**Museum Experience Survey**

by:

Andrew Mueller

Edward Dinki

Jonathon Shippling

Robert Vrooman

Index

[Overview](#h.3mqvvhl11dym)

[Goals and Scope](#h.w08zv65ye140)

[In Scope](#h.i1zj3weygpbr)

[Out of Scope](#h.7i4euom56voy)

[Goals](#h.xote0wkz274d)

[Deliverables](#h.3dbho57pfsm3)

[Risk Management](#h.6b6bzg7nu6cr)

[Technical Process](#h.o3g7ulh9tumq)

[Scheduling and Estimates](#h.9r01p2k5yk3q)

[Measurement and Metrics](#h.hkwmn3m23qjx)

# Overview

Lockheed Martin has a number of volunteer members involved with The Discovery Center of the Southern Tier, a non-profit museum, where Lockheed Martin has sponsored the addition of a number of new exhibits highlighting engineering. In order to improve the Discovery Center, Lockheed Martin wishes to implement a system for tracking user engagement and feedback on the exhibits. The purpose of this project is to create that system.

The Museum Experience Survey will provide an electronic system for visitors of the museum to provide feedback and demographic statistics with as little manual entry as possible. The Museum Experience Survey will ask visitors basic demographic questions such as zip code, number of kids, whether or not it’s a first-time visit, as well as allow the visitor to rate and provide feedback on the exhibits. Volunteers working at the museum will be able to see the data and statistics received from the Museum Experience Survey and use that data to better the museum.

There are only two users of the system: visitors and admins. Visitors will be families coming to the museum to view the exhibits. It is yet to be decided how the visitors will interact with the system: the two leading ideas are that there will be tablets placed in kiosks at each exhibit, with a set of general feedback questions displayed for the user to fill out, as well as basic demographic information, or there will be one (or more) central tablet kiosks located at the front desk of the museum for users to fill out general feedback and demographic information for the entire museum. Admins will be volunteers at the museum who are technologically-proficient and have the responsibility to create new exhibit questionnaires and view data, statistics, graphs, and more information collected from the system.

Software development work will be done by Team MESSE starting the week of

November 3, 2014, and working through April of 2015. Team MESSE is responsible for all software portions of the project, and hardware will be provided by the Discovery Center and Lockheed Martin, including tablets, web servers, and networking hardware.

# Goals and Scope

The scope defines everything that the project will (in scope) and will not (out of scope) be. Scope is not a requirements listing; requirements will be stated in a separate document. The goals are not specific and instead will be given more precise definitions of success in the requirements document.

The main focus of the “museum survey” is to gather personal and demographic information in order to recruit volunteers. The secondary focus is to make a modifiable survey that can collect data such as exhibit ratings to determine which exhibits are liked most.

## In Scope

1. Web based application shall
   1. Be fully functional on Google Chrome.
   2. Be hosted on a Windows Server machine.
   3. Be restricted to local access only.
   4. Run on an android tablet.
   5. Collect data
      1. On demographic and personal information (Age, Gender, Email).
      2. Of ratings for each exhibit on a scale of 1-5 where 5 indicates a favorite exhibit.
      3. Through a page that doesn’t require authentication.
   6. Report data
      1. Of exhibit rankings in order from highest to lowest.
      2. Of average ranking for all exhibits.
      3. Of min and max rankings
      4. Through a page only accessible with admin privileges.
      5. By exporting
         1. As a csv dump
         2. As a jpeg
   7. Provide security against CSRF attacks and sql injections.
   8. Allow administrators to modify and add questions
   9. Allow administrators to edit the list of exhibits for visitors to rate
2. Installer shall install all necessary components for the web application server.

## Out of Scope

* Supporting cell phone access.
* RFID tag tracking for visitors.
* Limiting the number of concurrent users for the website

## Goals

Goals are listed in order of priority.

* Security - Keep museum visitor information secure, so that it isn’t accessed by unauthorized personnel.
* Usability - The application should be easy to use since it will be used by the general public and museum administrators who have little computer knowledge.

# Deliverables

RIT Software Engineering Department Deliverables

1. Project website holding all work products and project artifacts maintained in the project account on the se.rit.edu web server
2. Project plan, schedule, and process methodology definition prepared by the end of week 3 of the first term.
3. Tracking report for time/effort worked on the project, and at least two other product/process metrics appropriate to the project and development methodology. Tracking reports updated on the project website at least every two weeks.
4. Interim status and final project presentations
5. Project poster and presentation at “SE Senior Project Day”
6. Project technical report

Sponsor Deliverables

1. Team Information
2. Individual Informal Work Reports

# Risk Management

Please see the Risk Assessment document.

