Acceptance Test Plan for DAKA

Version 1.1
April 20, 2014

Prepared by:

Sergey Matskevich, Christopher Harvey, Ernesto Cejas Padilla,
ByeongGil Jeon and Jirakit Songprasit

Stakeholder: iPipeline
Advisors: Mr. William M. Mongan, Dr. Jeffrey Popyack

Drexel University

Revision History

Name	Date	Reason for changes	Version
Sergey Matskevich, Christopher Harvey, Ernesto Cejas Padilla, ByeongGil Jeon, Jirakit Songprasit	February 4, 2014	Initial version	1.0
Sergey Matskevich, Christopher Harvey, Ernesto Cejas Padilla, ByeongGil Jeon, Jirakit Songprasit	April 20, 2014	Updated Traceability Matrix	1.1

Table of Contents

1 Introduction

1.1 Background

This document provides the plan for completing the testing activities required for the Acceptance Test Plan of DAKA. DAKA is a business intelligence tool that uses BigData techniques to extract knowledge from large data sets. DAKA runs data mining algorithms on a variety of large data sets. It is described in greater detail in the Software Requirements Specification for DAKA.

1.2 Structure of Document

- Section 2 - Describes the overall approach to the Acceptance Test Plan process
- Section 3 - Describes in more detail individual issues covered or not covered by the Acceptance Test Plan process
- Section 4 - Describes the criteria which have to be satisfied for the Acceptance Test Plan project
- Section 5 - Describes the roles and responsibilities of the staff members involved in the Acceptance Test Plan project
- Section 6 - Describes the test cases used during the Acceptance Test Plan

1.3 Glossary

Classification

The process of finding a model (or function) that describes and distinguishes data classes or concepts. The model are derived based on the analysis of a set of training data.

Closed Itemset

An itemset with no proper superset that has the same support count.

Confidence

The conditional probability of a set of attributes to appear in the data if this set has a given support.

Data Node

A machine in HDFS which stores file data.

Frequent Itemset

An itemset that occurs in the data with specified minimum support.

Frequent Pattern

A set of individual attributes that is shared among many different entities of data.

Functional Requirements

Functional Requirements define the internal workings of the software.

Hadoop Distributed File System (HDFS)

A distributed file system designed to run on commodity hardware.

Horizontal Scalability

A feature of software to increase performance linearly with the addition of new nodes in a cluster.

Itemset

A set of items in a transaction.

Maximal Itemset

An itemset that is not a subset of any other itemset.

Name Node

A machine in HDFS which keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept.

Project Leader

The Project Leader is the person in charge of the whole project.

Stakeholder's Representatives

The Stakeholder's Representatives are people who overlook the Acceptance Test Plan execution on behalf of the customers.

Support

The probability of a given set of attributes to be in a random entry in database

System Test

System Test is conducted after Integration Test has been conducted to evaluate the system's compliance with its specified requirements.

Test Team Leader

The Test Team Leader is the person in charge of all the testers.

Training Data

A set of data objects for which the class labels are known.

Unit Tests

Unit Tests are procedures in the software to validate that individual modules and other units of source code are working properly.

2 Test Approach and Constraints

This section describes the overall approach, particular techniques and testing tools which are used during the Acceptance Test Plan of DAKA and any constraints that may apply.

2.1 Test Objectives

The Acceptance Test Plan process prompts the stakeholder to evaluate DAKA and verify whether it performs in accordance with the stakeholder’s requirements, listed in the Software Requirements Specification.

2.2 Test Structure

The Acceptance Test Plan consists of a subset of test cases and methods, previously utilized in the Unit Tests, Integration Test and System Test conducted on the DAKA. The test cases are carefully selected and agreed upon by both the developers and the stakeholder, and are allowed for the most adequate verification of the functional requirements of the DAKA, as listed in the Software Requirements Document, without the extensiveness of the full-scale System Test.

It is essential that all appropriate Unit Tests, Integration Test and the System Test were successfully performed for DAKA prior to the Acceptance Test Plan and their results were reported and presented to the stakeholder.

3 Test Assumption and Exclusion

3.1 Introduction

This section provides more details about what issues and features of DAKA are covered by Acceptance Test Plan process, and what issues and features of DAKA are not covered.

3.2 Test Assumptions

It is assumed that all issues covered by the Acceptance Test Plan were also previously addressed by the Unit Tests, Integration Test and System Test of DAKA.

The Acceptance Test Plan is covered

- The functional requirements of the system listed in the Software Requirements
- Specification Usability of the system
- Consistency of the user related system documentation

3.3 Test Exclusions

It is assumed that all issues not covered by the Acceptance Test Plan were previously addressed by Unit Tests, Integration Tests and System Tests of DAKA.

The Acceptance Test Plan is not covered

- The non-functional requirements of the system (except the aforementioned Usability) listed in the Software Requirements Specification.
- Structural integrity of the source code.

4 Entry and Exit Criteria

4.1 Introduction

This section lists the criteria which must be satisfied in order for the Acceptance Test Plan to begin, as well as the criteria which must be satisfied in order for the Acceptance Test Plan to stop.

4.2 Entry Criteria

- DAKA is successfully undergone Unit Tests, Integration Test and System Test.
- The testing environment which satisfies the system Requirement of Software Requirements Specification is setup and inspected by the stakeholder's representative.
- A copy of the latest version of the Software Requirements Specification is received
- The latest released version of DAKA is appropriately resources.
- Consent of the Project Leader is obtained.
- Consent of the Stakeholder is obtained.
- Consent of the Test Team Leader is obtained.

