CS673: WinChangeMonitor

Midterm Presentation
Group 3
Date: October 7th, 2025

WinChangeMonitor

- Ensuring compatibility across multiple Windows version
- Records and saves file and registry changes in a database.
- Generates detailed reports of those changes with timestamps.
- Offering user-friendly interface for effective reporting and monitoring

Project Scope

- C# and .NET Framework 4.8.
- Capturing changes in registry entries, file attributes, and directory structures.
- Generating a report in human readable format in JSON or XML.
- Using Jenkins, Docker, GitHub and Visual Studio to automate builds, testing, and deployment
- Delivering a **Minimum Viable Product (MVP)** capable of recording inventory, analyzing differences, and generating exportable reports
- The final release version will include installation instructions and documentation

Target Users

- System Administrators
- Computer Science Students
- Software Developers
- Digital Forensics Students and Educators
- IT Auditors

Roles and Responsibilities

Primary role - The main responsibility or function assigned to each team member.

Contributing role - Ensures fair and balanced distribution of work across the team.

	Princely Oseji	***Jeff Rose***	***Yeryoung Kim***	***Anjian Chen***	***Yu Wu***
Project Manager/Configuration Management	Primary	-	-	-	-
Software Development	Contributing	Primary	Contributing	Contributing	Primary
UX Designer	Contributing	Contributing	Primary	Contributing	Contributing
Quality Engineer	Contributing	Contributing	Contributing	Primary	Contributing

Functional Requirements

Design (Need) Requirements:

- Enable/disable monitoring for File System, Registry, and Service configuration, and select subsets for monitoring
- Perform pre-installation system inventory via UI
- Display UI status during inventory execution
- Allow system reboot between execution and post-installation execution
- Perform post-installation inventory and difference analysis
- Generate HTML difference report
- Show counts of modified files by directory
- Show new, modified, deleted files separately in UI with color distinction

Wish Requirements:

- Export reports as PDF, JSON, etc. (4)
- Provide filtering/search within reports (3)
- Support saving/loading monitoring configurations (3)

1-5, 1 = Lowest Priority

Non-Functional Requirements

Design (Need) Requirements:

- Interface must be intuitive for technical and non-technical users
- Reports must be accurate and consistent across runs
- Must support multiple Windows versions (Win10, Win11, Embedded, IoT)
- Reports must avoid exposing sensitive system data such as user credentials, authentication tokens, or secure registry values

Wish Requirements:

- Exported reports should be viewable on any system without the WinChangeMonitor tool installed (3)
- Codebase should follow documented standards and versioning practices (3)
- Color distinctions in UI must be perceivable by colorblind users (3)
- Optional database integration (SQL) for larger inventories (2)
- Key user and system actions, such as initiating inventories, generating reports, exporting files, and configuration changes, should be logged for troubleshooting and auditing (2)

