Git Client Tool Validation

Super Smart People

July 16, 2022

Abstract

Throughout this document Git is the Configuration Item that is being validated and Report Package is Git Report Package. Git is considered validated after this Report Package has been executed, test evidence has been obtained, and that test evidence supports the conclusion the Intended Use Requirements (IUR) have been satisfied.

Contents

1 Introduction . . . **3**
1.1 Overview . . . 3
1.2 Purpose . . . 3
1.3 Scope . . . 3
1.4 Deviations . . . 3
1.5 Tool Validation Objectives . . . 3
1.6 General Terms . . . 4
1.7 Configuration Item Specific Terms . . . 4
1.8 General Acronyms . . . 5
1.9 Configuration Item Specific Acronyms . . . 5
1.10 References . . . 5
1.11 Training . . . 5
1.12 Tool Validation Test Approach . . . 5
1.13 Configuration Management . . . 5
1.14 Test Plan Instructions . . . 5
1.15 Test Plan Storage and Review . . . 6

2 Intended Use . . . **7**

3 Test Plan Overview . . . **8**

4 Prerequisites . . . **9**

5 Test Evidence . . . **10**
5.1 Test Plan: Git Client Validation Plan . . . 10
5.1.1 Test Suite: Git Client Setup . . . 10
5.1.2 Test Case: Check Install . . . 10
5.1.3 Test Case: Configure Git . . . 12
5.1.4 Test Case: Create a Local Git Repository . . . 20
5.1.5 Test Suite: Clone Remote Repository . . . 36
5.1.6 Test Case: Clone Repository . . . 36
5.1.7 Test Case: Revert a merged change . . . 42
5.1.8 Test Suite: Quick Fix to Software . . . 51
5.1.9 Test Case: Create a branch named Quick Fix . . . 51
5.1.10 Test Case: Undo a program change made on branch Quick Fix. . . . 67
5.1.11 Test Case: Merge a programming change made on branch named Quick Fix . . . 83
5.1.12 Test Suite: Git Tagging . . . 99
5.1.13 Test Case: Tag Commits . . . 99

6 Configuration Item Conclusion 114

Change Summary 114

1 Introduction

1.1 Overview

Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git allows repository clones to act as fully functioning repositories complete with history and full revision tracking capabilities that are not dependent on network access or a central server.

1.2 Purpose

This Report Package is a detailed record that provides a Configuration Item overview, a list of its Intended Use Requirements (IUR), one or more Test Reports, evidence the Test Reports ran, along with the output produced by the Test Report. The Test Report includes a pass/fail result for each Test Step and Test Report, a statement indicating the Configuration Item has a Configuration Identification and a conclusion that the Configuration Item has been validated for its intended use.

1.3 Scope

This Report Package applies to Company medical device software projects that have determined a Configuration Item must be validated for its intended use. This Report Package covers activities associated with validating a Configuration Item for its intended use requirements.

1.4 Deviations

The process governing the creation of this protocol and report deviates from the normal standard operating procedure (SOP005 Validation of Computerized Systems). This document combines both the protocol and the report. Normally the protocol is released first and report is released after the protocol is executed. This document represents an automated protocol execution facilitated through the use of automation scripting and software. The review of a paper protocol and pre-approval of said protocol does not satisfy the need to review the automated components used for the generation of this document. As a result, the automated components which codify the actual test protocol are reviewed by a technical approver as this document and the components are developed. This technical approver is an approver of this document and their approval indicates the automated components effectively test the article under test to meet the intended use as specified in the user requirements.

Additionally data obtained from the execution of the protocol is collected and presented in the grey boxes as objective evidence from the automated test application. Normally this would not be presented together with the protocol, but given this is an automated process in a combined document; this is an effective means of retaining and presenting the objective evidence for review and approval.

Finally, presenting the protocol and the report together allows for a single step automation process that can be easily maintained and re-executed. Re-execution is often desired due to changes to the article under test or changes to user needs.

1.5 Tool Validation Objectives

Document 123-VNV-056022 Validation Determination report provides a Determination of Validation decision tree and Determination of Level of Risk and Validation Rigor decision tree to aid a Development Team when assessing the need for validation. 123-VNV-056022 was updated to indicate this tool required Validation and the Level of Risk needed. The steps below describe the steps used to validate a tool.

1. Describe the intended use of the tool.
2. Set the purpose and scope for the tool validation effort.
3. Enumerate intended use requirements.
4. Disclose compliance criteria.
5. Define Tool validation acceptance criteria.

6. Identify responsible persons and their roles.
7. Document required deliverables.
8. Define specific test steps and test steps to confirm that the Tool's intended use requirements have been met.
9. Collect test evidence.
10. Record Tool validation conclusion.

1.6 General Terms

Configuration Control The systematic process for managing changes to and established baseline.

Configuration Identification A unique identifier used to associate a collection of software artifacts.

Configuration Items Software source code, executables, build scripts, and other software development and software test artifacts relevant to creating and maintaining a software project.

Configuration Status Accounting The recording and reporting of the information needed to effectively manage the software and documentation components of a software project.

Report Package A detailed record that provides a Configuration Item overview, a list of its Intended Use Requirements (IUR), one or more Test Reports, evidence the Test Reports ran along with the output produced by Test Report including a pass/fail result for each Test Step and Test Report, and a statement indicating the Configuration Item has a Configuration Identification, and conclusion that the Configuration Item has been validated for its intended use.

Test Plan A test plan is a collection of one or more test suites a tester has determined to use to challenge requirements.

Test Suite A test suite is a collection of one or more test cases a tester has determined to use to challenge requirements.

Test Case A test case is a set of conditions under which a tester will determine whether the test is working as it was originally established for it to do.

Test Step A unique test identifier with predetermined expectation, confirmation criteria, and pass/fail result.

Test Report A test report consists of Detailed instructions for the set-up, execution, and evaluation of results for a given test. The test protocol may include one or more test cases for which the steps of the protocol will repeat with different input data. Test cases are chosen to ensure that corner cases in the code and data structures are covered. A test protocol may be a script that is automatically run by the computer.

1.7 Configuration Item Specific Terms

Branch A line of development.

Checkout Updates files in the working tree to match the version in a remote repository.

Clone To make a local copy of a remote repository.

Commit Record changes to a repository.

HEAD A pointer to the commit being worked on.

log A record of commit entries.

merge Join two or more development histories together. The index or the specified tree. If no paths are given, git checkout will also update the HEAD to set the specified branch as the current branch.

tag - annotated Tag objects are called 'annotated' tags; they contain a creation date, the tagger name and email, a tagging message, and an optional GnuPG signature.

tag - lightweight Simply a name for a commit object. The object is usually a commit object.

1.8 General Acronyms

FDA Food and Drug Administration

IUR Intended Use Requirements

LMS Learning Management System

SOP Standard Operating Procedure

SOUP Software Of Unknown Provenance

1.9 Configuration Item Specific Acronyms

n/a Not applicable

1.10 References

- SOP004 Software Devopment Procedure
- SOP003 Software Configuration Management
- SOP001 Good Documentation Practices
- SOP005 Validation of Computerized Systems
- SOP002 Change Management

1.11 Training

Company's training records are stored in the Quality Management System. Additional training is not required because this is an automated test that is executed by the automated testing platform. SOP004 Software Devopment Procedure provides training required to create, maintain, and execute this testing protocol.

1.12 Tool Validation Test Approach

This Test Plan describes a series of Test Suites, Test Cases, and Test Steps. When executed, each Test Step determines if the Configuration Item satisfies one or more system requirements. When a Test Step indicates that the system requirements are satisfied, the Test Step's result is "pass". Otherwise, the Test Step's result is "fail". The computer records all "pass" and "fail" results in the Test Plan record. The Configuration Item is considered verified when all Test Steps are executed and the Test Plan record contains no "fail" results. Each Test Step that results in a deviation, observation, incident, or failure shall be represented in the final report.

1.13 Configuration Management

When a Configuration Item is changed, we will review the manufacturer's release notes or our design history file (DHF) to determine if regression testing or adjustments to this Report Package is necessary. We will verify the changes do not impact product operation, product quality, or quality decision made prior to performing the upgrade.

1.14 Test Plan Instructions

This Test Plan describes Test Suits, Test Cases, and Test Steps that demonstrate how the Configuration Item satisfies the IUR. Each Test Plan describes any setup criteria needed to conduct the test. Each Test Plan contains a list of IUR's and the steps that demonstrate how the Configuration Item satisfies the IUR. Each Test Step is marked passed or failed as it is completed. Each Test Plan is marked passed when all Test Steps pass or failed if a single Test Step fails. Failures are addressed per SOP002 Change Management. This serves as a record of the completed test.

