|  |
| --- |
|  |
| Tunnel-K Software Development Plan (SDP) |
|  |
|  |
| **Josh Calahan, Wes Cothran, Chris Davis, Michael Lynch, Brian Pittman** |
| **11/20/2011** |

**Approvals**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Instructor Approval: Dr. Jeffrey Kulick Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Team Member: Josh Calahan Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Team Member: Wes Cothran Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Team Member: Chris Davis Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Team Member: Michael Lynch Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

Team Member: Brian Pittman Date

**Revisions**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| Initial | 11/20/11 | Original Draft |

Contents

[1) Scope 8](#_Toc309714896)

[1.1) Identification 8](#_Toc309714897)

[1.2) System overview 8](#_Toc309714898)

[1.3) Document overview 8](#_Toc309714899)

[1.4) Relationship to other plans 9](#_Toc309714900)

[2) Referenced documents 9](#_Toc309714901)

[3) Overview of required work 9](#_Toc309714902)

[4) Plans for performing general software development activities 9](#_Toc309714903)

[4.1) Software development process 9](#_Toc309714904)

[4.2) General plans for software development 9](#_Toc309714905)

[4.2.1) Software development methods 9](#_Toc309714906)

[4.2.2) Standards and practices for software products 9](#_Toc309714907)

[4.2.3) Traceability 9](#_Toc309714908)

[4.2.4) Reusable software products 9](#_Toc309714909)

[4.2.5) Handling of critical requirements 9](#_Toc309714910)

[4.2.6) Computer hardware resource utilization 9](#_Toc309714911)

[4.2.7) Recording rationale 9](#_Toc309714912)

[4.2.8) Access for acquirer review 9](#_Toc309714913)

[5) Plans for performing detailed software development activities 9](#_Toc309714914)

[5.1) Project planning and oversight 9](#_Toc309714915)

[5.1.1) Software development planning (covering updates to this plan) 9](#_Toc309714916)

[5.1.2) Software item test planning 9](#_Toc309714917)

[5.1.3) System test planning 9](#_Toc309714918)

[5.1.4) Software installation planning 9](#_Toc309714919)

[5.1.5) Software transition planning 9](#_Toc309714920)

[5.1.6) Following and updating plans, including the intervals for management review 9](#_Toc309714921)

[5.2) Establishing a software development environment 10](#_Toc309714922)

[5.2.1) Software engineering environment 10](#_Toc309714923)

[5.2.2) Software test environment 10](#_Toc309714924)

[5.2.3) Software development library 10](#_Toc309714925)

[5.2.4) Software development files 10](#_Toc309714926)

[5.2.5) Non-deliverable software 10](#_Toc309714927)

[5.3) System requirements definition 10](#_Toc309714928)

[5.3.1) Analysis of user input 10](#_Toc309714929)

[5.3.2) Operational concept 10](#_Toc309714930)

[5.3.3) System requirements 10](#_Toc309714931)

[5.4) System design 10](#_Toc309714932)

[5.4.1) System-wide design decisions 10](#_Toc309714933)

[5.4.2) System architectural design 10](#_Toc309714934)

[5.5) Software requirements definition 10](#_Toc309714935)

[5.6) Software design 10](#_Toc309714936)

[5.6.1) Software item-wide design decisions 10](#_Toc309714937)

[5.6.2) Software item architectural design 10](#_Toc309714938)

[5.6.3) Software item detailed design 10](#_Toc309714939)

[5.7) Software implementation and unit testing 10](#_Toc309714940)

[5.7.1) Software implementation 10](#_Toc309714941)

[5.7.2) Preparing for unit testing 10](#_Toc309714942)

[5.7.3) Performing unit testing 10](#_Toc309714943)

[5.7.4) Revision and retesting 10](#_Toc309714944)

[5.7.5) Analyzing and recording unit test results 11](#_Toc309714945)

[5.8) Unit integration and testing 11](#_Toc309714946)

[5.8.1) Preparing for unit integration and testing 11](#_Toc309714947)

[5.8.2) Performing unit integration and testing 11](#_Toc309714948)

[5.8.3) Revision and retesting 11](#_Toc309714949)

[5.8.4) Analyzing and recording unit integration and test results 11](#_Toc309714950)

[5.9) Software item qualification testing 11](#_Toc309714951)

[5.9.1) Independence in software item qualification testing 11](#_Toc309714952)

[5.9.2) Testing on the target computer system 11](#_Toc309714953)

[5.9.3) Preparing for software item qualification testing 11](#_Toc309714954)

[5.9.4) Dry run of software item qualification testing 11](#_Toc309714955)

