Software Requirements Specification

## Workflow Diagram Widget



## 

## 

## Version 0.2 - Initialization

## Prepared By: VortekSolutions

Last Modified: 21 March 2017

## 

**Table of Contents**

[**Workflow Diagram Widget**](#_d58nl74sq40y) **1**

[**Version 0.2 - Initialization**](#_1qil7wpxnkis) **1**

[**Prepared By: Vortek Solutions**](#_kpy4gpp36q98) **1**

[**Last Modified: 20 March 2017**](#_t51txy6ogeyq) **1**

[**Introduction**](#_1fob9te) **4**

[**Purpose**](#_3znysh7) **4**

[**Document Conventions**](#_2et92p0) **4**

[**Intended Audience and Reading Suggestions**](#_tyjcwt) **4**

[**Project Scope**](#_3dy6vkm) **4**

[**References**](#_1t3h5sf) **4**

[**Overall Description**](#_4d34og8) **5**

[**Product Perspective**](#_2s8eyo1) **5**

[**Product Functions**](#_17dp8vu) **5**

[**User Classes and Characteristics**](#_3rdcrjn) **8**

[**Manufacturing**](#_xu1kic2gcl1l) **8**

[**Developers**](#_srsdgyrkox0j) **8**

[**Operating Environment**](#_26in1rg) **8**

[**Design and Implementation Constraints**](#_lnxbz9) **8**

[**User Documentation**](#_35nkun2) **8**

[**Assumptions and Dependencies**](#_1ksv4uv) **9**

[**External Interface Requirements**](#_44sinio) **9**

[**User Interfaces**](#_2jxsxqh) **9**

[**Software Interfaces**](#_3j2qqm3) **10**

[**Communications Interfaces**](#_1y810tw) **10**

[**System Features**](#_4i7ojhp) **11**

[**Read JSON**](#_2xcytpi) **11**

[**Display a Basic Workflow**](#_321qzopziz38) **11**

[**Allow Customization to the Workflow**](#_elpxptuw8146) **12**

[**Other Nonfunctional Requirements**](#_3whwml4) **13**

[**Performance Requirements**](#_2bn6wsx) **13**

[**Safety Requirements**](#_qsh70q) **13**

[**Security Requirements**](#_3as4poj) **13**

[**Software Quality Attributes**](#_1pxezwc) **13**

[**Business Rules**](#_49x2ik5) **13**

[**Other Requirements**](#_2p2csry) **14**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Vortek Solutions | 03/12/2017 | Init | 0.1 |
| Justin Goulet | 03/20/2017 | Added more information to unfinished sections | 0.2 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to describe the requirements for the Workflow Diagram Widget. The scope of this project includes the creation of a workflow diagram using a provided node sample. Using just the name and ID, the widget should be customizable such that fields can be added “on the fly.” If there is time, after the completion of the workflow widget in JS, other modules such as a diagram layout configurator and a scheduling module will be created.

## Document Conventions

This document assumes general knowledge of GitHub and JS terminology and conventions. Some sections of this document include relations to branches, including version control, and mentions of cloud-hosting services that enable collaboration among developers.

This document uses conventions such as bolding and headers to emphasize points. Whilst not all details require its own section, there should be importance on their mention. This document includes hyperlinks, which enable the digital document to link with relative domains (including test and live environments).

## Intended Audience and Reading Suggestions

This document is intended for developers to get familiar with our product and be able to make any modifications or updates to our product as they see fit.

## Project Scope

The “Workflow Diagram Widget” is an browser embeddable visualization tool built with the intention of being able to visualize workflow diagrams and their relationships quickly and easily. The widget is being built to replace paper workflow charts that are currently being utilized which will increase visibility into potential bottlenecks and pain points during the manufacturing process.

The software is built with the intention that it will be able to visualize not just workflows but other node based relationships such as parent-child cost charts.

## References

[*Systems Requirements Specifications*](https://docs.google.com/a/cougars.csusm.edu/document/d/1FBvXaSeZIsp7oOMNgWWfGTHuAPhJZadgh3N1JNydCYs/edit?usp=sharing)

* Details summary of project with specific use cases, terminology and proposed project applications.

[*Communication Document*](https://github.com/jstngoulet/vorteksolutions/blob/master/Aditional%20Resources/VortekSolutions%20Communications.docx?raw=true)

* Demonstrates means for communications between team, client and instructor.

*Project Plan*

* Summary of how the project will be implemented. Will include references to MS Project

[*Roles*](https://github.com/jstngoulet/vorteksolutions/blob/master/Aditional%20Resources/VortekSolutions%20Roles.docx?raw=true)

* Summary of team member's abilities in addition to key roles they will play in this project. Descriptions are provided in relation to project requirements.

*Key Stakeholders*

* Describes specific stakeholders for this project, how they relate and where they can find information in detail

*Change-Log*

* In addition to standard Version Control Software (such as Git), we are going to record all changes in the Change-log document. You can see a breakdown of all items created, fixed, updated or removed in a sub-version sequence.