Test Plans are automatically run by the computer, generating a report in PDF format. This Report Package is reviewed prior to execution per SOP004 Software Devopment Procedure. The Report Package

is routed and archived in the Quality Management System. When it becomes necessary to annotate a computer generated document SOP001 Good Documentation Practices must be followed.

1.15 Test Plan Storage and Review

This Test Plan is part of a Company's automated validation framework. The framework consists of following parts:

LaTeX files are used to provide an Abstract, Introduction, Intended Use Requirements, Test Plan Overview, Test Equipment, Configuration Item Validation, Conclusion, and Change Summary. LaTeX files are converted assembled into PDF documents. PDF documents are routed using the Company's document management system for approval.

Ruby software is used to run the automated framework to collect test evidence.

Git is used as the storage repository for LaTeX & YAML files, a Git pull-request is used to review the LaTeX & YAML files prior to use.

Evidence Test Plan output includes one Test Suite, Test Plan, and Test Step, and Test Evidence.

YAML files define the Test Plan, Test Suite, and Test Steps that are processed to generate test evidence.

2 Intended Use

Intended use requirements are defined using the following story format:
As a <type of user>, I want <some goal> so that <some reason>

IUR-000 As a developer, I want to configure a Git client appropriate to my operating system so that I can interact with Git server.

IUR-001 As a developer, I want to create and modify a local repository so that I can backup my daily work products.

IUR-002 As a developer, I want to add, modify and delete files from a local repository so that I can preserve my daily work products.

IUR-003 As a developer, I want to obtain files from a remote repository so that I can synchronize my work with my team mates.

IUR-004 As a developer, I want to annotate changes made to a local repository so that I track my work products.

IUR-005 As a developer, I want to isolate my work products so that I can work on multiple assignments concurrently.

IUR-006 As a developer, I want know the status of my local repository so that I can determine the state of my work products with respect to revision control.

IUR-007 As a developer, I want to share my work products with other developers so that I can support team collaboration.

IUR-008 As a developer, I want to know the difference between my local repository and a remote repository so that I can correctly integrate my work with the work of others.

IUR-009 As a developer, I want to apply a tag to a git-commit so that I can re-create a branch by tag.

IUR-010 As an operations personnel, I want to use a source control system to manage and track relevant system state and configuration within a device.

3 Test Plan Overview

This section describes Test Plans, Test Suites, Test Cases, and Test Steps that demonstrate how a Configuration Item satisfies the IUR. Each Test Plan describes any setup criteria needed to conduct the Test Steps. Each Test Plan contains a list of IUR's and the Test Steps that demonstrate how the Configuration Item satisfies the IUR. Each Test Step is marked passed or failed as it is completed. Each Test Plan is marked passed when all Test Steps pass or failed if a single Test Step fails. This serves as a record of completed Test Plans and Test Steps.
Each Test Plan is described in its own section. The order the Test Plans are listed is the order they are run. Each Test Plan defines:

name	Each Plan, Suite, and Case has a unique name.
purpose	Each Plan, Suite, and Case has a purpose.
Test Steps	Each step has a confirmation and expectation along with the command needed to challenge the IUR.
Objective Evidence	A record the Test Plan was run along with any evidence collected while the Test Steps were run.
Traceability	Suites and Cases are traced to an IUR that is challenged. IUR can be traced to multiple Suites and Cases.

Each Test Plan, Test Suite, Test Case, and Test Step has been designed to be run by the computer. However, a person may choose to manually run the Test Step, save the test results, and generate this test report as specified in the appropriate design documentation. The example below runs these commands:

1. git help
2. cat .gitconfig

The output from both commands are written to the system console.

```
plan:
  name: A Test Plan Name
  purpose: purpose of the plan

suite:
  name: A Test Suite Name
  purpose: a suite purpose
  requirement: IUR01 and IUR02

  - case:
    name: A Test Case name
    purpose: A Test Case purpose
    steps:
      - confirm: Confirm git help is written to the console output.
        expectation: Git help is displayed.
        command: git
        argument: help

      - confirm: Confirm .git config is written to the console.
        expectation: .gitconfig is written to the console output.
        command: cat
        argument: .gitconfig
```

4 Prerequisites

1. A valid Git username and password. Request these from your Git administrator.
2. Read and write access to https://github.com/traap//quicksort
3. A private / public SSH key.
4. Your public key needs to be upload to your Git account. Request assistance from your Git administrator.
5. Administrative rights to your computer. Contact your computer systems administrator.
6. Your computer requires HTTP, HTTPS, and SSH access to the Internet. Contact your network administrator.
7. Configure your /.ssh/config file to reference Git your Git Server.

```
# Access to HostName when disconnected from a Company network.
Host github.com
    User username
    IdentityFile ~/.ssh/github_rsa

# All hosts
Host *
    ServerAliveInterval 300
    IdentityFile ~/.ssh/github_rsa
    User gahoward
    Port 22
```

5 Test Evidence

The Company's automation framework assembles the content in this section. The section has one or more Test Plans, Test Suites, Test Cases, and Test Evidence. The evidence provided is used to conclude the Tool has met the Intended Use Requirements.

5.1 Test Plan: Git Client Validation Plan

Purpose: This plan uses white box methods. The output is sufficient for DHF purposes. However, the output is generally only used by engineers.

5.1.1 Test Suite: Git Client Setup

Purpose: This test suite ensures the Git client has been installed and is operational.

Requirement: IUR-000, IUR-001, IUR-002, IUR-003, IUR-004, IUR-005, and IUR-006.

5.1.2 Test Case: Check Install

Purpose: This protocol is used to demonstrate that the Git Client program has been properly installed and configured. This is accomplished by asking the operating system where the program is installed.

Requirement: IUR-000

Step: 1

Confirm: The git client has been installed.

Expectation: git installation location is listed.

Command: which git

Execution start: Jul 16, 2022 11:36:45.246399

Execution end: Jul 16, 2022 11:36:45.247006

Test Result: PASS

Evidence: Starts on next line.

```
1 /usr/bin/git
```

Step: 2

Confirm: The git client version is displayed.

Expectation: git reports it version on the console.

Command: git –version

Execution start: Jul 16, 2022 11:36:45.247217

Execution end: Jul 16, 2022 11:36:45.247951

Test Result: PASS

Evidence: Starts on next line.

```
1 git version 2.37.1
```

5.1.3 Test Case: Configure Git

Purpose: This protocol is used to demonstrate that the Git software has been properly installed and configured. This is done by using Git to configure the .gitconfig file and then printing the results of the changes made to .gitconfig.

Requirement: IUR-000

Step: 1

Confirm: Confirm a developer is able to access Git Help.

Expectation: Git help is displayed.

Command: git help

Execution start: Jul 16, 2022 11:36:45.248613

Execution end: Jul 16, 2022 11:36:45.249364

Test Result: PASS

Evidence: Starts on next line.

```
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
   show      Show various types of objects
   status    Show the working tree status

grow, mark and tweak your common history
   branch    List, create, or delete branches
   commit    Record changes to the repository
   merge     Join two or more development histories together
   rebase    Reapply commits on top of another base tip
   reset     Reset current HEAD to the specified state
   switch    Switch branches
   tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch     Download objects and refs from another repository
   pull      Fetch from and integrate with another repository or a local branch
   push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

Step: 2

Confirm: A developer is able to initialize the repository.

Expectation: No error.

Command: git init

Execution start: Jul 16, 2022 11:36:45.249647

Execution end: Jul 16, 2022 11:36:45.251274

Test Result: PASS

Evidence: Starts on next line.

