- **What is it:** Defining what work items need to happen for v1.0. From here we can determine our workback schedule.
- **Authors:** Clint Rutkas
- **Spec Status:** Living document

1. Overview

1.1. Elevator Pitch / Narrative

A PowerToy is a utility that helps an end user to do a task faster. Batch renaming files, rotating / resizing an image quickly tweaking a monitor display setting, being able to mount an ISO, robustly copying a file from one place to another. These are all examples of items that are / have been a PowerToy. Most PowerToys from 95 and XP were graduated into the Shell!

1.2. Customers

PowerToys exist for two reasons. Users want to squeeze more efficiency out of the Windows 10 shell and customize it to their individual workflows. We can be more targeted for scenarios to help do rapid iterations. Think about the countless small utilities that Microsoft engineers have written to make themselves more productive.

1.3. Existing Solutions or Expectations

There are a bunch of one-off applications in the Windows ecosystem. Each new PowerToy would look at the landscape and see how we can best address users' needs and what would be the best experience. There are also items in different states of support. Since PowerToys is Open Source, community can help update and contribute back to them.

1.4. Goals/Non-Goals

Goals:

- Quickly iterate and test new functionality that blocks Power users such as developers from adopting Windows
- Have functionality be designed in a way where the code could migrate back into Windows
- PowerToys uses public APIs only.

Non-goals:

- Create a custom shell for Windows
- Full transition to new UX

2. Definition of Success

Expected Impact: Business, Customer, and Technology Outcomes, Experiments …

No.	Outcome	Measure
1	Users population grows.	End user install
2	PowerToys is deemed bug free	Bugs opened against v1 PowerToys
3	PowerToys is stable	Crashes

| 4 | Users enjoy PowerToys | Users continue to use PowerToys 2 & 10 days after |

3. Requirements

3.1. Goals for v1

DevOps

P0 (must)

- Define how the PowerToys project operates: feature/bug requests are triaged, work is built/tested, releases are deployed, community communication
- Begin maintaining a public product roadmap (#932)
- Users are always on the latest (#413)
- Transparency report on telemetry
- Workflow for how items are reviewed, processed
- Workflow for how we build and ship (this doc + #932)

P1 (want)

- All shipping utilities have automated unit tests ([tests tag](#))

Enable everyone to use PowerToys

PO (must)

- In MS Store / MSIX Installer (#413)
- Telemetry must be able to be disabled (#964)
- All strings are localized for shipping utilities (#199)
- Support all processor types (#602, #490)
- All shipping utilities are accessible

Raise product quality

PO (must)

- Current shipping utilities are stable
- Current shipping utilities are straight forward and easy to use
- Settings no longer uses WebView (#243)
- Not running in elevated by default (#413, #942)
- Shortcut Guide UX is driven off a JSON file
- Shortcut Guide UX is rendered with controls to make it accessible with Narrator
- FancyZones Editor has a clearer model for working with layouts with multiple monitors
- FancyZones layouts are relative, not absolute so they can be used on different monitors with different resolutions
- FancyZones remembers app locations (Dock / undock scenario / Virtual Desktop)
- All utilities have gone through a security review

P1 (want)

- PT is aware of when elevated states occur and then alerts user that itself needs to be activated
- Attribution given to authors in app (#862)
- Start of migration to new UX via WinUI 3
  - Full transition will happen over the v1.x timeframe but not v1.0

P2 (nice to have)

- Shortcut Guide is UX can be configured by end user
- Unified UX Style
- FancyZones remembers app locations (Virtual Desktop)
  - Virtual Desktop DLL is a requirement
- Shippable Virtual Desktop DLL

3.2. What is shipping in v1

Current utilities shipping

- Shortcut Guide
- FancyZones
- PowerRename

New utilities for v1

- Image Resizer (#53)
- Quick Launcher (#44)
- Keyboard Remapper (#6)
- Markdown preview pane for Explorer (#914)
- SVG preview pane for Explorer (#963)
- Alt-Tab launcher (#861)

4. Timelines for stability

We are driving to what we are calling "stability". Stability is defined by [this list of items in our roadmap](#).

- 0.16 (March)
  - SVG/Markdown panes, image resizer,
  - Additional quality tests
- 0.17 (April)
  - Auto-updating
- 0.18 (May)
  - Launcher init release
  - Keyboard remapper init release
  - settings v2 initial release
- 0.19 (June)
  - Stability / bug fixes / quality push
- 0.20 (July)
  - Stability / bug fixes / quality push
  - ~~FZ Editor v2 init release~~
  - ~~OOBE ships~~
- 0.21 (August)
  - ~~Shortcut guide init release~~
  - Stability / bug fixes / quality push
  - Video conference mute pre-release (0.22 pre-release)
- 0.23 (September)
  - Video conference mute initial release

- - Localization initial release
    - Keystroke elevation issue
  - 0.24 (October)
    - Stability / bug fixes / quality push
    - OS Detection fix
  - 0.25 (November)
    - Stability / bug fixes / quality push
    - OOBE
    - See prioritized list
  - 0.26 (December)
    - Stability / bug fixes / quality push
    - See prioritized list

5. Utilities suggested for beyond stable

Please reference the [roadmap document](#) for a prioritized list of scenarios as a team we're directly targeting. This list can change as it is a living document.