1-5, 1 = Lowest Priority

GUI Design with Mock Image

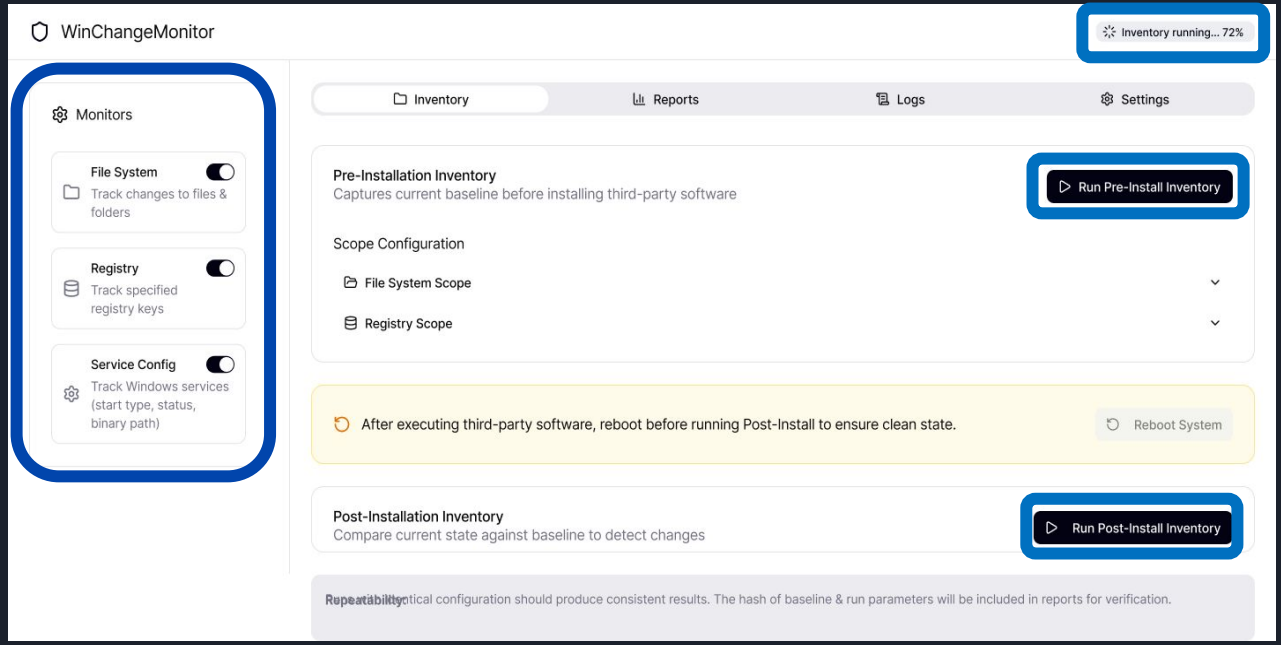
Purpose: Provide an intuitive, accessible interface to track and compare system changes before and after third-party installations.

Key Features

Monitoring Controls :
Enable/disable File System, Registry, and Service monitoring

