

використання тільки фрикційних засобів проти ожеледі, поступово відмовившись від хімічних, що б в значній мірі зменшило хімічне навантаження на ґрунти.

Висновки. На жаль, нині в Україні для боротьби з ожеледдю на автодорогах використовується піщано-соляна суміш, яка несе не тільки загрозу навколишньому природному середовищу, але і величезні економічні збитки як для економіки, так і для окремо взятої людини. Тому з метою вирішення даної проблеми варто запозичити досвід розвинених країн, що дозволить зберегти не тільки довкілля, але і здоров'я людям. Дотримання наведених в тезі заходів також має додатково позитивно вплинути на вище зазначену проблему.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Піщано-соляна суміш – справжня «отрута» не тільки для ґрунту. – [Електронний ресурс]. – Режим доступу: <https://www.prostir.ua/?news=pischno-solyana-sumish-spravzhnya-otruta-ne-tilky-dlya-hruntu>.
2. Вирожемський В. К. Екологічні наслідки зимового утримання автомобільних доріг / В. К. Вирожемський, Н. А. Бородіна, М. Є Трух // Автошляховик України. – 2006. – №2 – С. 35–38.
3. Шимчук О. П. Вплив протиожеледних матеріалів для зимового утримання доріг на навколишнє середовище та безпеку дорожнього руху / О. П. Шимчук // Містобудування та територіальне планування. – 2014. – Вип. 54. – С. 486–490.
4. Засолена Україна, або як за кордоном борються зі снігом. – [Електронний ресурс]. – Режим доступу: <https://kiev.pravda.com.ua/columns/56a879c849195/>.
5. Технічні правила ремонту та утримання автомобільних доріг загального користування України. – [Електронний ресурс]. – Режим доступу: http://www.uazakon.com/documents/date_6m/pg_gcgvst/pg8.htm.
6. Нормативи прибирання снігу. – [Електронний ресурс]. – Режим доступу: <https://drda.org.ua/node/912>.

Поліщук Діана Олегівна – студентка групи ЕКО–14б, Інститут екологічної безпеки та моніторингу довкілля, Вінницький національний технічний університет.

Науковий керівник: Заюков Іван Вікторович – кандидат економічних наук, доцент, доцент кафедри безпеки життєдіяльності та педагогіки безпеки, Вінницький національний технічний університет, м. Вінниця, E-mail: Zivan@i.ua.

Polishchuk Diana O. – student of group EKO–14b, Institute of ecological safety and monitoring of environment, Vinnytsia National Technical University, Vinnytsia.

Supervisor: Zayukov Ivan V., Cand. Sc. (Econ.), Assistant Professor, Assistant Professor of Department of Health and Safety Studies, Vinnytsia National Technical University, Vinnytsia. E-mail: Zivan@i.ua

УДК 007.2

В. О. Отришко
М. А. Томчук

АНАЛІЗ МЕТОДІВ ТЕСТУВАННЯ БЕЗПЕКИ ПРОГРАМНИХ ПРОДУКТІВ

Вінницький національний технічний університет

В статті було розглянуто методи тестування безпеки програмних продуктів.

Ключові слова: безпека, інспектування, методи тестування, верифікація, атестація систем.

ANALYSIS OF SOFTWARE SAFETY TESTING METHODS

The article examined the methods of software security testing.

Keywords: security, inspections, testing methods, verification, system certification.

В сучасному світі, на етапі інформатизації всіх сфер життя, перед розробниками постають нові задачі, що вимагають ефективних рішень. Однією з таких задач є тестування програмного забезпечення. Сьогодні тестування програмного забезпечення – один з найбільш затратних етапів розробки, на нього виділяється від 50% до 65% загальних витрат до проекту.

Верифікацією й атестацією називаються процеси перевірки й аналізу, у ході яких перевіряється відповідність програмного забезпечення своєї специфікації і вимогам замовників. Верифікація й атестація охоплюють весь цикл життя ПЗ – вони починаються на етапі аналізу вимог і завершуються перевіркою програмного коду на етапі тестування програмної системи.

У процесах верифікації й атестації використовуються дві основні методики перевірки й аналізу систем: інспектування ПЗ і тестування ПЗ. Інспектування ПЗ має на увазі аналіз і перевірку різних представлень системи, наприклад, документації. Інспектування відбувається на всіх етапах розробки програмної системи. Тестування – динамічний метод верифікації й атестації, тому що застосовується до системи, що виконується [1].

На (рис. 1) показано місце інспектування і тестування в процесі розробки ПЗ. Стрілки вказують на ті етапи процесу розробки, на яких можна застосовувати дані методи.

