The Plan

Specifically: The overall testing plan for the CoApp project.

Initial author: Timothy Rogers ([tiroger@microsoft.com](mailto:tiroger@microsoft.com))

Contents

[1 General Overview 3](#_Toc296585980)

[1.1 What needs tested? 3](#_Toc296585981)

[1.1.1 Individual Developer Tools 3](#_Toc296585982)

[1.1.2 End-User Tools 3](#_Toc296585983)

[1.1.3 Dev-side full-system 3](#_Toc296585984)

[1.2 How should we be testing? 3](#_Toc296585985)

[1.2.1 Automation! 3](#_Toc296585986)

[1.2.2 Unit tests / Code coverage 3](#_Toc296585987)

[1.2.3 Integration/Scope 3](#_Toc296585988)

[1.2.4 Full system 3](#_Toc296585989)

[1.3 Expectations 4](#_Toc296585990)

[1.3.1 Initial release 4](#_Toc296585991)

[1.3.2 Final release 4](#_Toc296585992)

[1.4 Expected Problems 4](#_Toc296585993)

[1.4.1 Documentation/Specifications 4](#_Toc296585994)

[2 Specifics 5](#_Toc296585995)

[2.1 Early-phase Plans *(1-2 months)* 5](#_Toc296585996)

[2.1.1 Project documentation 5](#_Toc296585997)

[2.1.2 Initial test development 5](#_Toc296585998)

[2.2 Mid-phase Plans *(5-8 months)* 5](#_Toc296585999)

[2.2.1 Test development 5](#_Toc296586000)

[2.2.2 Reporting 5](#_Toc296586001)

[2.2.3 Automation 6](#_Toc296586002)

[2.3 Ongoing 6](#_Toc296586003)

[2.3.1 Test development and maintenance 6](#_Toc296586004)

[2.3.2 Reporting 6](#_Toc296586005)

[3 Timeline 7](#_Toc296586006)

# 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.

#### The end objective is for tests to be sufficiently comprehensive and automated to preclude the necessity of a dedicated tester to the ongoing maintenance of the project. (Although it is expected that a tester would be needed from time to time to add and update tests as additional features are considered.)

### 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

## Early-phase Plans (1-2 months)

### Project documentation

#### I intend to step through each class of each component in the project and make note of where sufficient documentation is lacking. **(Completed: 6/30/11, tiroger)**

#### Following 2.1.1.1, I intend to enlist the aid of the project community in production working documentation regarding components in question. **(We have some volunteers. I’m hoping that this will let me move on to other test work earlier than planned. 7/12/11 tiroger)**

#### Necessary documentation should include, at a minimum, general description of purpose, parameters, expected pre-conditions, and expected post-conditions of all public members throughout the project. **(This is presently in progress. A refactor of the system resulted in much of the core service getting documented. Dev tools and client components are planned for upcoming documentation meetings with community members. 8/12/11 tiroger)**

### Initial test development

#### Only limited test construction is expected to occur during this phase, primarily due to the necessity to complete functioning documentation for the project and components. Tests written during this time are expected to be trivial or near-trivial tests, excepting those tests explicitly requested by the developers. **(Trivial testing has been entirely abandoned in light of a recent project refactoring. Work is in progress for test tools requested by developers. 8/15/11 tiroger)**

#### Depending upon the level of community involvement in 2.1.1, it may be possible for serious test development to begin during this initial timeframe. Based upon experience with the open-source development communities and their general dynamics, this is not highly anticipated.

## Mid-phase Plans (5-8 months)

### Test development

#### It is expected that the overwhelming majority of test construction will occur during this phase. **(Original test design plans have been scrapped due to project refactoring. Will update document as new plan develops. 8/12/11 tiroger)**

#### Primary tests for consideration at this time are unit tests to cover individual components, and end-to-end automated build tests to evaluate end-user use scenarios. **(See note on 2.2.1.1)**

#### It may be possible to reduce the time necessary to construct the complete test suite, depending upon community participation.

### Reporting

#### All instances of significant failures will be reported in the tracking system designated for the project.

#### All issues which are declared “closed” by individuals should be cross-checked independently by the project tester, and re-opened if necessary.

### Automation

#### During this phase, as time allows, appropriate scripts and triggers will be constructed to allow new project changes to be automatically compiled, tested, and failure results relayed to the project tester. **(Due to compressed time schedule, it is necessary to move up work on end-to-end automation. Present expectation is to complete basic test automation before returning to work on 2.1.2 and 2.2.1. 7/14/11 tiroger)**

## Ongoing

### Test development and maintenance

#### Existing tests may require alteration if a component’s specifications change, otherwise all existing tests should require no maintenance.

#### New tests may need to be constructed to accommodate new component functionality.

### Reporting

#### See 2.2.2.

# Timeline

No specific timeline available at this time.