[5.9.5) Performing software item qualification testing 11](#_Toc309714956)

[5.9.6) Revision and retesting 11](#_Toc309714957)

[5.9.7) Analyzing and recording software item qualification test results 11](#_Toc309714958)

[5.10) Software/hardware item integration and testing 11](#_Toc309714959)

[5.10.1) Preparing for software/hardware item integration and testing 11](#_Toc309714960)

[5.10.2) Performing software/hardware item integration and testing 11](#_Toc309714961)

[5.10.3) Revision and retesting 11](#_Toc309714962)

[5.10.4) Analyzing and recording software/hardware item integration and test results 11](#_Toc309714963)

[5.11) System qualification testing 11](#_Toc309714964)

[5.11.1) Independence in system qualification testing 11](#_Toc309714965)

[5.11.2) Testing on the target computer system 11](#_Toc309714966)

[5.11.3) Preparing for system qualification testing 12](#_Toc309714967)

[5.11.4) Dry run of system qualification testing 12](#_Toc309714968)

[5.11.5) Performing system qualification testing 12](#_Toc309714969)

[5.11.6) Revision and retesting 12](#_Toc309714970)

[5.11.7) Analyzing and recording system qualification test results 12](#_Toc309714971)

[5.12) Preparing for software use 12](#_Toc309714972)

[5.12.1) Preparing the executable software 12](#_Toc309714973)

[5.12.2) Preparing version descriptions for user sites 12](#_Toc309714974)

[5.12.3) Preparing user manuals 12](#_Toc309714975)

[5.12.4) Installation at user sites 12](#_Toc309714976)

[5.13) Preparing for software transition 12](#_Toc309714977)

[5.13.1) Preparing the executable software 12](#_Toc309714978)

[5.13.2) Preparing source files 12](#_Toc309714979)

[5.13.3) Preparing version descriptions for the maintenance site 12](#_Toc309714980)

[5.13.4) Preparing the “as built” software item design and other software maintenance information 12](#_Toc309714981)

[5.13.5) Updating the system design description 12](#_Toc309714982)

[5.13.6) Updating the software requirements specification 12](#_Toc309714983)

[5.13.7) Updating the system/subsystem specification 12](#_Toc309714984)

[5.13.8) Preparing maintenance manuals 12](#_Toc309714985)

[5.13.9) Transition to the designated maintenance site 12](#_Toc309714986)

[5.14) Software configuration management 12](#_Toc309714987)

[5.14.1) Configuration identification 12](#_Toc309714988)

[5.14.2) Configuration control 13](#_Toc309714989)

[5.14.3) Configuration status accounting 13](#_Toc309714990)

[5.14.4) Configuration audits 13](#_Toc309714991)

[5.14.5) Packaging, storage, handling, and delivery 13](#_Toc309714992)

[5.15) Software product evaluation 13](#_Toc309714993)

[5.15.1) In-process and final software product evaluations 13](#_Toc309714994)

[5.15.2) Software product evaluation records, including items to be recorded 13](#_Toc309714995)

[5.15.3) Independence in software product evaluation 13](#_Toc309714996)

[5.16) Software quality assurance 13](#_Toc309714997)

[5.16.1) Software quality assurance evaluations 13](#_Toc309714998)

[5.16.2) Software quality assurance records, including items to be recorded 13](#_Toc309714999)

[5.16.3) Independence in software quality assurance 13](#_Toc309715000)

[5.17) Corrective action 13](#_Toc309715001)

[5.17.1)Problem/change reports, including items to be recorded 13](#_Toc309715002)

[5.17.2) Corrective action system 13](#_Toc309715003)

[5.18) Joint technical and management reviews 13](#_Toc309715004)

[5.18.1) Joint technical reviews, including a proposed set of reviews 13](#_Toc309715005)

[5.18.2) Joint management reviews, including a proposed set of reviews 13](#_Toc309715006)

[5.19) Risk management 13](#_Toc309715007)

[5.20) Software management indicators 13](#_Toc309715008)

[5.21) Administrative security and privacy protection 13](#_Toc309715009)

[5.22) Managing subcontractors 13](#_Toc309715010)

[5.23) Interfacing with software IV&V agents 14](#_Toc309715011)

[5.24) Coordinating with associate developers 14](#_Toc309715012)

[5.25) Project process improvement 14](#_Toc309715013)

[6) Schedules and activity network 14](#_Toc309715014)

[7) Project organization and resources 14](#_Toc309715015)

[7.1) Project organization 14](#_Toc309715016)

[7.2) Project resources 14](#_Toc309715017)

[8) Notes 14](#_Toc309715018)

[A) Annexes 14](#_Toc309715019)

## 1) Scope

## 1.1) Identification

Tunnel-K is an effort by graduate software engineering students at the University of Alabama in Huntsville enrolled in a two-semester Software Engineering Studio course. The endeavor is aimed at building a small-scale wind tunnel and associated software systems and is intended to be used by science museums, schools, etc. for educational purposes. The work is being done in association with the Hands-On Science Center (HOSC) in Tullahoma, TN.