[*Team Resume*](https://github.com/jstngoulet/vorteksolutions/blob/master/Aditional%20Resources/VortekSolutions%20Team%20Doc.docx?raw=true)

* Brief breakdown of specialties

[*Individual Resumes Folder*](https://github.com/jstngoulet/vorteksolutions/tree/master/Aditional%20Resources/Resum%C3%A9s)

* Redirects to a hosted site with the current list of individual team member's resumés.

[*Meeting Agendas / Minutes*](https://github.com/jstngoulet/vorteksolutions/tree/master/Meeting%20Notes%20and%20Agendas)

* Directory of each meeting we will have, and already had, including meeting notes, agendas and topics we covered.

[*Widget Testing Page*](https://jstngoulet.github.io/vorteksolutions/)

* Location of where the widget is being tested on the master branch.

# Overall Description

## Product Perspective

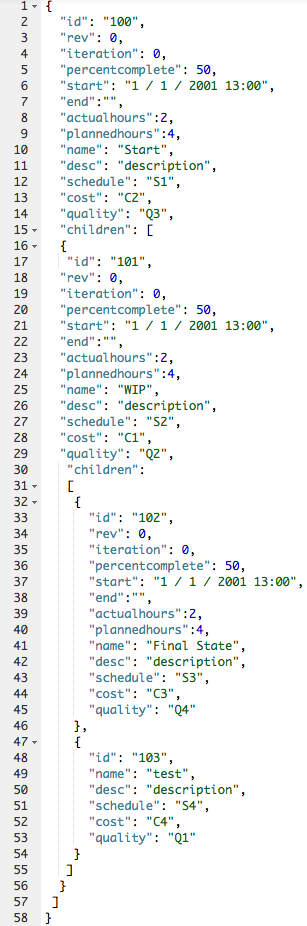
This product is being developed for Northrop Grumman as a tool that allows them to have a full overview of how their workflow is functioning. This product is standalone built on Northrop Grumman’s necessity of having a fully functioning, well designed interface to replace their current system: the use of excel spreadsheets. Contingent upon finishing this product with ample time, a scheduler and then a layout visualization would be created to go along with the workflow widget.

## Product Functions

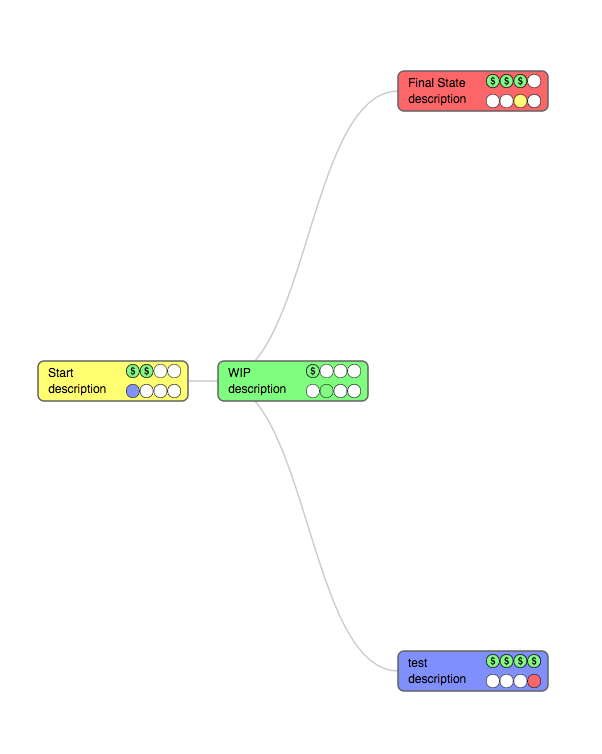
*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

* Widget will accept information regarding workflow passed as JSON
* Widget will display relationships between each node passed within the workflow
* Widget will dynamically adjust UI depending on values within each node

*Example of JSON Input:*



*Result of JSON Input Within Widget:*



## User Classes and Characteristics

For use of this product, our aim is to target developers of whom have particular knowledge working with the hosted servers by the client, those who have sufficient skills managing and editing CSS documentation, and those who can manage JavaScript and JSON scripting / object notions.

As such, there are 2 main groups that will use our product: those in manufacturing and developers.

### Manufacturing

Those in manufacturing will be using this product to determine part state history that will allow them to better troubleshoot. Currently, the system is handled by a series of documents posted on bulletin boards around the facility - our aim is to digitize them.

### Developers

Those within the stakeholder group, developers, will use this product and modify the code / themes to match their needs. The aim of this product is to be as customizable as possible - and this is so developers can alter the program as their needs change over time.

## Operating Environment

This software is currently being tested within a Virtual Machine running Linux Ubuntu 14.04.3 LTS. This software will run with Internet Explorer 8+, Microsoft Edge, Chrome, Firefox, as well as Safari 6+.

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## User Documentation

* [Sencha Ext JS 6.2.1 Documentation](http://docs.sencha.com/extjs/6.2.1/modern/Ext.html)
* [D3 API Reference](https://github.com/d3/d3/blob/master/API.md)
* User Manual (WIP)
* [GitHub Repository](https://github.com/jstngoulet/vorteksolutions)

## Assumptions and Dependencies

This software is dependent on the framework called ‘Ext Js’ which is developed by Sencha. Accompanied by this framework, we are also dependent the Javascript library called D3.

