Contents

[1. Introduction 2](#_Toc318816942)

[1.1 Project Abstract 2](#_Toc318816943)

[1.2 About the Document 2](#_Toc318816944)

[1.3 References 3](#_Toc318816945)

[2. Testing Scope and Deliverables 4](#_Toc318816946)

[2.1 Scope Included 4](#_Toc318816947)

[2.2 Scope Excluded 4](#_Toc318816948)

[2.3 Deliverables 5](#_Toc318816949)

[3. Testing Schedule 5](#_Toc318816950)

[4. Testing Resources 6](#_Toc318816951)

[5. Testing Environment 6](#_Toc318816952)

[5.1 Software Requirements 6](#_Toc318816953)

[5.2 Hardware Requirements 7](#_Toc318816954)

[6. Entry and Exit Criteria 7](#_Toc318816955)

[6.1 Entry Criteria: 7](#_Toc318816956)

[6.2 Exit Criteria: 7](#_Toc318816957)

[6.3 Suspension and Resumption Criteria 8](#_Toc318816958)

[7. Defect Management 8](#_Toc318816959)

[7.1 Defect Classification 8](#_Toc318816960)

[7.2 Defect Status Life Cycle 8](#_Toc318816961)

[8. Risks in Testing 9](#_Toc318816962)

[9. Change history 9](#_Toc318816963)

# Introduction

## Project Abstract

Milling is a process that removes a section of casing for sidetracking or deviating a well. In order to position the drill tools so that it deviates, a Window Milling System is used. This system provides accurate measurements of the position of mill with respect to the casing at any given depth. At different positions of the mill, for instance when the blades start to cut or mill begins to cut, an operator on the rig floor marks the pipe with a grease pencil. Marking the pipe at different stages of milling is necessary to know when a casing has been milled through. However, marking a drill pipe that is in rotation is an unsafe practice.

ControlCut will provide a safer and more efficient platform to perform this task, by allowing pipe marking to be done electronically. This also provides better, more accurate records of the actions performed on the rig. Data obtained from these jobs can be recorded and analyzed to further optimize milling operations.

The application will be connected to a simulator for use in trainings.

## About the Document

* **Purpose of the document**

The purpose of this document is to define a detailed, comprehensive plan for manual functionality testing, System testing, Integration testing, Performance and Load testing for Weatherford *‘ControlCut application’*  for the entire ControlCut Project team.

* **Target Audience**

This document is meant for all QA/QC team for their reference and no part of it is allowed for modification without prior approval from Quality Manager.

* **Document Organization**

This document is a part of set of documents prepared for *ControlCut application* testing. The other related documents are test cases, test data and test results.

* **Abbreviations and Definitions**

|  |  |
| --- | --- |
| **Version** | **Description** |
| WPTS | Weatherford performance tracking system |
| SRS | System Requirement Specification |
| RFP | Request For Proposal |
| FRS | Functional Requirement Specification |

## References

|  |  |  |
| --- | --- | --- |
| **#** | **Input Documents** | **Description** |
| 1 | Proposal | WFI-TSH-ESS-ControlCut-Proposal V 2.0 |
| 2 | SRS | ControlCut SRS |
| 3 | RFP | ControlCut RFP\_May update |

# Testing Scope and Deliverables

## Scope Included

|  |  |
| --- | --- |
| **Activity** | **Description** |
| Unit Testing | Generic Unit test cases will be prepared and executed on build under test |
| Functional Testing | Manual testing will be done based on the approved functional test cases. |
| Integration testing | Manual testing will be done based on the approved integration test cases. |
| Regression & Re-testing | This includes verifying the defects fixed for a given build and execution of all existing test cases |
| Performance and load testing | The schedule and Environment for Performance testing to be decided later |
| Code Coverage | The schedule and Environment for Performance testing to be decided later |

## Scope Excluded

* Testing of ControlCut client with Nucleus
* Testing with actual WITS data. Test team will depend on the simulator provided by the development team.
* Accuracy of calculations provided by HydraPro engine will be limited to ensuring that the data reflected in Hydrapro matches to that of Nucleus for a given set of test data.
* Accurate job results

## Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Owner** | **Remarks** |
| Test Plan Document | QC |  |
| Test cases | QC |  |
| Test Data | QC |  |
| Test Results | QC |  |
| Versioned baselines of test inputs, data and test results from various test cycles | QC |  |
| Automation Scripts | QC | This will depend on the availability of resources and business priority |

# Testing Schedule

Testing Schedule will be aligned with the overall Project schedule.

