1. Пару слів про актуальність вибраної теми - тестування програмного забезпечення — це процес дослідження програмного забезпечення, що має на меті вимір якості програмного забезпечення. Регресійне тестування – дорогий, але необхідний етап тестування програмного забезпечення, спрямований на повторну перевірку коректності зміненої програми.

Регресійне тестування потрібно виконувати при випуску будь-якої версії ПЗ, а темпи розвитку програмного забезпечення набирають таких обертів, що виробники змушені випускати нові версії програмного забезпечення все частіше і частіше додаючи нові функції та виправляючи існуючі проблеми

1. У даній роботі ставиться за мету: Аналіз методів регресійного тестування;Аналіз систем управління версіями; Розробка методу регресійного тестування на основі систем управління версіями; Розробка програмного продукту для планування регресійного тестування на основі систем управління версіями
2. Постановка задачі - Планування регресійного тестування ставить такі задачі:- мінімізація кількості виконуваних тестів - вибирати тести, що мають максимальну можливість виявити несправність   
   - мінімізувати час самого планування
3. ***Дано: P*** – певна версія програмного забезпечення – множина з ***N*** тестів, що використовується при первинному тестуванні програми ***P. P'*** – нова версія програми ***P.***

*Необхідно знайти таку підмножину тестів , що містить тільки ті тести що* задіюють код в *P* в точці, де в код був змінений, ця множина повинна містити – невідома, ідеальна підмножина регресійних тестів для тестування нової версії програми ***P'***, тобто

1. Існуючі методи регресійного тестування недосконалі, вони або пропонують вибирати тести випадковим чином, або вимагають проводити дуже ретельний аналіз змін щоб виключити можливість пропустити помилку, або не дають чіткого алгоритму знаходження множини
2. У галузі розробки програмного забезпечення, контроль версій є єдиною можливістю відстежувати і забезпечувати контроль за змінами у вихідному коді. Команди розробників одночасно проектують, розробляють та впроваджують декілька версій одного і того ж програмного забезпечення, які будуть розгорнуті в різних місцях, і одночасно працюють над виправленнями(patches). Певні помилки або функції програмного забезпечення часто присутні тільки в деяких версіях. Таким чином, з метою пошуку та виправлення помилок, життєво важливо мати можливість контролювати процес регресійного тестування і запускати різні набори тестів на різних версіях.

Системи контролю версій дозволяють отримати детальну інформацію в яких саме файлах(або модулях програмного забезпечення) відбулися зміни в порівнянні з попередньою версією.

1. На даному слайді представлена формалізація залежностей у програмному забезпеченні. Зв'язок між двома файлами визначється експертом – особою що приймає рішення, основуючись на цьому будується формула зв’язку між різними модулями програмного забезпечення. Це нам дає можливість ввести когнітивну карту залежностей у програмному забезпеченні – модель подання знань експерта у вигляді знакового орграфа
2. Для отримання когнітивної карти змін у програмному забезпеченні потрібні данні із систем управління версіями, що дозволяють ввести функцію-індикатор **)**, що вказує чи змінився файл , у версії у порівнянні з . За допомогою цієї функції будується міра зміни модулю ***j*** у зв’язку зі змінами у модулі ***i*** у версії **, що дає** можливість побудувати когнітивну карту змін у новій версії програмного забезпечення.
3. Проведений аналіз методів регресійного тестування та систем управління версіями дав можливість:

розробити метод планування регресійного тестування заснований на використанні систем управління версіями

розробити програмне забезпечення що реалізує дані функції за допомогою побудови когнітивних карт