```
1 Initialized empty Git repository in /home/traap/git/git-client/test-output/tmp/package
    -2/.git/
```

Step: 3

Confirm: A developer is able to configure a Git code pager.

Expectation: No error.

Command: git config –local code.pager cat

Execution start: Jul 16, 2022 11:36:45.251486

Execution end: Jul 16, 2022 11:36:45.252389

Test Result: PASS

Evidence: n/a

Step: 4

Confirm: A developer is able to configure a Git user name.

Expectation: No error.

Command: git config –local user.name "Gary A. Howard"

Execution start: Jul 16, 2022 11:36:45.252604

Execution end: Jul 16, 2022 11:36:45.255105

Test Result: PASS

Evidence: n/a

Step: 5

Confirm: A developer is able to configure a Git user email address.

Expectation: No error.

Command: git config –local user.email gary.a.howard@mac.com

Execution start: Jul 16, 2022 11:36:45.255324

Execution end: Jul 16, 2022 11:36:45.256217

Test Result: PASS

Evidence: n/a

Step: 6

Confirm: A developer is able to configure a Git credentials.

Expectation: No error.

Command: git config –local credential.helper store

Execution start: Jul 16, 2022 11:36:45.256513

Execution end: Jul 16, 2022 11:36:45.257384

Test Result: PASS

Evidence: n/a

Step: 7

Confirm: A developer is able to disable color output to the terminal.

Expectation: No error.

Command: git config –local color.ui false

Execution start: Jul 16, 2022 11:36:45.257596

Execution end: Jul 16, 2022 11:36:45.258433

Test Result: PASS

Evidence: n/a

Step: 8

Confirm: Confirm a developer has a properly configured .gitconfig file.

Expectation: .gitconfig file is displayed.

Command: cat .git/config

Execution start: Jul 16, 2022 11:36:45.258635

Execution end: Jul 16, 2022 11:36:45.259028

Test Result: PASS

Evidence: Starts on next line.

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[code]
  pager = cat
[user]
  name = Gary A. Howard
  email = gary.a.howard@mac.com
[credential]
  helper = store
[color]
  ui = false
```

5.1.4 Test Case: Create a Local Git Repository

Purpose: This test case is used to demonstrate a local Git repository is properly initialized, files are added to a local working copy of the repository causing them to become tracked by Git, tracked files are committed to the local repository, and files not tracked by Git are removed from the Git repository working area.

Requirement: IUR-001, IUR-002, IUR-004, IUR-005, and IUR-006

Step: 1

Confirm: Confirm a developer is able to touch a file named f1.

Expectation: No error.

Command: touch f1

Execution start: Jul 16, 2022 11:36:45.260006

Execution end: Jul 16, 2022 11:36:45.260504

Test Result: PASS

Evidence: n/a

Step: 2

Confirm: Confirm a developer is able to touch a file named f2.

Expectation: No error.

Command: touch f2

Execution start: Jul 16, 2022 11:36:45.261578

Execution end: Jul 16, 2022 11:36:45.262123

Test Result: PASS

Evidence: n/a

Step: 3

Confirm: Confirm a developer is able to list newly touched files.

Expectation: Directory listing has newly touched files.

Command: ls -la

Execution start: Jul 16, 2022 11:36:45.262334

Execution end: Jul 16, 2022 11:36:45.264328

Test Result: PASS

Evidence: Starts on next line.

```
total 8
drwxr-xr-x 2 traap traap 4096 Jul 16 11:36 .
drwxr-xr-x 4 traap traap 4096 Jul 16 11:36 ..
-rw-r--r-- 1 traap traap    0 Jul 16 11:36 f1
-rw-r--r-- 1 traap traap    0 Jul 16 11:36 f2
```

Step: 4

Confirm: Confirm a developer is able to initialize a local repo.

Expectation: No error.

Command: git init

Execution start: Jul 16, 2022 11:36:45.264537

Execution end: Jul 16, 2022 11:36:45.266157

Test Result: PASS

Evidence: Starts on next line.

```
1 Initialized empty Git repository in /home/traap/git/git-client/test-output/tmp/package
      -0/.git/
```

Step: 5

Confirm: A developer is able to disable color output to the terminal.

Expectation: No error.

Command: git config –local color.ui false

Execution start: Jul 16, 2022 11:36:45.266387

Execution end: Jul 16, 2022 11:36:45.267291

Test Result: PASS

Evidence: n/a

Step: 6

Confirm: Git is configured to disable color.

Expectation: .git/config ui section shows color set to true.

Command: cat .git/config

Execution start: Jul 16, 2022 11:36:45.267644

Execution end: Jul 16, 2022 11:36:45.268118

Test Result: PASS

Evidence: Starts on next line.

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[color]
  ui = false
```

Step: 7

Confirm: Confirm a developer is able to add file f1 to a local repo.

Expectation: No error.

Command: git add f1

Execution start: Jul 16, 2022 11:36:45.268318

Execution end: Jul 16, 2022 11:36:45.269393

Test Result: PASS

Evidence: n/a

Step: 8

Confirm: Confirm a developer is able to add file f2 to a local repo.

Expectation: No error.

Command: git add f2

Execution start: Jul 16, 2022 11:36:45.269594

Execution end: Jul 16, 2022 11:36:45.270584

Test Result: PASS

Evidence: n/a

Step: 9

Confirm: Confirm a developer is able to touch a file named f3.

Expectation: No error.

Command: touch f3

Execution start: Jul 16, 2022 11:36:45.270785

Execution end: Jul 16, 2022 11:36:45.271263

Test Result: PASS

Evidence: n/a

Step: 10

Confirm: Confirm a developer is able to touch a file named f4.

Expectation: No error.

Command: touch f4

Execution start: Jul 16, 2022 11:36:45.271522

Execution end: Jul 16, 2022 11:36:45.271982

Test Result: PASS

Evidence: n/a

Step: 11

Confirm: Confirm a developer has obtained a list of tracked and untracked repo changes.

Expectation: Status shows tracked files f1 and f2 and untracked files f3 and f4.

Command: git status

Execution start: Jul 16, 2022 11:36:45.272196

Execution end: Jul 16, 2022 11:36:45.273316

Test Result: PASS

Evidence: Starts on next line.

```
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   f1
  new file:   f2

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  f3
  f4
```

Step: 12

Confirm: Confirm a developer is able to obtain a list of untracked files.

Expectation: Untracked files f3 and f4 are listed.

Command: git clean -n

Execution start: Jul 16, 2022 11:36:45.273533

Execution end: Jul 16, 2022 11:36:45.274413

Test Result: PASS

Evidence: Starts on next line.

```
1 Would remove f3
2 Would remove f4
```

Step: 13

Confirm: Confirm a developer is able to remove untracked files.

Expectation: Untracked files f3 and f4 are removed.

Command: git clean -f

Execution start: Jul 16, 2022 11:36:45.274694

Execution end: Jul 16, 2022 11:36:45.275593

Test Result: PASS

Evidence: Starts on next line.

```
1 Removing f3
2 Removing f4
```

Step: 14

Confirm: Confirm a developer is able to commit files f1 and f2 to the local repo.

Expectation: No errror.

Command: git commit -m "Initial-commit."

Execution start: Jul 16, 2022 11:36:45.275805

Execution end: Jul 16, 2022 11:36:45.279853

Test Result: PASS

Evidence: Starts on next line.

```
[master (root-commit) 8a5b91e] Initial-commit.
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1
 create mode 100644 f2
```

Step: 15

Confirm: Confirm a developer is able determine tracked files have been committed to local repo.

Expectation: The working directory is clean.

Command: git status

Execution start: Jul 16, 2022 11:36:45.280093

Execution end: Jul 16, 2022 11:36:45.281274

Test Result: PASS

Evidence: Starts on next line.

```
On branch master
nothing to commit, working tree clean
```

Step: 16

Confirm: Confirm a developer is able to list commits to local repo.

Expectation: Commit has been processed.

Command: git log –all

Execution start: Jul 16, 2022 11:36:45.281492

Execution end: Jul 16, 2022 11:36:45.282539

Test Result: PASS

Evidence: Starts on next line.

```
commit 8a5b91e1488422a130b9ab6f6eedb265100917f6
Author: Traap <gary.a.howard@mac.com>
Date:   Sat Jul 16 11:36:45 2022 -0500

    Initial-commit.
```

5.1.5 Test Suite: Clone Remote Repository

Purpose: This test suite ensures a developer can clone a remote repository, make changes to it, and revert the changes.

Requirement: IUR-003, IUR-004, IUR-005, IUR-007, and IUR-008

5.1.6 Test Case: Clone Repository

Purpose: This test case demonstrates a developer is able to clone a remote repository and use it as a local repository.

Requirement: IUR-004 and IUR-007

Step: 1

Confirm: A developer is able to clone a remote repository.

Expectation: No error.

Command: git clone git@github.com:Traap/quicksort.git

Execution start: Jul 16, 2022 11:36:45.423604

Execution end: Jul 16, 2022 11:36:46.130609

Test Result: PASS

Evidence: Starts on next line.

Step: 2

Confirm: A developer is able to disable color output to the terminal.

Expectation: No error.

Command: git config –local color.ui false

Execution start: Jul 16, 2022 11:36:46.130977

Execution end: Jul 16, 2022 11:36:46.132175

Test Result: PASS

Evidence: n/a

Step: 3

Confirm: Git is configured to disable color.

Expectation: .git/config ui section shows color set to true.

Command: cat .git/config

Execution start: Jul 16, 2022 11:36:46.132450

Execution end: Jul 16, 2022 11:36:46.133019

Test Result: PASS

Evidence: Starts on next line.

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = git@github.com:Traap/quicksort.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
  rebase = true
[color]
  ui = false
```

Step: 4

Confirm: A developer is working on master branch and up-to-date after cloning a remote repository.