Buttons :
Pre-installation Inventory, and Post-installation Inventory

Status indicator :
Indicating progress during inventory scans

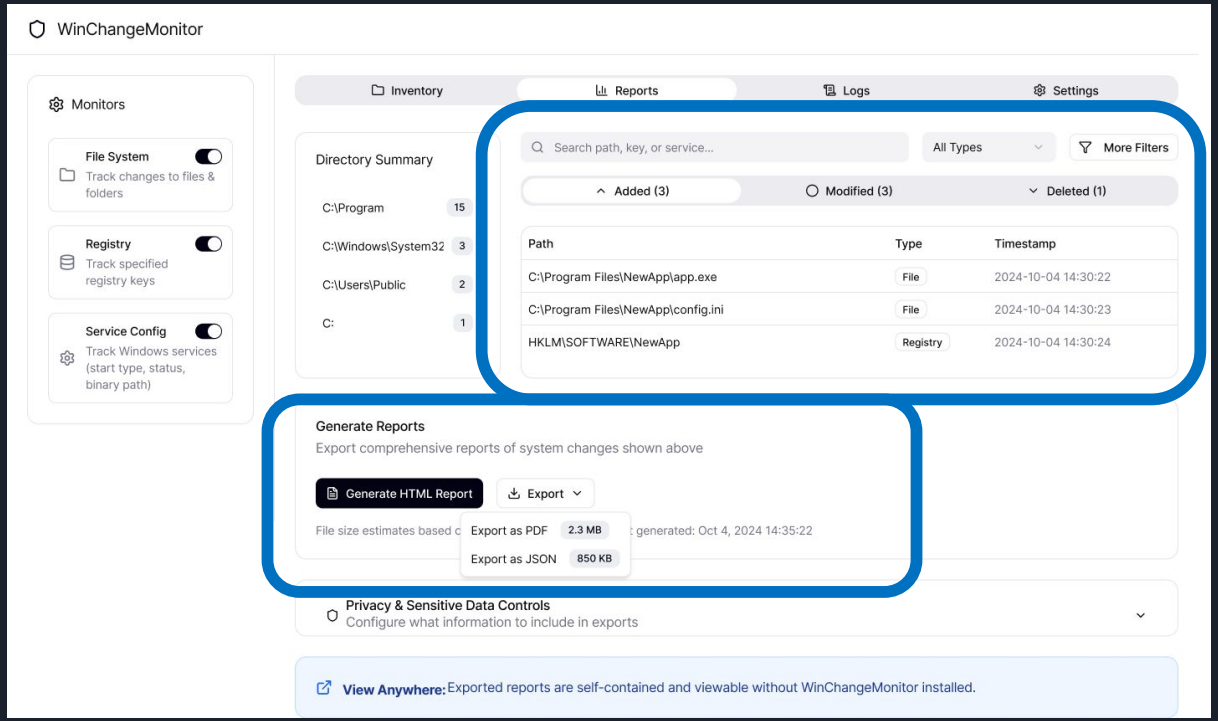


GUI Design with Mock Image

Report Panel :
Displays a summary of detected changes and allows exporting results in HTML, PDF, or JSON formats.

Design Goals:

- Simple workflow for both technical and non-technical users
- Clear, human-readable results
- Color-blind-friendly visuals
- Smooth integration with the API backend (GUI triggers inventory and diff operations)