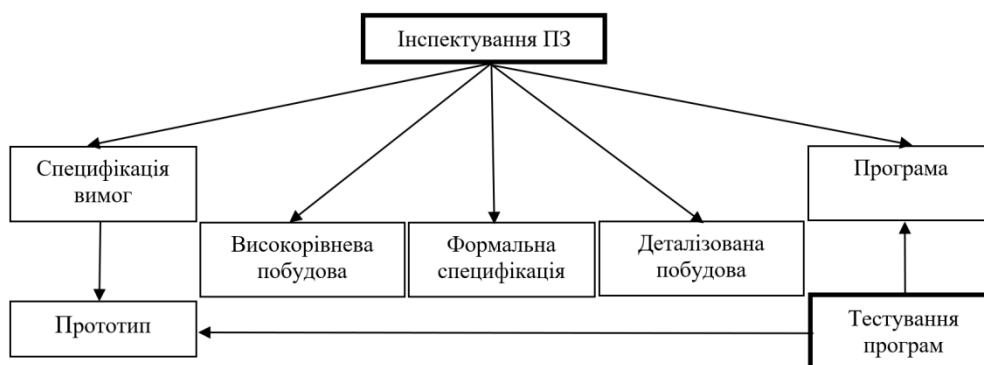


Рис. 1. Статична і динамічна верифікація й атестація

Відповідно до цієї схеми, інспектування можна виконувати на всіх етапах розробки системи, а тестування – у тих випадках, коли створений прототип або виконуюча програма. До методів інспектування відносяться: інспектування програм, автоматичний аналіз вихідного коду і формальна верифікація.

На різних етапах процесу розробки ПЗ застосовують різні види тестування [2].

Тестування дефектів проводиться для виявлення невідповідностей між програмним продуктом і його специфікацією, що обумовлені помилками в програмному коді. Такі тести розробляються для виявлення помилок у системі, а не для імітації її роботи.

Статистичне тестування оцінює продуктивність і надійність програм, а також роботу програми при використанні різних режимів її експлуатації. Тести розробляються з метою імітування, причому імітується реальна робота системи з реальними вихідними даними. Надійність функціонування системи визначається по кількості збоїв, відзначених у роботі програм.

Інспектування програмних систем. Системне тестування програм вимагає розробки величезної кількості тестів, їхнього виконання і перевірки. Це значить, що даний процес досить трудомісткий і дорогий. Кожен тест здатний знайти в програмі одну, чи кілька помилок. Інспектування програм не жадає від останніх бути завершеними, тому інспектувати можна навіть на початкових стадіях розробки. Під час інспектування перевіряється остаточне представлення системи.

Доведено, що інспектування є ефективним методом виявлення помилок, причому воно значно дешевше екстенсивного тестування. Інспектуванням можна знайти більш 60% усіх помилок, а при більш формальному підході (використовуючи математичні методи) – більш 90% [2].

Процес інспектування – формалізований. У ньому бере участь невелика група людей (звичайно не більш, ніж чотири людини). У кожного в групі є своя роль. Обов'язково повинні бути присутнім: автор, рецензент, інспектор, координатор. Рецензент «озвучує» програмний код, інспектор перевіряє його, координатор відповідає за організацію процесу.

Для початку процесу інспектування програми необхідні наступні умови: наявність точної специфікації коду (без повної специфікації неможливо знайти дефекти в програмному компоненті,

що перевіряються,); члени інспекційної групи повинні добре знати стандарти розробки; у розпорядженні групи повинна бути синтаксично коректна остання версія програми (немає рації розглядати код, що «майже завершений»).

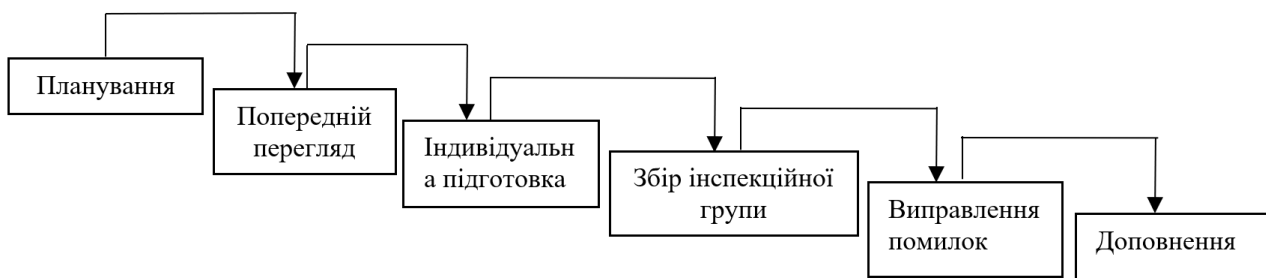


Рис. 1.2. Процес інспектування

На (рис. 2) показаний загальний процес інспектування. Він адаптований до вимог організацій, що використовують інспектування програм.