Expectation: Branch information is displayed.

Command: git status

Execution start: Jul 16, 2022 11:36:46.133309

Execution end: Jul 16, 2022 11:36:46.135271

Test Result: PASS

Evidence: Starts on next line.

```
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Step: 5

Confirm: A developer is able to review a set of tracked repositories.

Expectation: Fetch and pull URLs are identical

Command: git remote -v

Execution start: Jul 16, 2022 11:36:46.135682

Execution end: Jul 16, 2022 11:36:46.136796

Test Result: PASS

Evidence: Starts on next line.

```
origin  git@github.com:Traap/quicksort.git (fetch)
origin  git@github.com:Traap/quicksort.git (push)
```

Step: 6

Confirm: A developer's local and remote branches are mapped to master branch.

Expectation: Local and remote branch are mapped to master branch..

Command: git branch -vv

Execution start: Jul 16, 2022 11:36:46.137086

Execution end: Jul 16, 2022 11:36:46.138502

Test Result: PASS

Evidence: Starts on next line.

```
* master 0685fa2 [origin/master] Change all [Z] to [a]
```

5.1.7 Test Case: Revert a merged change

Purpose: This test case demonstrates a developer is able to make a change to a local git repository, push the changes to a remote repository, and then revert the entire change.

Requirement: IUR-003, IUR-004, IUR-005, IUR-007, and IUR-008

Step: 1

Confirm: A developer is able to pull the remote repository to the local repository.

Expectation: No error.

Command: git pull

Execution start: Jul 16, 2022 11:36:46.139567

Execution end: Jul 16, 2022 11:36:46.897149

Test Result: PASS

Evidence: n/a

```
1 Already up to date.
```

Step: 2

Confirm: A developer is able to remove all files from the local repository.

Expectation: No error.

Command: git rm src/Main.hs src/QuickSort.hs

Execution start: Jul 16, 2022 11:36:46.898317

Execution end: Jul 16, 2022 11:36:46.905069

Test Result: PASS

Evidence: n/a

```
rm 'src/Main.hs'
rm 'src/QuickSort.hs'
```

Step: 3

Confirm: A developer is able to commit file removals.

Expectation: No error.

Command: git commit -m "Remove all quicksort files."

Execution start: Jul 16, 2022 11:36:46.906158

Execution end: Jul 16, 2022 11:36:46.928145

Test Result: PASS

Evidence: Starts on next line.

```
[master ead21fe] Remove all quicksort files.
 2 files changed, 60 deletions(-)
 delete mode 100644 src/Main.hs
 delete mode 100644 src/QuickSort.hs
```

Step: 4

Confirm: A developer is able to commit file removals.

Expectation: No error.

Command: git push

Execution start: Jul 16, 2022 11:36:46.929226

Execution end: Jul 16, 2022 11:36:47.735307

Test Result: PASS

Evidence: n/a

Step: 5

Confirm: A developer is able to review the commits to the repository.

Expectation: Update has been processed.

Command: git log –oneline -3

Execution start: Jul 16, 2022 11:36:47.736756

Execution end: Jul 16, 2022 11:36:47.742071

Test Result: PASS

Evidence: Starts on next line.

```
ead21fe Remove all quicksort files.
0685fa2 Change all [Z] to [a]
506540d Change all [a] to [Z]
```

Step: 6

Confirm: A developer is able to pull the remote repository to the local repository.

Expectation: No error.

Command: git pull

Execution start: Jul 16, 2022 11:36:47.743153

Execution end: Jul 16, 2022 11:36:48.369691

Test Result: PASS

Evidence: Starts on next line.

```
1 Already up to date.
```

Step: 7

Confirm: A developer is able ot undo the delete operation.

Expectation: No error.

Command: git revert –no-edit HEAD

Execution start: Jul 16, 2022 11:36:48.370832

Execution end: Jul 16, 2022 11:36:48.380663

Test Result: PASS

Evidence: Starts on next line.

```
[master 672ae01] Revert "Remove all quicksort files."
 Date: Sat Jul 16 11:36:48 2022 -0500
 2 files changed, 60 insertions(+)
 create mode 100644 src/Main.hs
 create mode 100644 src/QuickSort.hs
```

Step: 8

Confirm: A developer is albe to update the remote repository.

Expectation: No error.

Command: git push

Execution start: Jul 16, 2022 11:36:48.385638

Execution end: Jul 16, 2022 11:36:49.216545

Test Result: PASS

Evidence: n/a

Step: 9

Confirm: A developer is able to review the commits to the repository.

Expectation: Update has been processed.

Command: git log –oneline -3

Execution start: Jul 16, 2022 11:36:49.217731

Execution end: Jul 16, 2022 11:36:49.223099

Test Result: PASS

Evidence: Starts on next line.

```
672ae01 Revert "Remove all quicksort files."
ead21fe Remove all quicksort files.
0685fa2 Change all [Z] to [a]
```

5.1.8 Test Suite: Quick Fix to Software

Purpose: This test suite ensures a developer is able to switch working contexts from the master branch to a quick-fix branch to make programming changes.

Requirement: IUR-002, IUR-003, IUR-004, IUR-005, IUR-006, IUR-007, IUR-008, and IUR-009.

5.1.9 Test Case: Create a branch named Quick Fix

Purpose: This test case demonstrates a developer is able to create a new branch to make a programming fix. The new branch is tracked by both the local and remote git repository. The developer is able to make a code change and commit the change to the local git repository and push it to the remote git repository.

Requirement: IUR-002, IUR-003, IUR-004, IUR-005, IUR-006, IUR-007, IUR-008, and IUR-009.

Step: 1

Confirm: A developer is able to download a git repository.

Expectation: No error.

Command: git pull

Execution start: Jul 16, 2022 11:36:49.375871

Execution end: Jul 16, 2022 11:36:50.025787

Test Result: PASS

Evidence: n/a

```
1 Already up to date.
```

Step: 2

Confirm: A developer is able to create a new branch.

Expectation: No error.

Command: git checkout -b quick-fix

Execution start: Jul 16, 2022 11:36:50.026687

Execution end: Jul 16, 2022 11:36:50.031105

Test Result: PASS

Evidence: n/a

Step: 3

Confirm: A developer is able to obtain a listing of all branches.

Expectation: No error.

Command: git branch -a

Execution start: Jul 16, 2022 11:36:50.032072

Execution end: Jul 16, 2022 11:36:50.036730

Test Result: PASS

Evidence: Starts on next line.

```
  master
* quick-fix
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
```

Step: 4

Confirm: A developer is able to make a code change.

Expectation: No error.

Command: sed -i.bak -e s/

a

/

Z

/g QuickSort.hs

Execution start: Jul 16, 2022 11:36:50.038199

Execution end: Jul 16, 2022 11:36:50.055146

Test Result: PASS

Evidence: n/a

Step: 5

Confirm: A developer is able to remove untracked files.

Expectation: No error.

Command: git clean -f

Execution start: Jul 16, 2022 11:36:50.056174

Execution end: Jul 16, 2022 11:36:50.061144

Test Result: PASS

Evidence: Starts on next line.

```
Removing src/QuickSort.hs.bak
```

Step: 6

Confirm: A developer is able to add the changed file to the local repository.

Expectation: No error.

Command: git add .

Execution start: Jul 16, 2022 11:36:50.062241

Execution end: Jul 16, 2022 11:36:50.068731

Test Result: PASS

Evidence: n/a

Step: 7

Confirm: A developer is able to determine the changed file has been added to the tracked changes on the local git repository.

Expectation: Tracked file is listed.

Command: git status

Execution start: Jul 16, 2022 11:36:50.069902

Execution end: Jul 16, 2022 11:36:50.076297

Test Result: PASS

Evidence: Starts on next line.