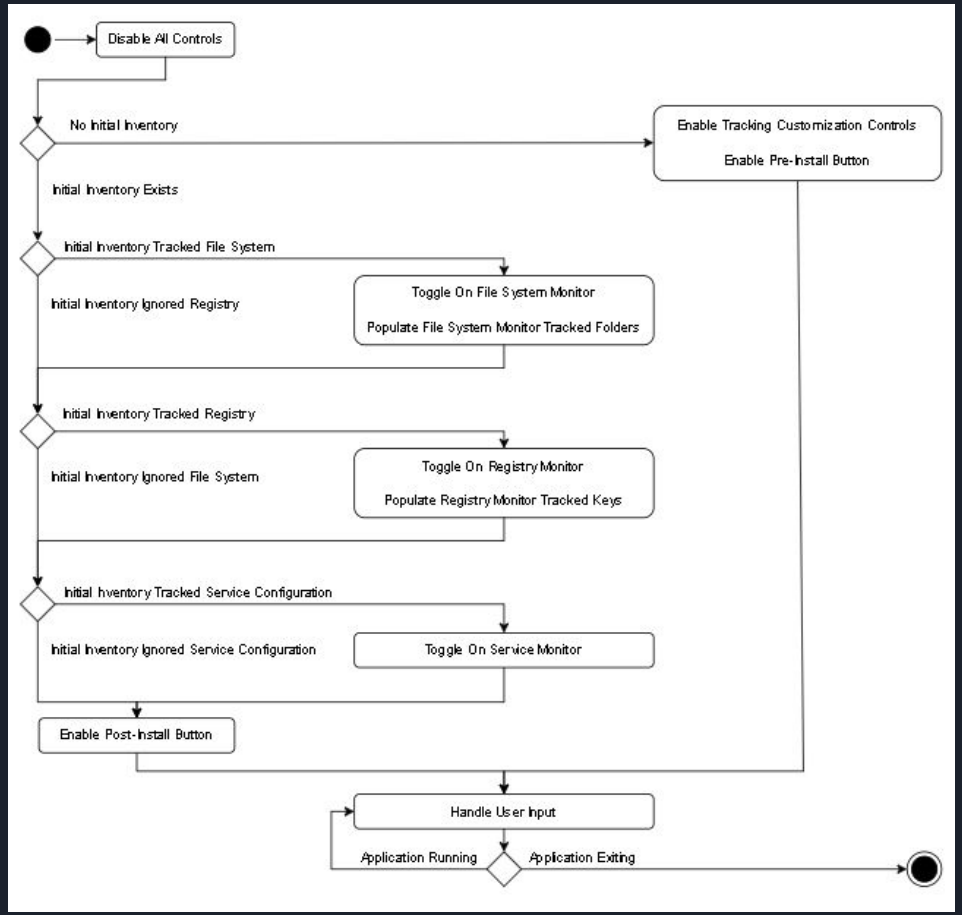


Program Start State Diagram Current Status

C# WinForms Controls use the Observer design pattern at design time rather than needing to subscribe each Listener to the desired Event at run-time.

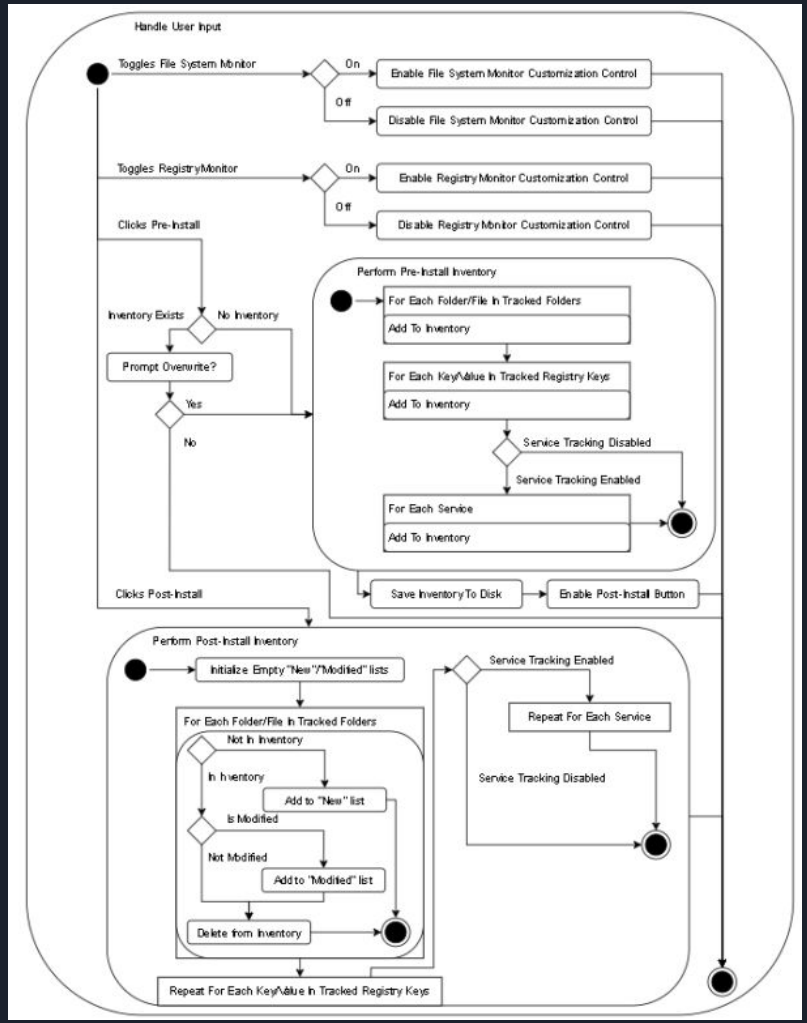
KEY LESSON LEARNED

Having requirements written as User Stories greatly reduces the effort involved in creating State Diagrams.



