**Usability Testing and Verification**

Lukas Klöck - 11014902

Assignment 1:

**Test Plan**

[1. Introduction 2](#_Toc159688714)

[2. Objectives and Tasks 2](#_Toc159688715)

[3. Scope 2](#_Toc159688716)

[4. Testing Strategy 2](#_Toc159688717)

[5. Hardware Requirements 2](#_Toc159688718)

[6. Environment Requirements 2](#_Toc159688719)

[7. Test Schedule 2](#_Toc159688720)

[8. Control Procedures 2](#_Toc159688721)

[9. Features to be Tested 2](#_Toc159688722)

[10. Features not to be Tested 2](#_Toc159688723)

[11. Resources/Roles & Responsibilities 2](#_Toc159688724)

[12. Schedules 2](#_Toc159688725)

[13. Significantly Impacted Departments (SIDs) 2](#_Toc159688726)

[14. Dependencies 2](#_Toc159688727)

[15. Risks/Assumptions 2](#_Toc159688728)

[16. Tools 2](#_Toc159688729)

[17. Approvals 2](#_Toc159688730)

# Introduction

In the following report, the procedure of testing of the Function “Array Shift” will be described. The tests are subject to the following specifications and follow the objective of extensively checking the following functions for their *suitability, usability, and reliability* regarding the absence of errors and flaws. The above mentioned function is required to fulfil three different practices. For the input, there must be two different input criteria, firstly and numerical array and secondly a specific value which is expected to be contained within the array. In case the function works as desired three different outputs should be provided, depending on the given value to be analyzed:

1. If the value is contained in the array, the output should be the next sequential value of the array
2. If the value is not contained in the array, the output should be the first element of the array.
3. If the value is the last element of the array, the output should be -1.

# Objectives and Tasks

# Scope

# Testing Strategy

# Hardware Requirements

# Environment Requirements

# Test Schedule

# Control Procedures

# Features to be Tested

# Features not to be Tested

# Resources/Roles & Responsibilities

# Schedules

# Significantly Impacted Departments (SIDs)

# Dependencies

# Risks/Assumptions

# Tools

# Approvals

# Source code #

const assert = require("assert");

function arrayShift(array, value) {

let index = array.findIndex((element) => element === value);

if (index < 0) return array[0];

if (index == array.length -1) return -1;

return array[index + 1]; };

//const testarr = [1,2, 3, 4, 5]

//const testval = 1;

//console.log(arrayShift(testarr, testval));

assert.equal(arrayShift(data, 3), 7);

assert.equal(arrayShift(data, 5), 2);

assert.equal(arrayShift(data, 5), -1);

console.log("Finished");