Сам процес інспектування повинний бути відносно коротким (не більш двох годин) і зосередженим тільки на виявленні дефектів, аномалій і невідповідностей стандартам. Інспекційна група не повинна пропонувати способи виправлення дефектів або рекомендувати які-небудь зміни в інших програмних компонентах [3].

Ціль автоматичного статичного аналізу – привернути увагу перевіряючого до аномалій у програмі, наприклад, до змінних, котрі використовуються без ініціалізації або ініціалізовані, але в програмі не використовувалися.

Статичний аналіз складається з декількох етапів (рис 3):

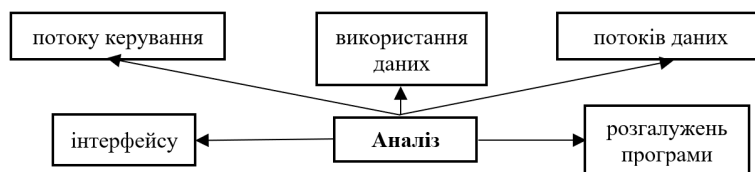


Рис. 3 Статичний аналіз

Слід зазначити, що аналіз потоку даних і аналіз розгалужень програми генерують величезну кількість інформації.

При розробці ПЗ є багато методів тестування наприклад метод «чиста кімната» для усунення дефектів використовується процес строгого інспектування. Ціль даного методу – створення ПЗ без дефектів. Назва «чиста кімната» узято за аналогією з виробництвом кристалів напівпровідників, де вирощування кристалів без дефектів відбувається в надчистій атмосфері (чистих кімнатах) [3].

У розробці ПЗ методом «чиста кімната» виділяють п'ять ключових моментів:

1. Формальна специфікація.
2. Покрокова розробка.
3. Структурне програмування.
4. Статична верифікація.
5. Статичне тестування системи.

Процес розробки ПЗ методом «чиста кімната» планується таким чином, щоб забезпечити строге інспектування програм, що супроводжується строгими математичними доказами погодженості і коректності перетворень.

Звичайно розробкою великих систем методом «чиста кімната» займаються три групи розроблювачів: група специфікації, група розробки (розробляє і перевіряє ПЗ), група сертифікації (розробляє контрольні тести).

У результаті використання методу «чиста кімната» програмний продукт містить украй мало помилок, а його вартість менше, ніж у розробленого традиційними методами.

Метою тестування дефектів є виявлення в програмній системі прихованих дефектів до того, як вона буде здана замовникові. Тестування дефектів протилежно атестації, у ході якої перевіряється відповідність системи своєї специфікації. Під час атестації система повинна коректно працювати з усіма заданими тестовими даними. При тестуванні дефектів запускається такий тест, що викликає некоректну роботу програми, і, отже, виявляє дефект.

Тестування дефектів демонструє наявність, а не відсутність дефектів у програмі.

Повне тестування, коли перевіряються всі можливі послідовності виконання програми, неможливо. Тому тестування повинне базуватися на деякій підмножині всіляких тестових сценаріїв.

Методів тестування дефектів існує не багато [4].

Тестування методом чорної скриньки полягає в тому, що вся система представляється як «чорна скринька», поведінку якої можна визначити тільки за допомогою вивчення вхідних і відповідних вихідних даних. Інша назва цього методу – функціональне тестування, тому що проводиться аналіз тільки виконуваних функцій.

Області еквівалентності. Вхідні дані, які належать одному класові, мають загальні властивості, наприклад це позитивні числа. Звичайно для всіх даних з якого-небудь класу поведінка програми однакова (еквівалентно).

Структурне тестування. Метод структурного тестування припускає створення тестів на основі структури системи і її реалізацій. Такий підхід іноді називають методом «білої скриньки», щоб відрізнити його від тестування методом чорної скриньки.

Тестування розгалужень. Це метод структурного тестування, при якому перевіряються всі незалежно виконувани розгалуження компонента або програми. Якщо виконуються всі незалежні розгалуження, то і всі оператори повинні виконуватися принаймні один раз. Більш того, всі умовні оператори тестуються як із вірними, так і з хибними значеннями умов.

Тестування зборки повинне починатися відразу після створення працездатних версій компонентів системи [4].

