installed on machines

Team Quicken Loans

Status Report

(3 of 4)

Game of Loans

- Client Contact
  - First contact 1/15/16 (initial meeting)
  - Weekly Meetings set Fridays 9:30am
  - Onsite meeting scheduled for 2-12-2015
- Team Meetings
  - Mondays and Wednesdays after class
  - Fridays after the client meeting
- Team Organization
  - Google drive for shared documents
  - Mobile messaging app installed on each member's mobile phone

Team Quicken Loans

Status Report (4 of 4)

Game of Loans

Risks

- Risk 1
  - Only one team member is familiar with Unity development.
  - There is documentation and tutorials online available for reference.
- Risk 2
  - No experience with setting up servers.
  - We will look up documentation and tutorials online for reference.
- Risk 3
  - The client is vague on how the overall game should be structured.
  - We will continuously iterate our game to ensure quality
- Risk 4
  - We do not know the skills of each team member.
  - Clear communication , especially when struggling

Team Spectrum Health

Status Report (1 of 4)

Mobile Rounding Application

- Project Description
  - Provide means for surgeon to connect with patients
  - Allow patients to view their personal care plan
  - Provide patients with their timeline to be discharged
- Project Plan Document
  - Started on 1/20/2016
  - 30 percent complete
  - Outline, Table of Contents, Project Overview and Risks sections completed

Team Spectrum Health

Status Report (2 of 4)

Mobile Rounding Application

- Server Systems / Software
  - Server set up with Windows Server 2012
  - IIS (Internet Information Services) running/ready for app deployment
- Development Systems / Software
  - Visual Studio 2015 running current web prototype
  - ASP.NET 5 MVC / C#
  - RESTful Web Services

Team Spectrum Health

Status Report (3 of 4)

Mobile Rounding Application

- Client Contact – Markus Neuhoff
  - Weekly Conference Call scheduled every Friday at 1 PM
  - In person meeting to be scheduled soon
- Team Meetings
  - Weekly Meetings scheduled Monday/Wednesday after All Hands
  - We have met as a team 8 times, including conference calls and triage meetings
- Team Organization
  - We have a team calendar set up with all meeting information
  - We have a Github repository set up to host our project

Team Spectrum Health

Status Report (4 of 4)

Mobile Rounding Application

Risks

- Secure Provider/Patient Messaging
  - Properly securing all personal patient/confidential records
  - Work directly with Spectrum to comply with patient confidentiality regulations
- Vidyo Messaging API
  - Implementing preferred messaging API directly into mobile application
  - Consult Spectrum developers with knowledge of the API
- Overcommitting on Features
  - Not overdoing our workload to provide deliverables on-time
  - Prioritize feature list and commit to completing most important features first
- Learning ASP.NET MVC, Frontend/Backend Software
  - Using current experience to develop with software never used before
  - Assign roles to all team members

Team TechSmith

Status Report

(1 of 4)

Cloud Based Video Face Tracking

- Project Description
  - Automated face tracking within uploaded videos
  - Blurring or highlighting of individual faces based on user input
  - Cloud based storage of videos
  - User login and continuation of editing user uploaded videos
- Project Plan Document
  - Full Outline, Mock-Ups
  - Overview & Risks
  - Technical Structure Outlined
  - ~15%

Team TechSmith

Status Report

(2 of 4)

Cloud Based Video Face Tracking

- Server Systems / Software
  - Set up server and installed program (won't be using, though)
  - Azure - waiting on full access, began testing
  - Active Directory - waiting on full access
- Development Systems / Software
  - Visual Studio, C# - downloaded, created test applications
  - Face Tracking API - chose, integrated with a C# project
  - Video Editing - Testing APIs and libraries

Team TechSmith

Status Report

(3 of 4)

Cloud Based Video Face Tracking

- Client Contact
  - Scheduled weekly in-person meetings for Fridays
  - Have met twice in person; Consistent update emails
- Team Meetings
  - Scheduled bi-weekly meeting
  - Have met three times
- Team Organization
  - Client Contact and UI teammates defined
  - Have designated project features per team member

Team TechSmith

Status Report (4 of 4)

Cloud Based Video Face Tracking

Risks

- Integrating the face tracker API
  - Learn how to integrate the most appropriate API into a C# application (high priority, medium difficulty)
  - Consulted TechSmith, gone over tutorials, hard deadline for prototype
- Video editing within the application
  - Learn how to change properties within a video and refactor with changes (high priority, high difficulty)
  - Using tutorials, part of prototype with hard deadline
- Identity-Specific Face Tracking
  - Figure out a way to track a specific face throughout a video for filtering (medium priority, high difficulty)
  - Trying to use API to accomplish, part of second prototype with hard deadline
- Design of the application
  - Create a user friendly way to navigate the application, learn more front-end development techniques (medium priority, medium difficulty)
  - Made mock-ups (adjust with functionality), assigned team member

Team Union Pacific

Status Report (1 of 4)