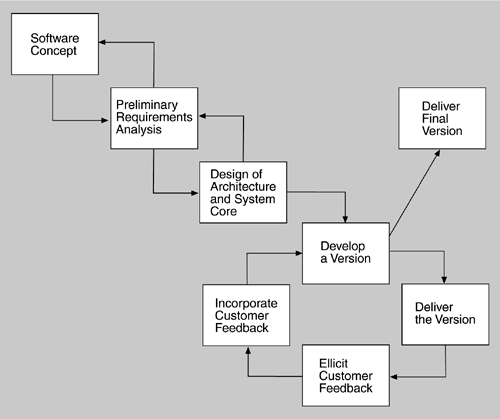
# Technical Process

The team has decided to go with an Evolutionary Delivery methodology for this project. This process works similar to traditional waterfall with upfront requirements analysis and architectural decisions, but breaks down the development phase into cycles which are similar to iterations. This allows us to get customer feedback during development so that we can incorporate their feedback as we go, instead of waiting until the end. This helps reduce risk and ensure that the final product is what the customer is looking for.

Requirements should be defined as concretely as possible during the requirements phase through communication with both the project sponsor and the end customer (the museum itself). During this time, any architecturally significant requirements should be identified. A better solution can be design when requirements are defined up front, and lowers the chances of misinterpretations that can cause major setbacks during development.

Though the customer has a pretty good understanding of what they want, there still is a pretty good chance that requirements could change slightly over time. In this case, visibility is critical so that actions can be taken early based on customer feedback. Due to the importance of visibility, the focus will be on vertical slices of the end solution (model, control, and view for part of the functionality). The requirements and architecture will be captured in a living document that we may choose to update as customer feedback is received.

Here is a graphical representation of the Evolutionary delivery model:



# Scheduling and Estimates

Current scheduling is subject to change with new requirements, so the schedule is loosely defined.

1. Project Plan End of Week 4
2. Requirements Document First Draft End of Week 5
3. Requirements Document Revised and Reviewed End of Week 6
4. Architecture Document End of Week 9
   1. Technology choices
   2. UML
   3. UI Mockups
5. Cycles Begin Week 11
   1. Gathering demographic/personal information
   2. Gathering exhibit rating information
   3. Display Data
   4. Export Data
   5. Testing
   6. Installing/ deployment

# Measurement and Metrics

Measurement and metrics will be broken down into two broad categories. These categories are aptly named maintainability and efficiency. Maintainability refers to metrics that are designed to capture information that will serve to increase the quality and lifespan of the product. Efficiency deals with the fact that most, if not all, users will be non technical. Each task the system performs must be efficient and easily understandable..

**Maintainability**

1. **Bug Fix Velocity-** An indirect measure of the program’s complexity may be determined by the Bug Fix Velocity. This is a measure of the time since a bug is proven and recorded in the tracker to the time the bug fixed or deferred. Each member of the team shall calculate his own Bug Fix Velocity for bugs assigned to him.
2. **Cyclomatic Complexity**- Cyclomatic Complexity measure the number of independent linear paths through the software to an endpoint for a similar tasks. An example of this would be branching paths for a user deciding “Yes” or “No” on a dialog. Increased Cyclomatic Complexity leads to user confusion and therefore must be measured and minimized accordingly.

**Efficiency**

1. **Time to completion -** Given a group of sample users of the system, indicate specific tasks they must perform. With no assistance from the observers, calculate the time it takes for the user to perform the operation. This may be performed several times over the course of a session to simulate increased knowledge of the user.
2. **Error Rate -** Calculated during a “Time to completion” session, record the number of mistakes a user makes while attempting to complete the task. Also, record any error messages and help screens the user employs to attempt to finish the task. In general, no fatal errors or mistakes should occur. The following are definitions of what will be recorded.
   1. Corrected Mistakes - Mistakes the user committed that they corrected themselves within 10 seconds.
   2. Mistakes - Mistakes the user committed that were only corrected after a single hint from the observer after the ten second mark. These hints will be recorded along with the mistake.
   3. Fatal Mistake - Mistakes that the user committed that remained uncorrected, causing the task to not get completed.
   4. Error - Any error messages brought up by the system or any incorrect behavior that does not include core functionality. Record what actions the user attempted to bring about the error.
   5. Fatal Error - A user action caused the system to malfunction. A fatal malfunction includes crashing, incorrect functionality of core behavior, or security breaches.
3. **Page Views/Clicks**- With efficiency in mind, the number of clicks and navigations to accomplish a task should be minimized. Also, like tasks should be grouped in pages and therefore decrease the number of pages that must be viewed. For each task, calculate number of clicks required and record each page viewed along the happy path (the shortest path to complete the task). Since this is a “happy path” only metric, this may be calculated without an actual user performing the task.