The schedule for testing a given build would depend on **the features made available** in the same **and also the required scope**. The test team will perform the following tasks for any given build.

* Product understanding Sanity testing
* Defect/Feature testing
* Regression

# Testing Resources

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Contact info** | **Role** | **Responsibilities** |
| Ashok Krishna | Ashok.krishna@me.weatherford.com | QC Team | * Test case preparation, * Test execution, * Defect logging, |
| Deepankar Bandopadhya | [Deepankar.Bandopadhyay@me.weatherford.com](mailto:Deepankar.Bandopadhyay@me.weatherford.com) | QC Manager | Signoffs and Approvals |

# Testing Environment

## Software Requirements

|  |  |
| --- | --- |
| **Software** | **Purpose** |
| **Team Explorer and Test Manager** | **Maintaining test cases and defects in TFS** |
| **Windows XP** | **Operating system for Client machine** |
| **Win 2003** | **Operating System for server machine** |
| **WIN 7 32 bit** | **Operating system for Client machine** |
| **WIN 7 64 bit** | **Operating system for Client machine** |
| **Visual Studio 2010 Ultimate edition** | **Developing**  **automation scripts** |

## Hardware Requirements

|  |  |  |
| --- | --- | --- |
| **Type of Test** | **Description**  **of Hardware Items** | **Model No.**  **/ Remarks** |
| Unit, Functional, integration and Regression | Client machine | Processor: **Intel Core 2 Duo**  Ram: 3 GB |
| Unit, Functional, integration and Regression | Server Machine | Standard Desktop machine |

# Entry and Exit Criteria

## Entry Criteria:

* Clearly defined scope of testing
* Availability of sufficient functional knowledge with QC team. Test team will rely on inputs from the development team and also seek inputs from the subject matter experts working with other groups
* Test plans and test cases reviewed and signed off
* Working Test Environment
* No open show stopper defects

## Exit Criteria:

* All test cases executed successfully
* All high and critical defects fixed are tested successfully
* No Showstoppers
* No open show stopper defects

## Suspension and Resumption Criteria

**Suspension Criteria:**

If application fails to deliver the basic functionality with critical bugs whose impact on application is high then application testing cannot proceed further.

**Resumption Criteria**

Application delivers basic functionality with all high severity defects fixes

# Defect Management

Defect Classification

|  |  |  |
| --- | --- | --- |
| **Severity** | **Value** | **Description** |
| Critical | 0 | Defects that cause the system to crash, corrupt data files, or completely disrupt service. |
| High | 1 | Defect results in severely impaired functionality. A work around may exist but its use is unsatisfactory. In general, you would not release the product with such a defect. |
| Medium | 2 | Defect causes failure of non-critical aspects of the system. There is a reasonably satisfactory work around. The product may be released if the defect is documented, but the existence of the defect may cause customer dissatisfaction. |
| Low | 3 | Defect of minor significance. A work around exists or, if not, the impairment is slight. Generally, the product could be released and most customers would be unaware of the defect's existence or only slightly dissatisfied. |

Defect Status Life Cycle

|  |  |
| --- | --- |
| **Status** | Event |
| **New** | When a new defect is logged |
| **Resolved** | After the defect is fixed by the development team |
| **Closed** | After the defect is tested successfully by the QC team |

# Risks in Testing

Risk analysis is a technique to identify and assess factors that may impact the schedules and quality of the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Items** | **Likelihood** | **Project**  **Impact** | **Mitigation** | **Owner** |
| Delay in release of application | High | High | Publish project plan after joint estimates | Development Team, PM |
| Undocumented feature requests | High | High | Changes should be minimal | Biz, Development Team |
| Non availability of Test environment | Low | High | Provide Test environment in time | Development Team |
| Non availability of input data | Low | High | System I/p should be provided | Development Team |
| Untrained testers | Medium | High | Testers should be trained for first cycle with proper functional/domain knowledge about the application | QC Manager |

# Change history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date of Change** | **Changed by** | **Changes made** |
| 0.1 | 24-Feb-2012 | Deepankar Bandopadhya | Draft version |
| 0.2 | 01-Mar-2012 | Charul Sharma | Review comments |
| 0.3 | 06-Mar-2012 | Deepankar Bandopadhya | Incorporated  review comments |