Спадне і висхідне тестування. Методики спадного (СТ) і висхідного тестування (ВТ) виявляють різні підходи до системної інтеграції. При спадній інтеграції компонента високого рівня інтегруються і тестуються ще до закінчення їхнього проектування і реалізації. При висхідній інтеграції перед розробкою компонентів більш високого рівня спочатку інтегруються і тестуються компоненти нижнього рівня.

Тестування інтерфейсів. Тестування інтерфейсів (ТІ) виконується в тих випадках, коли модулі або підсистеми інтегруються у великі системи. Кожен модуль або підсистема має заданий інтерфейс, що викликається іншими компонентами системи.

Тестування з навантаженням. Після повної інтеграції системи можна оцінити такі інтеграційні властивості системи, як продуктивність і надійність. Щоб переконатися, що система може працювати з заданим навантаженням, розробляються тести для виміру продуктивності.

Тестування – дорогий і трудомісткий етап розробки програмних систем. Тому створено широкий спектр інструментальних засобів для підтримки процесу тестування, що значно скорочують витрати на нього.

Щоб бути упевненим, що система відповідає вимогам, необхідно вимірити її показники безвідмовності, з огляду на роботу типового користувача. Процес виміру показників безвідмовності складається з чотирьох етапів: спочатку вивчаються аналогічні існуючі системи (визначається операційний профіль), потім йде підготовка тестових даних, подальший етап – власне тестування, останнім кроком виконується обчислення показників безвідмовності. Труднощі виникають з кількох причин [5]:

невизначеність операційного профілю (профілі можуть неточно показувати реальне використання системи)

висока вартість генерації тестових даних (якщо немає можливості автоматичної генерації тестових даних, то створення великої кількості тестових даних вимагає великих витрат часу і, відповідно, засобів)

статистична невизначеність у випадку високої безвідмовності (для точного виміру показників безвідмовності необхідно згенерувати статистично значиме число відмовлень).

Операційний профіль (ОП) показує практику використання системи. Він складається зі специфікації класів вхідних даних і імовірності їхньої появи. Якщо система ПЗ інноваційна, передбачати, як вона буде використовуватися, складно.

Одержання гарантій безпеки системи й атестація її безвідмовності – різні процеси. Безвідмовність можна визначити кількісно за допомогою різних числових показників. Безпеку не можна вірогідно визначити кількісними способами, отже, її неможливо вимірити в ході тестування системи [5].

Тому атестація безпеки визначає рівень надійності системи, що може варіюватися від «дуже низького» до «дуже високого». Тут потрібна професійна оцінка безпеки. У багатьох випадках визначення безпеки базується на досвіді організації, що розробляє систему. Якщо в організації вже є попередньо розроблені надійно функціонуючі безпечні системи, то розумно припустити, що в даній організації будуть розроблені подібні безпечні системи. З іншого боку, оцінка безпеки повинна спиратися на реальну архітектуру системи, результати верифікації й атестації, а також на процеси, що застосовувалися при розробці системи.

Верифікація й атестація систем, критичних по забезпеченню безпеки, має багато загального з тестуванням будь-яких систем з високими вимогами надійності. Щоб знайти найбільшу кількість помилок, варто застосовувати всебічне тестування, а при оцінці безпеки використовувати статичні методи тестування. Однак унаслідок надзвичайно низької частоти відмовлень, властивих багатьом КС, за допомогою статичного тестування не завжди вдається кількісно оцінити безвідмовність, тому що для цього потрібно дуже велике число тестів. Ці тести лише дають підставу вважати ту або іншу КС безпечною [5].

При створенні КС, важливий всебічний аналіз розроблюваної системи. Існує п'ять типів аналізу системи, обов'язкових для КС (рис 4):