```
On branch quick-fix
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  modified:   src/QuickSort.hs
```

Step: 8

Confirm: A developer is able to commit the changed file.

Expectation: No error.

Command: git commit -m "Change all [a] to [Z]"

Execution start: Jul 16, 2022 11:36:50.076605

Execution end: Jul 16, 2022 11:36:50.081026

Test Result: PASS

Evidence: Starts on next line.

```
[quick-fix 2e7ffd9] Change all [a] to [Z]
 1 file changed, 25 insertions(+), 25 deletions(-)
```

Step: 9

Confirm: A developer is able to update the remote repository.

Expectation: No error.

Command: git push origin quick-fix

Execution start: Jul 16, 2022 11:36:50.081261

Execution end: Jul 16, 2022 11:36:50.902285

Test Result: PASS

Evidence: n/a

Step: 10

Confirm: A developer is able to establish branch tracking between the remote repository and new branch.

Expectation: Tracking has been established.

Command: git branch -u origin/quick-fix quick-fix

Execution start: Jul 16, 2022 11:36:50.903523

Execution end: Jul 16, 2022 11:36:50.909414

Test Result: PASS

Evidence: Starts on next line.

```
1 branch 'quick-fix' set up to track 'origin/quick-fix' by rebasing.
```

Step: 11

Confirm: A developer is able to update the remote repository.

Expectation: No error.

Command: git push

Execution start: Jul 16, 2022 11:36:50.909625

Execution end: Jul 16, 2022 11:36:51.452771

Test Result: PASS

Evidence: n/a

Step: 12

Confirm: A developer is able to view difference between master and quick-fix branches.

Expectation: No error.

Command: git diff master..quick-fix

Execution start: Jul 16, 2022 11:36:51.453990

Execution end: Jul 16, 2022 11:36:51.460228

Test Result: PASS

Evidence: Starts on next line.

```
diff --git a/src/QuickSort.hs b/src/QuickSort.hs
index 286a966..7daee5d 100644
--- a/src/QuickSort.hs
+++ b/src/QuickSort.hs
@@ -1,45 +1,45 @@
 module QuickSort(
    quicksort
    , isInOrder
-   , h[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]meElements
+   , h[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]meElements
    ) where

-import D[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]t[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]].List (p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]rtition)
+import D[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]t[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]].List (p[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]rtition)

--- | Put [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] list of into [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]scending order
-quicksort :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]
-quicksort xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
+--- | Put [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]] list of into [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]scending order
+quicksort :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]
+quicksort xs = c[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]se xs of
     []     -> []
     (y:ys) -> let
-       (ls,rs) = p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]rtition (<y) ys
+       (ls,rs) = p[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]rtition (<y) ys
        in (quicksort ls) ++ y:(quicksort rs)

--- | Test to see if [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list is in order
-isInOrder :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> Bool
-isInOrder xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
-   [] -> True -- trivi[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]lly true
-   (x:ys) -> c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se ys of
-       [] -> True -- singul[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]r lists[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]re sorted
+--- | Test to see if [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]] list is in order
+isInOrder :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+isInOrder xs = c[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]se xs of
+   [] -> True -- trivi[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]lly true
+   (x:ys) -> c[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]se ys of
+       [] -> True -- singul[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]r lists[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]re sorted
```

```
40          (y:_) -> x<=y && isInOrder ys
41
42 ---- | Test if two lists h[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ve the s
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me elements
43 -h[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]]meElements :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]], Eq [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) =>
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
44 -h[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]]meElements xs ys
45 -    = ( xs `isSubB[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l` ys)
46 -    && (ys `isSubB[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l` xs)
47 +---- | Test if two lists h[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]ve the s
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]me elements
48 +h[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[Z
       ]]]]]]]]]]]]]]]]]]]]]]]]meElements :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[Z
       ]]]]]]]]]]]]]]]]]]]]]]]], Eq [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]])
      => [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]] ->
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
49 +h[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[Z
       ]]]]]]]]]]]]]]]]]]]]]]]]meElements xs ys
50 +    = ( xs `isSubB[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]l` ys)
51 +    && (ys `isSubB[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]l` xs)
52
53 ---- | Test th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]t the multiplicity of e
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ch element in the
54 ---- first list is less th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n or equ
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l to the multiplicity
55 ---- of the s[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me element in the second
       list.
56 -isSubB[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu[[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]]l :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]],
        Eq [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] ->
       Bool
57 -isSubB[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu[[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]]l xs ys = [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]
       ll (\ x -> let
58 +---- | Test th[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]t the multiplicity of e
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]ch element in the
59 +---- first list is less th[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]n or equ
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]l to the multiplicity
60 +---- of the s[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]me element in the second
       list.
61 +isSubB[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]l :: (Ord
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]], Eq [[[[[[[[[[[[[[[[[[[[[[[[Z
       ]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]
      -> [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
62 +isSubB[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]l xs ys =
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]ll (\ x -> let
63      mulxs = multiplicity x xs
64      mulys = multiplicity x ys
65      in mulxs <= mulys ) xs
66
67 ---- |  Count the number of times th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]t
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n element occurs
68 ---- within [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list.
69 -multiplicity :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]], Eq
       [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
       ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] ->
       Int
70 +---- |  Count the number of times th[[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]t
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]n element occurs
71 +---- within [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]] list.
72 +multiplicity :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]], Eq
       [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[Z
```

```
    ]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[Z]]]]]]]]]]]]]]]]]]]]
    -> Int
multiplicity _ [] = 0
multiplicity x (y:ys)
    | x==y  = 1 + multiplicity x ys
```

Step: 13

Confirm: A developer is able to verify branches have been linked.

Expectation: Branch linkage is established.

Command: git branch -vv

Execution start: Jul 16, 2022 11:36:51.461438

Execution end: Jul 16, 2022 11:36:51.467300

Test Result: PASS

Evidence: Starts on next line.

```
  master    672ae01 [origin/master] Revert "Remove all quicksort files."
* quick-fix 2e7ffd9 [origin/quick-fix] Change all [a] to [Z]
```

Step: 14

Confirm: A developer is able to review commits made to repository.

Expectation: Update has been processed.

Command: git log –oneline -3

Execution start: Jul 16, 2022 11:36:51.468418

Execution end: Jul 16, 2022 11:36:51.469602

Test Result: PASS

Evidence: Starts on next line.

```
2e7ffd9 Change all [a] to [Z]
672ae01 Revert "Remove all quicksort files."
ead21fe Remove all quicksort files.
```

5.1.10 Test Case: Undo a program change made on branch Quick Fix.

Purpose: This test case demonstrates a developers is able to checkout a branch named Quick Fix. The developer is able to reverse the programming change and commit the change to the local git repository and push it to the remote git repository.

Requirement: IUR-003, IUR-004, IUR-005, IUR-006, IUR-007, IUR-008, and IUR-009.

Step: 1

Confirm: A developer is able to switch to the master branch.

Expectation: No error.

Command: git checkout master

Execution start: Jul 16, 2022 11:36:51.470505

Execution end: Jul 16, 2022 11:36:51.472212

Test Result: PASS

Evidence: Starts on next line.

```
1 Your branch is up to date with 'origin/master'.
```

Step: 2

Confirm: A developer is able refresh their local repository after a remote repository has been downloaded.

Expectation: No error.

Command: git pull

Execution start: Jul 16, 2022 11:36:51.472546

Execution end: Jul 16, 2022 11:36:52.132002

Test Result: PASS

Evidence: Starts on next line.

```
1 Already up to date.
```

Step: 3

Confirm: A developer is able to create a new branch to work on a programming assignment.

Expectation: No error.

Command: git checkout quick-fix

Execution start: Jul 16, 2022 11:36:52.132272

Execution end: Jul 16, 2022 11:36:52.133860

Test Result: PASS

Evidence: Starts on next line.

```
1 Your branch is up to date with 'origin/quick-fix'.
```

Step: 4

Confirm: A developer is able to establish branch tracking between the local and remote branches.

Expectation: No error.

Command: git branch -u origin/quick-fix quick-fix

Execution start: Jul 16, 2022 11:36:52.134114

Execution end: Jul 16, 2022 11:36:52.135409

Test Result: PASS

Evidence: Starts on next line.

