The Plan

Specifically: The overall testing plan for the CoApp project.

Initial author: Timothy Rogers

Contents

[1 General Overview 3](#_Toc296414312)

[1.1 What needs tested? 3](#_Toc296414313)

[1.1.1 Individual Developer Tools 3](#_Toc296414314)

[1.1.2 End-User Tools 3](#_Toc296414315)

[1.1.3 Dev-side full-system 3](#_Toc296414316)

[1.2 How should we be testing? 3](#_Toc296414317)

[1.2.1 Automation! 3](#_Toc296414318)

[1.2.2 Unit tests / Code coverage 3](#_Toc296414319)

[1.2.3 Integration/Scope 3](#_Toc296414320)

[1.2.4 Full system 3](#_Toc296414321)

[1.3 Expectations 4](#_Toc296414322)

[1.3.1 Initial release 4](#_Toc296414323)

[1.3.2 Final release 4](#_Toc296414324)

[1.4 Expected Problems 4](#_Toc296414325)

[1.4.1 Documentation/Specifications 4](#_Toc296414326)

[2 Specifics 4](#_Toc296414327)

# General Overview

## What needs tested?

### Individual Developer Tools

### End-User Tools

#### Must be able to view, select, install, update, and remove packages. All of these should be accomplished in as few steps (keystrokes/clicks) as possible.

### Dev-side full-system

#### This requires that test runs be performed, on multiple packages, from the initial state of just the package source code. The results of these test runs should be fully compiled and processed packages. A complete package creation cycle should require as few steps for the dev as possible.

## How should we be testing?

### Automation!

#### Ideally, once all tests are written, all actual testing can be initiated by batch scripting and timers. Manpower is extremely limited for this project, so all tests should be designed to be operated and collated by a few as a single individual.

### Unit tests / Code coverage

#### All public scope components of all tools in the project should have rigorous unit tests built to test both success and failure conditions. All unit tests, by completion, should average not less than 90% code coverage of the tools and methods being tested.

### Integration/Scope

#### For each combination of discrete tools which are expected to be capable of communicating with each other, a dry-run test of each conceivable scenario class for interaction should be produced.

### Full system

#### Once all of the Dev tools are in a functionally usable state, a test must be run for each package expected to be available (at least for initial release), each beginning from a blank slate with CoApp not yet installed on the test system, and each ending with the package having been successfully created and CoApp having been (cleanly!) removed from the system.

#### Once all client-side components are in a functional state, tests should be built which will start from a bare system (no CoApp), install CoApp (by any of a series of available means), install the major packages, and confirm proper function of the installed packages.

#### After the above client-side tests become possible, additional tests must be produced to confirm that installed software can be properly updated, removed, or that appropriate notice is provided when any of the above cannot be accomplished.

#### Again, after the above client-side tests are possible, additional tests must be generated to confirm that CoApp can be safely and cleanly removed, and that the removal of CoApp does not adversely affect ANY installed packages on the system.

## Expectations

### Initial release

#### It is expected that all dev tools will function properly under valid inputs. All command-line switches should operate correctly, both individually and in combination with all other valid switches.

#### It is expected that the end-user will be able to install and open CoApp without incident. The user should be able to connect to a package repository/library, install a package, update that package (and related sub-packages) and uninstall a package without error.

### Final release

#### It is expected that all dev tools will perform as in the initial release and, in addition, will fail gracefully and meaningfully on invalid inputs.

## Expected Problems

### Documentation/Specifications

#### At present, there do not appear to be complete and meaningful specifications for any tool or API presently in the project.

#### The speed at which the rest of the testing plan will progress will likely be significantly reduced by the necessity of building documentation for the tools and methods on-the-fly.

# Specifics

#### There are no specifics available at this time.