GIS Technical Program Manager - MapLarge Test Project

As a Technical Program Manager, you have the responsibility of overseeing the successful planning, execution and delivery of software projects for MapLarge customers. We are interested in your detailed approach to understanding, documenting, and communicating the entire project with coworkers and customers.

1. You and the development lead have met with the customer. The project details and various requirements have been communicated by the customer below in the Sample Developer Project. Please familiarize yourself with the sample project including goals, requirements and implementation suggestions.

2. Create a slide deck presentation of how you approach technical and managerial execution, as well as an annotated, graphical mock-up of the delivery along with a task break down of units of work that might go into JIRA.

   Slide Deck Presentation Details:

   - Illustrate your approach to eliciting more detailed requirements
   - Describe execution methodologies and tools, (e.g., Agile and Jira)
   - Illustrate workflows or communication strategies with developers, managers and the customer
     - Be able to characterize how you handle customer interactions
     - Demonstrate how you decompose the project into milestones and deliveries
     - Illustrate how you document the project for different audiences
   - Describe best approach to project risk management (schedule, technology decisions, ad-hoc requirements, etc.)
   - Describe how you mitigate risk and failure
   - Provide an illustrated and annotated mock-up of the software user interface based on the sample project requirements.

3. We invite you to implement an actual prototype as an optional addition to this exercise if you would like to showcase your coding skills.

Sample Developer Project: Single-Page Application for File & Directory Browsing

The task is to create a web service API that allows users to query the contents of a directory on the web server and a single page web app that can be used to search and browse folders and files. A server side home directory should be configurable via variable.

Client Side, the application should be deep-linkable (the state of the UI should be kept in the URL). All of the UI work should be done client side via JavaScript that renders HTML - do not render HTML server side (no

ASPX, etc). Styling is not as important for this task, we are more concerned with functionality. Keep framework usage to a minimum, but feel free to use Knockout, jQuery and jQuery UI if it helps.

The goal is to produce a functional proof-of-concept to let us see how you think. It is important to provide a decent body of code, so that we can talk about the logic than it is to make it “pretty”.

The point of this exercise is to end up with a body of work that is big enough that our team can review it and evaluate your design and coding style and skills.

Sample Functional Requirements
In this example we would want a developer to implement both server side and client side logic in a primitive end to end flow that runs without errors for a few small features. If you chose to submit a project, the goal is to show a working proof of concept with enough C# and JavaScript code that we can get a feel for your coding style and have some interesting classes to discuss in a code review. You may delete all the default controllers and files you are not using, as it may help us focus on your code

Primary Requirements

- Ensure your solution builds in Visual Studio 2017
- Web API that allows Browse and Search Files & Folders and returns JSON
- Deep linkable URL pattern
- SPA (Single Page App using JavaScript)
- Upload/download files from the browser
- Show file and folder counts and sizes for the current view
- No third party plugins. (Knockout, jQuery and jQueryUI are not requirements but can be used.)

Secondary Requirements

- Entire component contained in a dialog widget, with a trigger element (button, etc)
- Delete, Move, Copy files and folders
- Performance - we highly value performance
- Any other cool stuff you want to show!

Spend a few hours: Put your best foot forward, submit a package that shows your best work. The evaluation will be discussions of your implementation and architecture choices.

Download the empty project: Here. We’ve provided a blank slate that a developer would start with. I has an index.html file and an empty controller. If you choose to tackle creating a working implementation, you can zip it up and share it with the team for review either as a download link or zip attachment to an email. If you share the project by email keep in mind the zip has exe files inside it and Gmail and other virus scanners may block the email. To avoid issues, rename the extension to something like MyProject.CHANGEME