```
1 branch 'quick-fix' set up to track 'origin/quick-fix' by rebasing.
```

Step: 5

Confirm: A developer is able to list all branches.

Expectation: No error.

Command: git branch -a

Execution start: Jul 16, 2022 11:36:52.135624

Execution end: Jul 16, 2022 11:36:52.136664

Test Result: PASS

Evidence: Starts on next line.

```
  master
* quick-fix
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/quick-fix
```

Step: 6

Confirm: A developer is able to make a software change.

Expectation: No error.

Command: sed -i.bak -e s/

Z

/

a

/g QuickSort.hs

Execution start: Jul 16, 2022 11:36:52.136959

Execution end: Jul 16, 2022 11:36:52.140157

Test Result: PASS

Evidence: Starts on next line.

Step: 7

Confirm: A developer is able to remove files that are not being tracked by the local repository.

Expectation: No error.

Command: git clean -f

Execution start: Jul 16, 2022 11:36:52.140401

Execution end: Jul 16, 2022 11:36:52.141439

Test Result: PASS

Evidence: Starts on next line.

```
1 Removing src/QuickSort.hs.bak
```

Step: 8

Confirm: A developer is able to add changed files to the local repository.

Expectation: No error.

Command: git add .

Execution start: Jul 16, 2022 11:36:52.141660

Execution end: Jul 16, 2022 11:36:52.142963

Test Result: PASS

Evidence: Starts on next line.

Step: 9

Confirm: A developer is able to determine their change has been added to the local repository.

Expectation: File changes are being tracked by local repository.

Command: git status

Execution start: Jul 16, 2022 11:36:52.143274

Execution end: Jul 16, 2022 11:36:52.144703

Test Result: PASS

Evidence: Starts on next line.

```
On branch quick-fix
Your branch is up to date with 'origin/quick-fix'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  modified:   src/QuickSort.hs
```

Step: 10

Confirm: A developer is able to commit change to the local repository.

Expectation: No error.

Command: git commit -m "Change all [Z] to [a]"

Execution start: Jul 16, 2022 11:36:52.144941

Execution end: Jul 16, 2022 11:36:52.149529

Test Result: PASS

Evidence: Starts on next line.

```
[quick-fix 20d41ae] Change all [Z] to [a]
 1 file changed, 25 insertions(+), 25 deletions(-)
```

Step: 11

Confirm: A developer is able to update a remote Git repository based on changes made to the local repository.

Expectation: No error.

Command: git push

Execution start: Jul 16, 2022 11:36:52.149797

Execution end: Jul 16, 2022 11:36:53.060184

Test Result: PASS

Evidence: Starts on next line.

Step: 12

Confirm: A developer is able to review the differences between their master branch and quick-fix branch.

Expectation: Differences are displayed.

Command: git diff master..quick-fix

Execution start: Jul 16, 2022 11:36:53.061708

Execution end: Jul 16, 2022 11:36:53.067366

Test Result: PASS

Evidence: Starts on next line.

```
diff --git a/src/QuickSort.hs b/src/QuickSort.hs
index 286a966..2a073aa 100644
--- a/src/QuickSort.hs
+++ b/src/QuickSort.hs
@@ -1,45 +1,45 @@
 module QuickSort(
    quicksort
    , isInOrder
-   , h[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]]]meElements
+   , h[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[[
    a]]]]]]]]]]]]]]]]]]]]]]]]]meElements
    ) where

-import D[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]t[[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]].List (p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]
    rtition)
+import D[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]t
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]].List (p
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]rtition)

-- | Put [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] list of into
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]scending order
-quicksort :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]]]
-quicksort xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
+-- | Put [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] list of into
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]scending order
+quicksort :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] ->
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]
+quicksort xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
    []     -> []
    (y:ys) -> let
-       (ls,rs) = p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]rtition (<y) ys
+       (ls,rs) = p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]rtition (<y) ys
        in (quicksort ls) ++ y:(quicksort rs)

-- | Test to see if [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list is in order
-isInOrder :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> Bool
-isInOrder xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
-   [] -> True -- trivi[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]lly true
-   (x:ys) -> c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se ys of
-       [] -> True -- singul[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]r lists
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]re sorted
+-- | Test to see if [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list is in
    order
+isInOrder :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+isInOrder xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
+   [] -> True -- trivi[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]lly true
+   (x:ys) -> c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se ys of
```

```
+        [] -> True -- singul[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]r lists
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]re sorted
         (y:_) -> x<=y && isInOrder ys

-- | Test if two lists h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ve the s
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me elements
-h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] , Eq [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]) =>
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
-h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements xs ys
-    = ( xs `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` ys)
-    && (ys `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` xs)
+-- | Test if two lists h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ve the s
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me elements
+h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] , Eq [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements xs ys
+    = ( xs `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` ys)
+    && (ys `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` xs)

-- | Test th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]t the multiplicity of e
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ch element in the
-- first list is less th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n or equ
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l to the multiplicity
-- of the s[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me element in the second
     list.
-isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]l :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]] ,
      Eq [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]] ->
     Bool
-isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]l xs ys = [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]
     ll (\ x -> let
+-- | Test th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]t the multiplicity of
     e[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ch element in the
+-- first list is less th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n or equ
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l to the multiplicity
+-- of the s[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me element in the
     second list.
+isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l :: (Ord
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]] , Eq [[[[[[[[[[[[[[[[[[[[[[[[
     a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l xs ys =
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ll (\ x -> let
       mulxs = multiplicity x xs
       mulys = multiplicity x ys
       in mulxs <= mulys ) xs

-- |  Count the number of times th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]t
      [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n element occurs
-- within [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list.
-multiplicity :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] , Eq
      [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] ->
     Int
+-- |  Count the number of times th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]
```

```
    t [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n element occurs
+-- within [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list.
+multiplicity :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]], Eq
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[
    a]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]] -> Int
 multiplicity _ [] = 0
 multiplicity x (y:ys)
    | x==y  = 1 + multiplicity x ys
```

Step: 13

Confirm: A developer is able to determine their new branch is tracked independently of the master branch.

Expectation: Branch tracking is displayed.

Command: git branch -vv

Execution start: Jul 16, 2022 11:36:53.067602

Execution end: Jul 16, 2022 11:36:53.068795

Test Result: PASS

Evidence: Starts on next line.

```
  master    672ae01 [origin/master] Revert "Remove all quicksort files."
* quick-fix 20d41ae [origin/quick-fix] Change all [Z] to [a]
```

Step: 14

Confirm: A developer is able to review a list of changes made to their repository.

Expectation: Commits are displayed.

Command: git log –oneline -3

Execution start: Jul 16, 2022 11:36:53.069013

Execution end: Jul 16, 2022 11:36:53.070077

Test Result: PASS

Evidence: Starts on next line.

```
20d41ae Change all [Z] to [a]
2e7ffd9 Change all [a] to [Z]
672ae01 Revert "Remove all quicksort files."
```

5.1.11 Test Case: Merge a programming change made on branch named Quick Fix

Purpose: This test case demonstrates a developer's ability to merge a code change from a working branch to the master branch that the team treats as their central repository makes the code available to others.

Requirement: IUR-002, IUR-003, IUR-004, IUR-006, IUR-007, IUR-008, and IUR-009.

Step: 1

Confirm: A developer is able to checkout the master branch to the local repository.

Expectation: No error.

Command: git checkout master

Execution start: Jul 16, 2022 11:36:53.071028

Execution end: Jul 16, 2022 11:36:53.072710

Test Result: PASS

Evidence: Starts on next line.

```
1 Your branch is up to date with 'origin/master'.
```

Step: 2

Confirm: A developer is able to update the local copy of their master branch.

Expectation: No error.

Command: git pull

Execution start: Jul 16, 2022 11:36:53.072928

Execution end: Jul 16, 2022 11:36:53.684249

Test Result: PASS

Evidence: n/a

```
1 Already up to date.
```

Step: 3

Confirm: A developer is able to checkout a quick-fix branch to their local repository.

Expectation: No error.

Command: git checkout quick-fix

Execution start: Jul 16, 2022 11:36:53.685146

Execution end: Jul 16, 2022 11:36:53.688887

Test Result: PASS

Evidence: Starts on next line.

```
1 Your branch is up to date with 'origin/quick-fix'.
```

Step: 4

Confirm: A developer is able to update the local copy of their quick-bix branch.

Expectation: No error.

Command: git pull

Execution start: Jul 16, 2022 11:36:53.689406

Execution end: Jul 16, 2022 11:36:54.366385

Test Result: PASS

Evidence: n/a

```
1 Already up to date.
```

Step: 5

Confirm: A developer is able to return to their master branch.

Expectation: No error.

Command: git checkout master

Execution start: Jul 16, 2022 11:36:54.367526

Execution end: Jul 16, 2022 11:36:54.375617

Test Result: PASS

Evidence: Starts on next line.