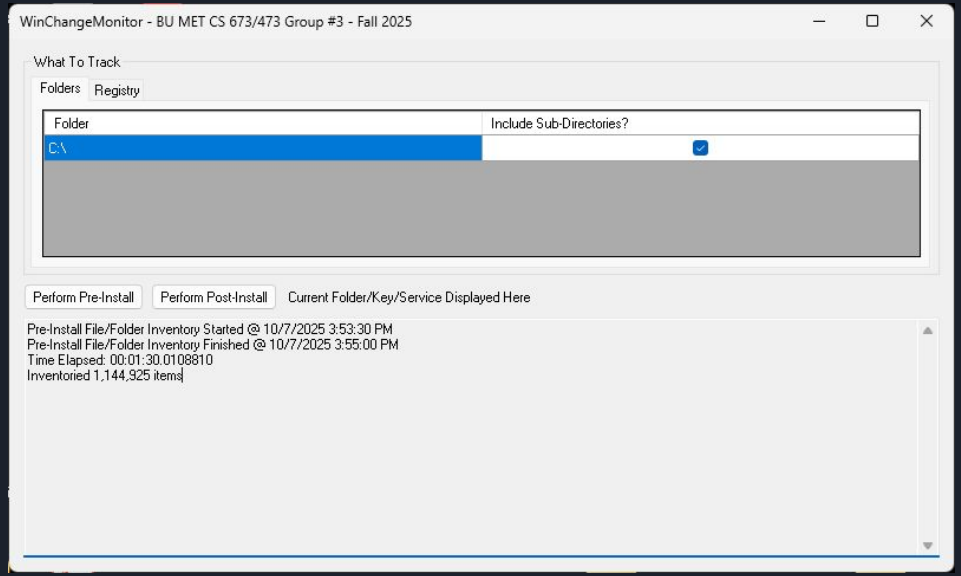
Handle User Input State Diagram Current Status

- The State Diagram will be expanded to include the Report generation workflow after Post-Install Inventory and Diff Analysis has completed.
- Separate data structures for each of the File System, Registry, and Services inventories will be necessary and will be reflected in the final State Diagram.



Current Program Status

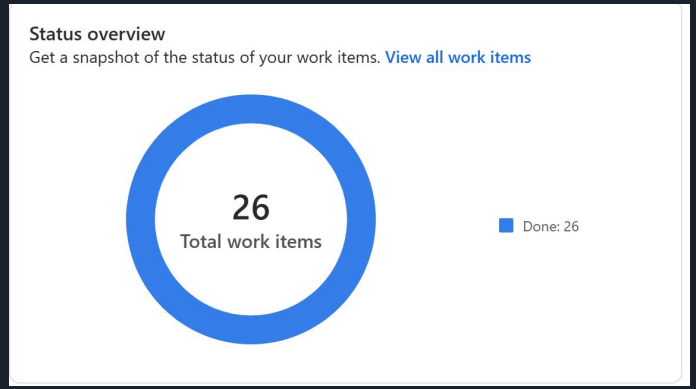
- A File/Folder Pre-Install Inventory with 1,144,925 items users 500 MB of RAM.
- Currently, a Dictionary is used to enable constant insertion/lookup/deletion operations.
- For systems with low RAM, a mechanism to offload inventory contents from RAM to disk will be ideal (Python's Lightning Memory-Mapped Database is a good example).



Progress Report

Key Achievements

- 2 sprints completed (1 ending today)
- Mock high-level interface design completed
- 25% of MVP has been completed
- State diagram is 90% complete



WNCHNGT3 Sprint 1 9 Sep – 23 Sep (6 work items)

- WNCHNGT3-12 Assign project roles
- WNCHNGT3-13 Define goals, objectives, and technologies
- WNCHNGT3-14 Draft and finalize project proposal
- WNCHNGT3-21 Develop Project Schedule
- WNCHNGT3-22 Identify Major Functionality
- WNCHNGT3-27 Configure JIRA and GitHub

