Specyfikacja wymagań programu File Manipulator

# Wstęp

Aplikacja narzędziowa dedykowana na platformę MS Windows, służąca do masowego zarządzania nazwami, atrybutami oraz treścią plików w wybranych katalogach, a także do monitorowania zmian, takich jak dodanie, usunięcie, zmiana nazwy pliku lub też zmiana atrybutów i treści pliku/folderu.

# Charakterystyka produktu

Program jest adresowany do osób, które są zmuszone pracować przy dużych zbiorach plików, gdzie czasem trzeba zautomatyzować pewne procesy zarządzania. Mogą to być więc inżynierowie oprogramowania, fotograficy, montażyści obrazu i dźwięku. Docelowym środowiskiem jest MS Windows oraz systemy plików FAT, FAT32, exFAT, NTFS, ReFS, a także sieciowe systemy plików jak SMB.  
Planowane jest użycie środowiska MS Visual Studio Community do użytku niekomercyjnego oraz technologii WPF.NET z użyciem języka C#.  
Wszelkie dane potrzebne do zbudowania programu składowane będą w publicznym repozytorium Git.  
Projekt zarządzany będzie według metodologii Scrum na publicznych tablicach MS Azure DevOps.  
Wprowadza się także zarządzanie wersjami zgodnie z wytycznymi Semantic Versioning v.2.0.0[[1]](#footnote-1).  
Aplikacja używać będzie bibliotek o otwartym kodzie źródłowym z plikami binarnymi dostępnymi na platformie NuGet, które zostaną dołączone do .

# Wymagania:

## Funkcjonalne:

* Możliwość tworzenia dowolnej liczby zadań Watcher (służących śledzeniu zmian) oraz Manipulator (służących zarządzaniu plikami/katalogami).
* Zapis treści zadania do presetu, który zachowywany będzie w bibliotece zadań oraz możliwość wczytania zadania z biblioteki.
* Zapis/odczyt biblioteki zadań do pliku XML/JSON.
* Blokada manipulacji w folderach wymienionych poniżej w przypadku nieposiadania przez aplikację uprawnień administracyjnych:
  + Folder systemowy (przeważnie C:\Windows)
  + Folder danych aplikacji (C:\ProgramData, C:\Program Files (x86))
  + Folder zawierający profile wszystkich użytkowników (C:\Users)
* W przypadku niedostępności wybranych plików i folderów lub braku możliwości zapisu zadanie zostanie zatrzymane

## Użytkownika:

* Przejrzysty interfejs w formie kart (podobieństwo do przeglądarki internetowej)
* System pomocy i podpowiedzi – szczególnie odnośnie informowania o czynnościach nieodwracalnych

## Pozafunkcjonalne:

* Stabilność działania i odporność na problemy związane z wejściem/wyjściem
* Tworzenie kodu w oparciu o obecnie obowiązujące zasady tzw. czystości (S.O.L.I.D oraz z użyciem dobrze znanych wzorców projektowych i architektonicznych jak MVVM) celem polepszenia czytelności i przyspieszenia dalszego rozwoju

# Gromadzenie i przetwarzanie danych

Oprogramowanie nie gromadzi ani nie rozsyła żadnych danych o użytkowniku. Nie wymaga dostępu do sieci (wyjątkiem jest odczyt/zapis zadania na nośnikach sieciowych).

1. https://semver.org [↑](#footnote-ref-1)