```
1 Your branch is up to date with 'origin/master'.
```

Step: 6

Confirm: A developer is able to review the differences between their master branch and quick-fix branch.

Expectation: Differences between master and quick-fix branch are listed.

Command: git diff master..quick-fix

Execution start: Jul 16, 2022 11:36:54.376894

Execution end: Jul 16, 2022 11:36:54.383017

Test Result: PASS

Evidence: Starts on next line.

```
diff --git a/src/QuickSort.hs b/src/QuickSort.hs
index 286a966..2a073aa 100644
--- a/src/QuickSort.hs
+++ b/src/QuickSort.hs
@@ -1,45 +1,45 @@
 module QuickSort(
    quicksort
    , isInOrder
-   , h[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]]]meElements
+   , h[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[[
    a]]]]]]]]]]]]]]]]]]]]]]]]]meElements
    ) where

-import D[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]t[[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]].List (p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]
    rtition)
+import D[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]t
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]].List (p
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]rtition)

--- | Put [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] list of into
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]scending order
-quicksort :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]]]
-quicksort xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
+--- | Put [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] list of into
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]scending order
+quicksort :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] ->
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]
+quicksort xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
    []     -> []
    (y:ys) -> let
-       (ls,rs) = p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]rtition (<y) ys
+       (ls,rs) = p[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]rtition (<y) ys
        in (quicksort ls) ++ y:(quicksort rs)

--- | Test to see if [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list is in order
-isInOrder :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> Bool
-isInOrder xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
-   [] -> True -- trivi[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]lly true
-   (x:ys) -> c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se ys of
-       [] -> True -- singul[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]r lists
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]re sorted
+--- | Test to see if [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list is in
    order
+isInOrder :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]) =>
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+isInOrder xs = c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se xs of
+   [] -> True -- trivi[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]lly true
+   (x:ys) -> c[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]se ys of
```

```
+        [] -> True -- singul[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]r lists
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]re sorted
         (y:_) -> x<=y && isInOrder ys

-- | Test if two lists h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ve the s
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me elements
-h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] , Eq [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]) =>
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
-h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements xs ys
-    = ( xs `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` ys)
-    && (ys `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` xs)
+-- | Test if two lists h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ve the s
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me elements
+h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] , Eq [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+h[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]sS[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]meElements xs ys
+    = ( xs `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` ys)
+    && (ys `isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]l` xs)

-- | Test th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]t the multiplicity of e
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ch element in the
-- first list is less th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n or equ
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l to the multiplicity
-- of the s[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me element in the second
     list.
-isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]l :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]] ,
      Eq [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]] ->
     Bool
-isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu[[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]]l xs ys = [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]]
     ll (\ x -> let
+-- | Test th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]t the multiplicity of
     e[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ch element in the
+-- first list is less th[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n or equ
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l to the multiplicity
+-- of the s[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]me element in the
     second list.
+isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l :: (Ord
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] , Eq [[[[[[[[[[[[[[[[[[[[[[[[
     a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
     ]]]]]]]]]]]]]]]]]]]]]]]] -> Bool
+isSubB[[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]gOrEqu
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]l xs ys =
      [[[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]ll (\ x -> let
       mulxs = multiplicity x xs
       mulys = multiplicity x ys
       in mulxs <= mulys ) xs

-- |  Count the number of times th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]t
      [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n element occurs
-- within [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list.
-multiplicity :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] , Eq
      [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[a
      ]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] ->
     Int
+-- |  Count the number of times th[[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]
```

```
    t [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]n element occurs
+-- within [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]] list.
+multiplicity :: (Ord [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]], Eq
    [[[[[[[[[[[[[[[[[[[[[[[[a]]]]]]]]]]]]]]]]]]]]]]]]) => [[[[[[[[[[[[[[[[[[[[[[[[
    a]]]]]]]]]]]]]]]]]]]]]]]] -> [[[[[[[[[[[[[[[[[[[[[[[[a
    ]]]]]]]]]]]]]]]]]]]]]]]] -> Int
 multiplicity _ [] = 0
 multiplicity x (y:ys)
    | x==y  = 1 + multiplicity x ys
```

Step: 7

Confirm: A developer is able to determine the quick-fix branch has not been merged with the master branch.

Expectation: Quick-fix branch has not been merged.

Command: git branch –merged

Execution start: Jul 16, 2022 11:36:54.384751

Execution end: Jul 16, 2022 11:36:54.395161

Test Result: PASS

Evidence: Starts on next line.

```
1 * master
```

Step: 8

Confirm: A developer is able to merge the quick-fix branch into the master branch.

Expectation: No error.

Command: git merge quick-fix

Execution start: Jul 16, 2022 11:36:54.395838

Execution end: Jul 16, 2022 11:36:54.409730

Test Result: PASS

Evidence: Starts on next line.

```
Updating 672ae01..20d41ae
Fast-forward
 src/QuickSort.hs | 50 ++++++++++++++++++++++++++-------------------------
 1 file changed, 25 insertions(+), 25 deletions(-)
```

Step: 9

Confirm: A developer is able to update the remote repository with the results of the merge operation.

Expectation: No error.

Command: git push

Execution start: Jul 16, 2022 11:36:54.410208

Execution end: Jul 16, 2022 11:36:55.313006

Test Result: PASS

Evidence: Starts on next line.

Step: 10

Confirm: A developer is able to review the results of the merge operation.

Expectation: Quick-fix branch has been merged.

Command: git branch –merged

Execution start: Jul 16, 2022 11:36:55.314494

Execution end: Jul 16, 2022 11:36:55.320214

Test Result: PASS

Evidence: Starts on next line.

```
* master
  quick-fix
```

Step: 11

Confirm: A developer is able to delete the quick-fix branch from the local repository.

Expectation: Local quick-fix branch has been removed.

Command: git branch –delete quick-fix

Execution start: Jul 16, 2022 11:36:55.320450

Execution end: Jul 16, 2022 11:36:55.321647

Test Result: PASS

Evidence: Starts on next line.

```
1 Deleted branch quick-fix (was 20d41ae).
```

Step: 12

Confirm: A developer is able to delete the quick-fix branch from the remote repository.

Expectation: No error.

Command: git push origin –delete quick-fix

Execution start: Jul 16, 2022 11:36:55.321883

Execution end: Jul 16, 2022 11:36:56.340624

Test Result: PASS

Evidence: Starts on next line.

Step: 13

Confirm: A developer is able to determine the remote and local quick-fix branches have been deleted.

Expectation: No error.

Command: git branch -vv

Execution start: Jul 16, 2022 11:36:56.341858

Execution end: Jul 16, 2022 11:36:56.348098

Test Result: PASS

Evidence: Starts on next line.

```
* master 20d41ae [origin/master] Change all [Z] to [a]
```

Step: 14

Confirm: A developer is able to review a list of changes made to the repository.

Expectation: Commits are listed.

Command: git log –oneline -3

Execution start: Jul 16, 2022 11:36:56.349317

Execution end: Jul 16, 2022 11:36:56.355056

Test Result: PASS

Evidence: Starts on next line.

```
20d41ae Change all [Z] to [a]
2e7ffd9 Change all [a] to [Z]
672ae01 Revert "Remove all quicksort files."
```

5.1.12 Test Suite: Git Tagging

Purpose: This test suite ensures a developer is able to tag git-commits, create new branches based on git-tags, and perform new work on a new branch.

Requirement: IUR-002, IUR-003, IUR-004, IUR-006, IUR-007, IUR-008, and IUR-009.

5.1.13 Test Case: Tag Commits

Purpose: This test case is used to demonstrate Git's ability to tag a commit. A Git repository has been pre-seeded with the following tags.

Tag	Git Checksum
v1.0	bcde6bf1ed2841f39e719ccd323df4cc33497e04
v1.1	b64d30b59ff026f12350a5c040d779a5a9a59214
v1.2	8aa0308a4b661e97df4002ccd7f0ee3dd2f55994
v2.0	ea06210d6f84c2daf26f00c39643a9cd9b98a4a5
v2.1	440ba8b56feb5941783b9cb57ed192391cd251c6
v2.2	c47279b87ed774137929088182l5f53364567bb1

A Git Checksum is unique to a Git installation and are listed here as examples. In the event these tags do not exist, the following commands are used to recreate them and publish them to the git server.

git tag v1.0 [git-checksum]

where git-checksum is the git-commit-id you are associating the tags with. Git-commit-id's are globally-unique.

git push origin –tags

Requirement: IUR-002, IUR-003, IUR-004, IUR-006, IUR-007, IUR-008, and IUR-009.

Step: 1

Confirm: Pre-existing tags are present.

Expectation: Pre-existing tags are listed without error.

Command: git tag

Execution start: Jul 16, 2022 11:36:56.531505

Execution end: Jul 16, 2022 11:36:56.532556

Test Result: PASS

Evidence: Starts on next line.