4.3 Exit Criteria

The Acceptance Test Plan should be halted after one of the following:

- All Priority 1 requirements were tested without any deviation from expected behavior.(Success)
- At least one Priority 1 requirement deviated from the document specification.(Failure)
- By Mutual agreement between Stakeholder's Representative and the Tester, in which both parties'' supervisor should be notified and the Acceptance Test Plan should be rescheduled for a later date. (Failure)

5 Testing Participants

5.1 Introduction

This section describes the roles and responsibilities of the staff members involved in the Acceptance Test Plan, as well as the procedure of reporting the test results and any problems that came up during testing.

5.2 Roles and Responsibilities

For the Acceptance Test Plan, the following roles were assumed by the following people

- Test Team Leader: Christopher Harvey
- Stakeholder’s Representative: A person in charge from the stakeholder's side who overviews the testing process.
- Tester: person who executes the use case tests

5.3 Training Requirements

All parties involved in the Acceptance Test Plan are familiar with the Linux based operating system and user interface of DAKA, as well as with the system documentation and the software Requirement Specification.

5.4 Problem Reporting

Any problem pointed out by either the Stakeholder's Representative or the Tester must be documented and reported to the Test Team Leader. Later the problem report is submitted to the project Leader, and addressed during a periodic or urgent staff meeting depending on the severity of the problems.

5.5 Progress Reporting

The Acceptance Test Plan Report is compiled once, after testing process is finished by the Test Team Leader and submitted to the Project Leader.

6 Test Cases

6.1 Introduction

The test cases are distributed in sections covering functionality elements and use cases in the Software Requirements Specification. Each of the following test cases is in the format.

Name - The name of the test case
Preconditions - Conditions needed to initiate the test case
Actions - The actions expected from a tester
Post conditions - The expected outcome of the test case

6.2 Test Cases

6.2.1 Input

Preconditions	The tester has input file provided from company.
Actions	The tester inputs the CSV file to the system.
Postconditions	The system accepts input file format as CSV.

6.2.2 Output

Preconditions	The tester has input file in the system.
Actions	The tester runs the command line interface.
Postconditions	The system generates output file in HDFS.

Preconditions	The tester has input file in the system.
Actions	The tester runs the Frequent Pattern algorithm
Postconditions	The system generates output according to Requirement 3.3.2.

Preconditions	The tester has input file in the system.
Actions	The tester runs the command line interface.
Postconditions	The system generates output according to Requirement 3.3.3.

6.2.3 Interface

Preconditions	The tester is at the command line.
Actions	The tester types -i [PATH] in the command line.
Postconditions	The system is specified data input folder in HDFS.

Preconditions	The tester is at the command line.
Actions	The tester types -o [PATH] in the command line.
Postconditions	The system is specified data input folder in HDFS.

Preconditions	The tester is at the command line.
Actions	The tester types -t FPGrownth in the command line.
Postconditions	The system runs the frequent pattern algorithm.

Preconditions	The tester is at the command line.
Actions	The tester types -t Classify in the command line.
Postconditions	The system runs the classification algorithm.

Preconditions	The tester is at the command line.
Actions	The tester types -s [VALUE] in the command line.
Postconditions	The system uses the specified minimum support value.

Preconditions	The tester is at the command line.
Actions	The tester types -l [VALUE] in the command line.
Postconditions	The system uses the specified location for training data set.

7 Requirements Traceability Matrix

7.1 Traceability by Requirement

Non-functional requirements are labeled "NFR" to denote that there are no explicit software components designed to specifically satisfy them. These requirements have cross-cutting concerns in which many different components and environmental factors address.

Requirement	Test case
3.1.1	6.2.2
3.1.2	6.2.2
3.1.3	6.2.2
3.1.4	6.2.3
3.1.5	6.2.3
3.2.1	6.2.1
3.2.3	6.2.1
3.2.4	6.2.1
3.3.1.1	6.2.2
3.3.2.1	6.2.2
3.3.2.2	6.2.2
3.3.2.3	6.2.2
3.3.2.4	6.2.2
3.3.2.5	6.2.2
3.3.3.1	6.2.2
3.3.3.2	6.2.2
3.3.3.3	6.2.2
3.4.1	NFR
3.4.2	NFR
3.4.3	NFR
3.5.1	6.2.3
3.5.2	6.2.3
3.5.3	6.2.3
3.5.4	6.2.3
3.5.6	6.2.3

7.2 Traceability by Test Case

Test case	Requirement
6.2.1	3.2.1, 3.2.2, 3.2.4

6.2.2	3.1.1, 3.1.2, 3.1.3, 3.3.1.1, 3.3.2.1, 3.3.2.2, 3.3.2.3, 3.3.2.4, 3.3.2.5, 3.3.3.1, 3.3.3.2, 3.3.3.3
6.2.3	3.1.4, 3.1.5, 3.5.1, 3.5.2, 3.5.3, 3.5.3, 3.5.6

8 Appendices

8.1 References

Cejas Padilla, Ernesto, Christopher Harvey, ByeongGil Jeon, Sergey Matskevich, and Jirakit Songprasit. 2014. "Software Requirements Specification for DAKA." Philadelphia.

Internet Pipeline, Inc. n.d. "About iPipeline." *iPipeline.* http://ipipeline.com/company/about-ipipeline.php.

The Apache Software Foundation. n.d. *Hadoop Home Page.* http://hadoop.apache.org/.