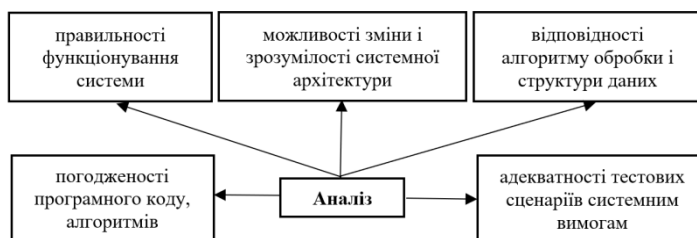


Рис. 4 Типи аналізу системи КС

Висновки: Таким чином, докази безпеки системи будуються на наступному припущенні: кількість помилок у системі, що приводять до аварійних ситуацій, набагато менше загального числа помилок у системі. Забезпечення безпеки повинне зосередитися на виявленні потенційно небезпечних помилок. Якщо виявляється, що ці помилки не виявляються або виявляються, але не приводять до серйозних наслідків, то система вважається надійною. Доказу правильності програм були запропоновані як методи верифікації ПЗ більш 25 років тому. Однак ці методи в основному використовуються тільки в лабораторіях. Практичні проблеми побудови доказу правильності ПЗ настільки складні, що деякі організації вважають використання даних методів у процесі розробки звичайних систем невиправдано дорогим. Але, як відзначалося раніше, для ряду КС економічно вигідно використовувати доказ правильності системи, чим ліквідувати наслідки відмовлень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Ю. Г. Карпов. Теория автоматов. – Спб.: Питер, 2002 – 224 с.: ил.
2. Электронный Архив для инженеров программного обеспечения. <http://www.cs.queensu.ca/Software-Engineering/>
3. Software Engineering Questions and Answers. <http://www.cs.queensu.ca/Software-Engineering/questions.html>
4. Ресурсы сервера Института Инженерии Программного Обеспечения Карнеги Меллона (Carnegie Mellon Software Engineering Institute). <http://www.sei.cmu.edu/>
5. SybaseDevel.Ru – русский портал для разработчиков. <http://www.sybasedevel.ru>

Олександрович Отришко Володимир – студент групи 2ПІ-17м, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, місто Вінниця, e-mail: wolf1702@ukr.net.

Томчук Микола Антонович, кандидат технічних наук, доцент кафедри Безпеки життєдіяльності та педагогіки безпеки, Вінницький національний технічний університет, Вінниця, e-mail: tomchuk.mykola@gmail.com

Otryshko Volodymyr O. - Faculty of Information Technologies and Computer Engineering, group 2PI-17m, Vinnytsia National Technical University, Vinnytsia, email: wolf1702@ukr.net;

Томчук Микола А., Cand. Sc. (Eng.), Assistant Professor of Department of Health and Safety Studies, Vinnitsa National Technical University, Vinnytsia, e-mail: tomchuk.mykola@gmail.com

УДК 666.13

В.О. Семенюк

ОРГАНІЗАЦІЯ РОБОТИ ТА ЧАСУ ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ПРАЦІВНИКІВ

Вінницький національний технічний університет

В статті розглянуто особливості та значення організації роботи та часу для підвищення продуктивності працівників. Вплив втручань та багатозадачності на продуктивність працівників.

Ключові слова: Організація роботи та часу; багатозадачність; продуктивність; фокус;

ORGANIZATION OF WORK AND TIME TO INCREASE EMPLOYEE PRODUCTIVITY

The article considers the peculiarities and importance of organization of work and time for increasing the productivity of employees. Influence of interventions and multitasking on employee productivity.

Keywords: Organization of work and time; multitasking; productivity; focus;

Особистий та професійний розвиток допомагає запобігти негативним думкам, що перешкоджають нам робити кроки до самовдосконалення. Ми часто відсиджуємося і чекаємо щорічного огляду результатів роботи, щоб визначити сфери, які нам потрібно покращити. Поставте собі за ціль бути відповідальним, покращувати свої навички та постійно навчатися, встановлюючи особисті орієнтири та регулярно переглядаючи їх. Навчання та самовдосконалення ведуть до кращої якості життя, підвищують впевненість і особистий розвиток, позитивно впливають на наше життя [1-9].

Ми також можемо зробити багато речей, щоб покращити свою продуктивність на роботі, речі, які не роблять багато людей. Наскільки успішним ви знаходитесь на роботі визначається в основному вашим ставленням і тим, як ви проводите свій час. Ось 20 способів підвищити репутацію та продуктивність на роботі, а також досягти нових рівнів кваліфікації та професійного розвитку:

1) Організувати та визначити пріоритети

Створіть щоденний графік і дотримуйтеся його. Визначте найважливіші три або чотири критичні проекти, які потрібно виконати. Переконайтеся, що список ваших завдань є керованим, доповнює вартість та вигідний вашій фірмі.

2) Зупиніть багатозадачність

Багатозадачність знижує IQ, знижує EI (емоційний інтелект), сповільнює роботу, підвищує рівень стресу і спричиняє помилки. Працюйте над однією справою замість цього.

3) Уникайте відволікань

Ви знали, що фокус є основною якістю продуктивних людей? Наші мізки мають таку систему, що працюють краще, коли ми зосереджуємося на одному завданні. Практика залишатися зосередженим і прагнути завершити одне завдання перед пірнанням в інше.

4) Управління перериваннями