```
v1.0
v1.1
v1.2
v2.0
v2.1
v2.2
```

Step: 2

Confirm: Pre-existing git-commits are present.

Expectation: Git-commit history is shown without error.

Command: git log –graph –pretty=format:"%h (%cr) <%cn> %d" –simplify-by-decoration

Execution start: Jul 16, 2022 11:36:56.532861

Execution end: Jul 16, 2022 11:36:56.535951

Test Result: PASS

Evidence: Starts on next line.

```
* 20d41ae (4 seconds ago) <Traap>  (HEAD -> master, origin/master, origin/HEAD)
* 49bc0d6 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.2)
* 77d6cf4 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.1)
* f9f3744 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.0)
* 3b57cb3 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.2)
* 6ed6849 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.1)
* f2433b0 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.0)
* 7fefb60 (4 years, 4 months ago) <Gary A. Howard>
```

Step: 3

Confirm: A complete list of files and revision information for v2.0 is listed.

Expectation: All files associated with v2.0 are listed.

Command: git ls-tree –full-tree -r –name-only v2.0

Execution start: Jul 16, 2022 11:36:56.536196

Execution end: Jul 16, 2022 11:36:56.537181

Test Result: PASS

Evidence: Starts on next line.

```
CODE_OF_CONDUCT.md
LICENSE
README.md
quicksort.cabal
src/Main.hs
src/QuickSort.hs
test/Main.hs
```

Step: 4

Confirm: New branch b2.0 is created from tag v2.0.

Expectation: Switch to a new branch "b2.0" is displayed.

Command: git checkout -b b2.0 v2.0

Execution start: Jul 16, 2022 11:36:56.537409

Execution end: Jul 16, 2022 11:36:56.539003

Test Result: PASS

Evidence: Starts on next line.

Step: 5

Confirm: New branch b1.1 is created from tag v1.1.

Expectation: Switch to a new branch "b1.1" is displayed.

Command: git checkout -b b1.1 v1.1

Execution start: Jul 16, 2022 11:36:56.539290

Execution end: Jul 16, 2022 11:36:56.540772

Test Result: PASS

Evidence: Starts on next line.

Step: 6

Confirm: (HEAD, tag: v1.1, b1.1) is displayed for first git-commit.

Expectation: Git-commit history shows (HEAD, tag: v1.1, b1.1) at first commit.

Command: git log –graph –pretty=format:"%h (%cr) <%cn> %d" –simplify-by-decoration

Execution start: Jul 16, 2022 11:36:56.541087

Execution end: Jul 16, 2022 11:36:56.543922

Test Result: PASS

Evidence: Starts on next line.

```
* 6ed6849 (4 years, 4 months ago) <Gary A. Howard>  (HEAD -> b1.1, tag: v1.1)
* f2433b0 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.0)
* 7fefb60 (4 years, 4 months ago) <Gary A. Howard>
```

Step: 7

Confirm: Branches b1.1, b2.0 and master exist.

Expectation: Branches b1.1, b2.0 and master are listed. More branches is okay.

Command: git branch -v

Execution start: Jul 16, 2022 11:36:56.544166

Execution end: Jul 16, 2022 11:36:56.545270

Test Result: PASS

Evidence: Starts on next line.

```
* b1.1   6ed6849 Change all [Z] to [a]
  b2.0   f9f3744 Change all [Z] to [a]
  master 20d41ae Change all [Z] to [a]
```

Step: 8

Confirm: Switch to master branch.

Expectation: Switch to branch "master" is displayed.

Command: git checkout master

Execution start: Jul 16, 2022 11:36:56.545494

Execution end: Jul 16, 2022 11:36:56.547159

Test Result: PASS

Evidence: Starts on next line.

```
1 Your branch is up to date with 'origin/master'.
```

Step: 9

Confirm: Master branch is not tagged with v3.0.

Expectation: At minimum, "(HEAD, origin/master, origin/HEAD, master)" is displayed for first commit.

Command: git log –graph –pretty=format:"%h (%cr) <%cn> %d" –simplify-by-decoration

Execution start: Jul 16, 2022 11:36:56.547531

Execution end: Jul 16, 2022 11:36:56.550650

Test Result: PASS

Evidence: Starts on next line.

```
* 20d41ae (4 seconds ago) <Traap>  (HEAD -> master, origin/master, origin/HEAD)
* 49bc0d6 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.2)
* 77d6cf4 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.1)
* f9f3744 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.0, b2.0)
* 3b57cb3 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.2)
* 6ed6849 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.1, b1.1)
* f2433b0 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.0)
* 7fefb60 (4 years, 4 months ago) <Gary A. Howard>
```

Step: 10

Confirm: Add tag "v3.0" is added to HEAD.

Expectation: No error.

Command: git tag v3.0

Execution start: Jul 16, 2022 11:36:56.550909

Execution end: Jul 16, 2022 11:36:56.551950

Test Result: PASS

Evidence: Starts on next line.

Step: 11

Confirm: Tag "v3.0" is added to HEAD.

Expectation: At minimum, '(HEAD, tag: v3.0, origin/master, origin/HEAD, master)' is displayed for first commit.

Command: git log –graph –pretty=format:"%h (%cr) <%cn> %d" –simplify-by-decoration

Execution start: Jul 16, 2022 11:36:56.552231

Execution end: Jul 16, 2022 11:36:56.555346

Test Result: PASS

Evidence: Starts on next line.

```
* 20d41ae (4 seconds ago) <Traap>  (HEAD -> master, tag: v3.0, origin/master, origin/
  HEAD)
* 49bc0d6 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.2)
* 77d6cf4 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.1)
* f9f3744 (4 years, 4 months ago) <Gary A. Howard>  (tag: v2.0, b2.0)
* 3b57cb3 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.2)
* 6ed6849 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.1, b1.1)
* f2433b0 (4 years, 4 months ago) <Gary A. Howard>  (tag: v1.0)
* 7fefb60 (4 years, 4 months ago) <Gary A. Howard>
```

Step: 12

Confirm: Tag "v3.0" is deleted.

Expectation: No error.

Command: git tag -d v3.0

Execution start: Jul 16, 2022 11:36:56.556487

Execution end: Jul 16, 2022 11:36:56.557565

Test Result: PASS

Evidence: Starts on next line.

```
1 Deleted tag 'v3.0' (was 20d41ae)
```

Step: 13

Confirm: Branch "b1.1" is deleted.

Expectation: No error.

Command: git branch -d b1.1

Execution start: Jul 16, 2022 11:36:56.557819

Execution end: Jul 16, 2022 11:36:56.559257

Test Result: PASS

Evidence: Starts on next line.

```
1 Deleted branch b1.1 (was 6ed6849).
```

Step: 14

Confirm: Branch "b2.0" is deleted.

Expectation: No error.

Command: git branch -d b2.0

Execution start: Jul 16, 2022 11:36:56.559479

Execution end: Jul 16, 2022 11:36:56.561156

Test Result: PASS

Evidence: Starts on next line.

```
1 Deleted branch b2.0 (was f9f3744).
```

Step: 15

Confirm: Master branch remains.

Expectation: No error.

Command: git branch -v

Execution start: Jul 16, 2022 11:36:56.561378

Execution end: Jul 16, 2022 11:36:56.562450

Test Result: PASS

Evidence: Starts on next line.

```
1 * master 20d41ae Change all [Z] to [a]
```

6 Configuration Item Conclusion

This Report Package has satisfied the IUR for the Configuration Item described herein thus the Configuration Item is considered validated for its intended use.

Change Summary

Change	Justification
v1.0.0 Initial version.	New document.
v1.0.1 1. Corrected gramitical errors throughout document. 2. Added test case to demonstrate advanced git branching and tagging.	Process engineering feedback from Bitbucket pull-request