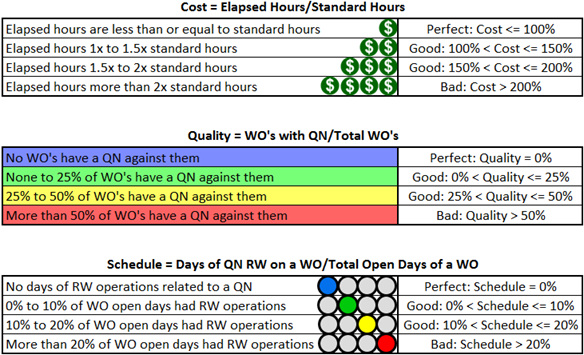
# External Interface Requirements

## User Interfaces

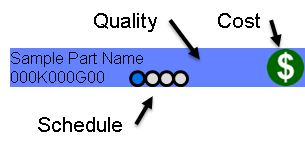
User interface will follow this **application style guide:**



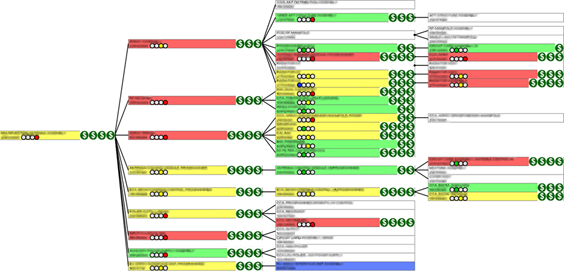
The application UI will adjust based on these passed in conditions:



With these adjusted characteristics of each state, they should apply to each node as such:



Each of these nodes will have their relationship passed in with the JSON data and then the resulting tree hierarchy will appear:



## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

Below demonstrates some of the key features this product will offer at completion.

## Read JSON

4.1.1 Description and Priority

In order for our product to work sufficiently, the product must be able to read JSON data and create a workflow using solely that data. This is the utmost top priority.

4.1.2 Stimulus/Response Sequences

The developers involved in this project should be able to input all JSON fields they wish to be displayed and hide all others. This comes with the customizable states, as mentioned later in this section.

4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

JavaScript (JS), included in the integrated libraries, will be used to enable the JSON integration. The objects are created and stored in JS.

Using the libraries as a base, JSON object must be configurable to show relevant data to the section that wishes to display it (departments, part locales, etc.).

REQ-1:

REQ-2:

## Display a Basic Workflow

4.1.1 Description and Priority

In addition to reading JSON, the product must have the ability to display a simple workflow using said data. This is also a top priority.

4.1.2 Stimulus/Response Sequences

The developers involved in this section are the same as mentioned in the JSON section, with the addition of UI / UX developers. Those involved will be charged with reading the JSON input and creating themes to coherently describe their prescribed data.

4.1.3 Functional Requirements

Just like reading the JSON data, the data should be managed by developers or those that are familiar with Object Notation. The difference, however, lies with the stakeholders using the visual aides - they should not need development experience to manipulate the created charts / diagrams.

When an object in the diagram is selected by a user, the next series of steps should appear. This is to minimize screen space and clutter until it is required.

REQ-1:

REQ-2:

## Allow Customization to the Workflow

4.1.1 Description and Priority

While this step is also critical to the project’s success, we must first focus on creating a working product with at least one (1) built-in theme. This task is to be set directly after said success.

4.1.2 Stimulus/Response Sequences

In order to work effectively, the user should be able to create their own style documentation and implement it by altering just a few lines of code.

Upon editing, the user should refer to the documentation How-To, described in the next section, for alterations of this product.

4.1.3 Functional Requirements

With simple CSS manipulation, developers should have the ability to manage / edit endless themes to work with the widget. With simple customization, the widget should still fully function as expected.

REQ-1:

REQ-2:

## Create a Developer: How-To

4.1.1 Description and Priority

In order for our product to work sufficiently, the product must be able to read JSON data and create a workflow using solely that data. This is the utmost top priority.

4.1.2 Stimulus/Response Sequences

The developers involved in this project should be able to input all JSON fields they wish to be displayed and hide all others. This comes with the customizable states, as mentioned later in this section.

4.1.3 Functional Requirements

JavaScript (JS), included in the integrated libraries, will be used to enable the JSON integration. The objects are created and stored in JS.

Using the libraries as a base, JSON object must be configurable to show relevant data to the section that wishes to display it (departments, part locales, etc.).

REQ-1:

REQ-2:

# Other Nonfunctional Requirements

## Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## Business Rules

*Operating Procedures of software are yet to be determined as this is a stakeholder-level requirement.*

# Other Requirements

*No other requirements have been defined.*

**Appendix A: Glossary**

*No terms yet to be defined.*

**Appendix B: Analysis Models**

*No images have been created yet for this document.*

*Original document template copyright:*

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***