Oculus Rift Inspection and Training Tool

- Project Description
  - Inspect Virtual 3D objects using the Oculus Rift
  - Free-form Viewing
    - Explode/Reassemble components, object manipulation
  - Guided lessons
  - Objective based quiz
- Project Plan Document
  - Functional specification ~75% complete
  - Design specification ~40% complete
  - Technical specification ~10% complete
  - Schedule/Risks ~90% complete

Team Union Pacific

Status Report (2 of 4)

Oculus Rift Inspection and Training Tool

- Server Systems / Software
  - Self-contained system, no server necessary
  - Models loaded on local machine
- Development Systems / Software
  - Unity 5.3 Professional
  - Oculus Rift DK1
  - Version control via private GitHub repo
  - All development systems are working correctly

Team Union Pacific

Status Report

(3 of 4)

Oculus Rift Inspection and Training Tool

- Client Contact
  - Weekly conference calls scheduled for Tuesdays
  - Face-to-Face meetings scheduled as needed
- Team Meetings
  - Polled availability of group members
  - Weekly meetings on Monday
- Team Organization
  - Grant King assigned to client liaison position
  - Development tasks assigned based on availability and strengths

Team Union Pacific

Status Report (4 of 4)

Oculus Rift Inspection and Training Tool

Risks

- Configuring Oculus to Integrate with Unity
  - Integration can be a complex procedure prone to errors
  - Mitigation: Client has experience and resources working with Oculus Rift
- Making Intuitive UI/UX for Oculus Rift
  - Standard Unity GUI is disorienting for the user
  - Mitigation: Perform adequate field testing
- Scalability With Exploded View of Complex Models
  - Nested Levels of intricate subcomponents can cause performance issues
  - Mitigation: Use appropriate algorithms and data structures
- Feature Creep
  - Many opportunities for extra features, need to focus on core goals
  - Mitigation: Compromise with client on quality of features vs quantity

Team Urban Science

Status Report (1 of 4)

Dealership Inventory Optimization System

- Project Description
  - Assist dealerships with inventory purchase recommendations
  - Enable dealerships to optimize their purchasing power
  - Aid customers in finding a vehicle they're looking for
  - Allow customers to recommend vehicles to dealerships
- Project Plan Document
  - 1/15 – Created project plan skeleton (outlined every section)
  - 1/22 – 50% completed (Use cases/tech. specs/testing/risk remain)
  - Currently 22 page rough draft
  - Completed first draft of mock ups for web and mobile app

Team Urban Science

Status Report (2 of 4)

Dealership Inventory Optimization System

- Server Systems / Software
  - Client's server up and running, waiting for login info
  - SQL Server 2012 installed on client's server
  - Ubuntu installed, Apache web server up and running
- Development Systems / Software
  - Bootstrap, AngularJS, and jQuery tested
  - Ionic framework installed and tested
  - PHPStorm, Xcode, and Android Studio installed and tested

Team Urban Science

Status Report (3 of 4)

Dealership Inventory Optimization System

- Client Contact
  - 1/13 - Conference call (weekly Wed. meetings scheduled)
  - 1/22 - Met with client at their Detroit headquarters
- Team Meetings
  - Full team has met 6 times
  - Scheduled weekly meetings Monday/Wednesday
- Team Organization
  - Web App: Tyler (Client Contact), Justin, Joey
  - Mobile App: Hannah, Anthony, Justin

Team Urban Science

Status Report (4 of 4)

Dealership Inventory Optimization System

Risks

- Risk 1
  - Unsure how to use SQL Server 2012
  - Utilize online documents and tutorials for self-learning
- Risk 2
  - Little experience with the Ionic framework
  - Consult the Ionic documentation and tutorials
- Risk 3
  - Incorporating unfamiliar libraries (Bootstrap and AngularJS)
  - Follow online documentation and tutorials while prototyping often
- Risk 4
  - User design has to be intuitive and simple
  - Frequent testing and reviews from our client

Team Whirlpool

Status Report (1 of 4)

Mobile Whirlpool Product Catalog

- Project Description
  - Turn existing product catalog (Web World) into a mobile app
  - Display product images and their specifications
  - Share products with customers through email and phone
  - Allow for comparison functionality between products
- Project Plan Document
  - Introduction, project overview, time table, layout completed
  - Wireframes/Application Mockup and Workflow completed
  - Use cases, in progress
  - Technical and Functionality Specifications, in progress

Team Whirlpool

Status Report (2 of 4)

Mobile Whirlpool Product Catalog

- Server Systems / Software
  - Whirlpool's existing servers. Up and running, and access granted
  - Ubuntu server in lab up and running. Basic hardening, Fail2Ban, SSH Keys.
  - MySQL server
- Development Systems / Software
  - iMacs set up with Android Studio and XCode
  - Adobe Creative Cloud for UX/UI/Design
  - GitHub Source Control

Team Whirlpool

Status Report (3 of 4)

