**Current Problem:** – Currently Engineering teams facing significant challenges while managing changes across interconnected repositories especially when updating shared libraries which are used by other applications. Teams are not able to identify all affected downstream applications easily. Manual verification of compatibility across multiple repositories are extremely time consuming.

**Target State:** By implementing this Cross Repository Dependency Impact analysis use case, engineers have complete visibility into cross repository dependencies with in their LOB applications and able to identify the potential compatibility issues early and improve overall efficiency in managing dependencies across projects.

This will enhance the user experience to collaboration, streamline multi repo updates, reducing the time spent on manual verification and decreasing production incidents caused by incompatible dependencies.

Currently, teams face significant inefficiencies with test environments due to booking conflicts, resource waste from incomplete manual cleanup, and poor visibility into environment availability. Engineers waste valuable time waiting for environments or troubleshooting contaminated ones, while managers lack insights into utilization patterns. The target solution will provide automated scheduling, provisioning, and cleanup of test environments with real-time status visibility. This will enhance user experience by eliminating booking conflicts, ensuring clean environments for every test cycle, optimizing resource utilization, and providing analytics on environment usage patterns—ultimately accelerating testing cycles and improving overall delivery speed.

**Current Problem:** Engineering teams are facing challenges with test environment management especially test environment booking. lack of visibility into environment availability and their status and manual data clean up process. This leads to resource waste and difficulty in tracking usage of testing environments for the respective releases.

**Target State:** An automated system that optimizes test environment management, ensuring efficient resource utilization and real-time visibility.

Implementing Automated Test Environment Management will streamline the booking of test environments. This will enhance user experience by reducing conflicts in test environment bookings. Users will benefit from increased productivity and reduced downtime.