## 1.2) System overview

The overall Tunnel-K system consists of a wind tunnel structure along with associated computer hardware and software, wiring, sensors, motors, fans, power supplies, etc. used for controlling and monitoring the operation of the tunnel. Additionally, a two-dimensional flow solver application suite will provide the opportunity for experimentation with various shapes in a virtual wind tunnel environment and graphically displayed mach and pressure gradients. The suite will also provide integration with the physical wind tunnel controls so that simulated conditions and be illustrated in the real world.

## 1.3) Document overview

This document addresses the plans for performing general and detailed software development activities.

## 1.4) Relationship to other plans

No other project management plans will be utilized for this development.

## 2) Referenced documents

* Tunnel-K Statement of Work (SOW)
* Tunnel-K Rough Order of Magnitude Estimate (ROM)
* Tunnel-K Software Requirements Specification (SRS)
* Trial-Use Standard for Information Technology Software Life Cycle Processes Software Development Acquirer-Supplier Agreement, J-STD-06-1995
* UAH CPE656 Fall 2011 Course Syllabus, Dr. Jeffry Kulick
* Code Conventions for the Java Programming Language, <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>
* Code Style Guidelines for Contributors, (Google Android Style Guide) <http://source.android.com/source/code-style.html>
* Writing a Library for Arduino, <http://arduino.cc/en/Hacking/LibraryTutorial>

## 3) Overview of required work

The requirements and constraints for this effort are identified in the SRS. Documentation requirements and constraints are specified by J-STD-016-1995 as well as the UAH CPE656 course syllabus. The work encompasses the system life cycle from inception through deployment. Support will be provided for operations and maintenance at least through the end of the Software Engineering Studio course, anticipated to end in May of 2012. Free and open-source software (FOSS), APIs, etc. will be used whenever possible to ensure the cost of replicating the system remains as low as possible.

## 4) Plans for performing general software development activities

## 4.1) Software development process

An iterative software development process will be used whereby small pieces of functionality are rapidly designed, implemented, and tested prior to being integrated into the larger system. Numerous builds will, therefore, be made, each supporting the goal of implementing a given portion of functionality.

## 4.2) General plans for software development

## 4.2.1) Software development methods

The software development standard J-STD-016-1995 will be used as a basis for documenting the various work stages in this effort, and the process described in section 4.1 will serve as the primary software development method. Numerous tools will be used to complete the work, including, but not limited to, Google Code, Google Groups, e-mail, Eclipse, the Android operating system APIs, Android emulators, the Arduino development environment, git, ArgoUML, Microsoft Office, Inkscape, Paint.NET, and Twitter.

## 4.2.2) Standards and practices for software products

Developers on this effort will conform to the standards, best practices, and conventions of the programming language(s) used. See section 2, Referenced Documents, for specific standards and conventions documents.

## 4.2.3) Traceability

This subclause shall describe the approach to be followed for performing upward and downward traceability.

## 4.2.4) Reusable software products

The process for identifying, evaluating, and incorporating reusable software products will entail evaluating by engineering judgment the openness of the products, the ease of integrating into the Tunnel-K system, and value of reusing a given product. Products expected to be reused include a 2D flow solver application written by Jim Masters (a consultant to the project), and the Arduino Ethernet library. Both are FOSS, and the benefit of incorporating the existing code far outweighs any drawbacks that may exist. No restrictions exist for the use of either product.

## 4.2.5) Handling of critical requirements

Critical requirements will be identified and handled in close cooperation with the HOSC. Of primary concern is ensuring the safety of science museum patrons, as a majority of them are expected to be children. Privacy of users is also of the utmost concern, as functionality for sharing wind tunnel data and results via social networks will be included in the system. Major security issues are not anticipated given the environment in which the system will exist.