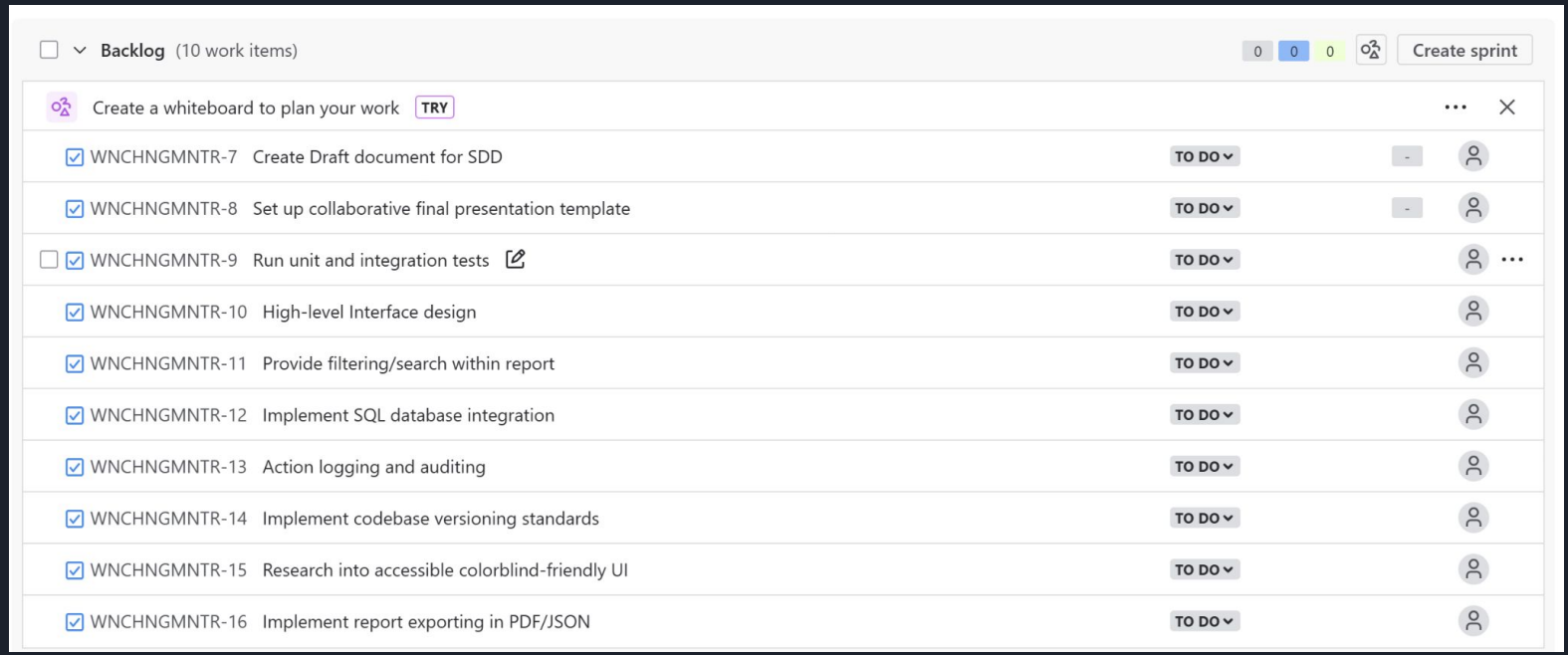
WNCHNGT3 Sprint 2 23 Sep – 30 Sep (9 work items)

Complete SCMP/SPMP and SRS documents and start working on prototypes

- WNCHNGT3-3 Create UI prototype
- WNCHNGT3-4 Create SCMP/SPMP doc
- WNCHNGT3-5 Draft Software Requirements Specification
- WNCHNGT3-10 Define project scope, risks, schedule
- WNCHNGT3-15 Develop estimation and risk strategy
- WNCHNGT3-16 Write user stories and acceptance criteria
- WNCHNGT3-17 Add usability and accessibility features
- WNCHNGT3-18 Upload documents to GitHub
- WNCHNGT3-20 Create Requirements List

Project Plans

Future plans have been included in the project backlog on JIRA and will be completed in subsequent sprints as seen in the figure below. This will be updated based on project requirements.



Important Links:

Github: https://github.com/pco30/Software_Eng

Thank you!