Mobile Whirlpool Product Catalog

- Client Contact
  - Initial and follow up meetings, daily email correspondence
  - Set up whirlpool accounts, Google Developer Tools access
- Team Meetings
  - Wednesdays - Google Hangout with Whirlpool
  - All-team meetings scheduled Mondays, Thursdays to discuss concerns, assistance, past and upcoming week
- Team Organization
  - Slack, Google Calendar, Google Drive, GitHub, Email
  - Division of Team Roles based on skillset and interests

Team Whirlpool

Status Report (4 of 4)

Mobile Whirlpool Product Catalog

Risks

- Application latency
  - Many assets needed, application speed and responsiveness crucial
  - Image caching, storage of frequently utilized materials
- Mostly non-technical audience
  - Application used by vendors with no technical training, may have older devices
  - Develop for lower SDKs, emphasize UX for maximum functionality
- Recreation of Whirlpool Catalog from XML
  - Product info given to team as 1 million line XML document
  - Need to learn and implement XML parsing into a team database for use with the application
- iOS Development - Swift
  - Team has vastly more Android programming experience, will need iOS
  - Assigned Team Members Jordan and Can to be primary iOS focused developers

Team Yello

Status Report

(1 of 4)

Syncing Mobile Data Without Internet Connectivity

- Project Description
  - Sync at least five tablets with candidate information
  - Create two libraries – one for iOS, one for Android
  - Utilize Wifi-Direct, Mulitpeer, and/or Bluetooth technology
  - Easy to setup for non-technical users
  - Exclusive networks for different clients using the apps
- Project Plan Document
  - Created cover page and table of contents
  - Formed an executive summary and functional specifications
  - Screen mockups and flow chart

Team Yello

Status Report

(2 of 4)

Syncing Mobile Data Without Internet Connectivity

- Server Systems / Software
  - Github – repositories for both Android and iOS
- Development Systems / Software
  - Xcode, Android Studio
  - Swift, Java
  - SQL, JSON

Team Yello

Status Report

(3 of 4)

Syncing Mobile Data Without Internet Connectivity

- Client Contact
  - Talked with client
  - Established plan for weekly status updates via email
  - Google Calendar set up to coordinate with client
- Team Meetings
  - Met for setting up software/server
  - Met for status report
  - Constant communication each day through Slack and Google Hangouts
  - Several weekly meetings
- Team Organization
  - Plan on splitting workload into two separate development teams (iOS & Android)
  - Other team members are starting architecture and planning

Team Yello

Status Report (4 of 4)

Syncing Mobile Data Without Internet Connectivity

Risks

- Risk 1
  - Unsuccessfully establishing communication between the tablets
  - Mitigation: We will begin with WiFi Direct and simple data models, but will move on to other multipeer tech in case of failure
- Risk 2
  - Unfamiliar with iOS development – Swift coding language
  - Mitigation: practice, tutorials, swift documentation
- Risk 3
  - Being unable to deliver on client's preferred mode of P2P syncing
  - Mitigation: Start small and basic, last resort being developing a client/server architecture
- Risk 4
  - Interface not being user-friendly
  - Mitigation: Field tests, client tests

01/25: Announcements

- Check Website Team Photo Names and Hometowns
- Project Plan Document and Presentation
  - Presenting and Due Dates
  - Schedule Conflicts
  - Read READ ME
- Include Message Body When Sending Email
  - Be Professional
  - Avoid Spam Filters and Junk Folders
- Issues? Problems? Questions?

What's ahead? (1 of 2)

- All-Hands Meetings
  - ~~M, 01/11: Capstone Overview~~
  - ~~W, 01/13: Project Plan~~
  - ~~M, 01/18: (Martin Luther King Day, No Meeting)~~
  - ~~W, 01/20: Risks and Prototypes~~
  - ~~M, 01/25: Status Report Presentations~~
  - W, 01/27: Schedule and Teamwork
  - M, 02/01: Team Project Plan Presentations
  - W, 02/03: Team Project Plan Presentations
  - M, 02/08: Team Project Plan Presentations
  - W, 02/10: Team Project Plan Presentations

What's ahead? (2 of 2)

- Project Plan Presentations
  - PowerPoint Template
    - Download Now
    - Read the Read Me Slide (Over and Over and Over…)
  - Submission
    - Both Project Plan Document and PowerPoint Slide Deck
    - Due 4:00 a.m., Monday, February 1
    - See Submission Instructions in Template
  - Presenting
    - 3 Teams Per Meeting Over 4 Meetings
    - Schedule Posted Sunday Evening
    - Strict 15 Minute Time Limit
    - Use Team Member Laptop
      - Bring Power Cord
      - Test In Meeting Room (in Advance)
    - Rehearse
    - 5% of Final Grade
    - Business Casual Dress
  - Formal Team Photos
    - Immediately Following Meeting
    - In Capstone Lab
  - Schedule Conflicts
    - Only for Interview Trips
    - Notify Dr. D. Well In Advance