## 4.2.6) Computer hardware resource utilization

The allocation of computer hardware will be handled using engineering judgment. Due to the embedded nature of the

## 4.2.7) Recording rationale

## 4.2.8) Access for acquirer review

## 5) Plans for performing detailed software development activities

## 5.1) Project planning and oversight

## 5.1.1) Software development planning (covering updates to this plan)

## 5.1.2) Software item test planning

## 5.1.3) System test planning

## 5.1.4) Software installation planning

## 5.1.5) Software transition planning

## 5.1.6) Following and updating plans, including the intervals for management review

## 5.2) Establishing a software development environment

## 5.2.1) Software engineering environment

## 5.2.2) Software test environment

## 5.2.3) Software development library

## 5.2.4) Software development files

## 5.2.5) Non-deliverable software

## 5.3) System requirements definition

## 5.3.1) Analysis of user input

## 5.3.2) Operational concept

## 5.3.3) System requirements

## 5.4) System design

## 5.4.1) System-wide design decisions

## 5.4.2) System architectural design

## 5.5) Software requirements definition

## 5.6) Software design

## 5.6.1) Software item-wide design decisions

## 5.6.2) Software item architectural design

## 5.6.3) Software item detailed design

## 5.7) Software implementation and unit testing

## 5.7.1) Software implementation

## 5.7.2) Preparing for unit testing

## 5.7.3) Performing unit testing

## 5.7.4) Revision and retesting

## 5.7.5) Analyzing and recording unit test results

## 5.8) Unit integration and testing

## 5.8.1) Preparing for unit integration and testing

## 5.8.2) Performing unit integration and testing

## 5.8.3) Revision and retesting

## 5.8.4) Analyzing and recording unit integration and test results

## 5.9) Software item qualification testing

## 5.9.1) Independence in software item qualification testing

## 5.9.2) Testing on the target computer system

## 5.9.3) Preparing for software item qualification testing

## 5.9.4) Dry run of software item qualification testing

## 5.9.5) Performing software item qualification testing

## 5.9.6) Revision and retesting

## 5.9.7) Analyzing and recording software item qualification test results

## 5.10) Software/hardware item integration and testing

## 5.10.1) Preparing for software/hardware item integration and testing

## 5.10.2) Performing software/hardware item integration and testing

## 5.10.3) Revision and retesting

## 5.10.4) Analyzing and recording software/hardware item integration and test results

## 5.11) System qualification testing

## 5.11.1) Independence in system qualification testing

## 5.11.2) Testing on the target computer system

## 5.11.3) Preparing for system qualification testing

## 5.11.4) Dry run of system qualification testing

## 5.11.5) Performing system qualification testing

## 5.11.6) Revision and retesting

## 5.11.7) Analyzing and recording system qualification test results

## 5.12) Preparing for software use

## 5.12.1) Preparing the executable software

## 5.12.2) Preparing version descriptions for user sites

## 5.12.3) Preparing user manuals

## 5.12.4) Installation at user sites

## 5.13) Preparing for software transition

## 5.13.1) Preparing the executable software

## 5.13.2) Preparing source files

## 5.13.3) Preparing version descriptions for the maintenance site

## 5.13.4) Preparing the “as built” software item design and other software maintenance information

## 5.13.5) Updating the system design description

## 5.13.6) Updating the software requirements specification

## 5.13.7) Updating the system/subsystem specification

## 5.13.8) Preparing maintenance manuals

## 5.13.9) Transition to the designated maintenance site

## 5.14) Software configuration management

## 5.14.1) Configuration identification

## 5.14.2) Configuration control

## 5.14.3) Configuration status accounting

## 5.14.4) Configuration audits

## 5.14.5) Packaging, storage, handling, and delivery

## 5.15) Software product evaluation

## 5.15.1) In-process and final software product evaluations

## 5.15.2) Software product evaluation records, including items to be recorded

## 5.15.3) Independence in software product evaluation

## 5.16) Software quality assurance

## 5.16.1) Software quality assurance evaluations

## 5.16.2) Software quality assurance records, including items to be recorded

## 5.16.3) Independence in software quality assurance

## 5.17) Corrective action

## 5.17.1)Problem/change reports, including items to be recorded

## 5.17.2) Corrective action system

## 5.18) Joint technical and management reviews

## 5.18.1) Joint technical reviews, including a proposed set of reviews

## 5.18.2) Joint management reviews, including a proposed set of reviews

## 5.19) Risk management

## 5.20) Software management indicators

## 5.21) Administrative security and privacy protection

## 5.22) Managing subcontractors

## 5.23) Interfacing with software IV&V agents

## 5.24) Coordinating with associate developers

## 5.25) Project process improvement

## 6) Schedules and activity network

## 7) Project organization and resources

## 7.1) Project organization

## 7.2) Project resources

## 8) Notes

## A) Annexes