**Лекція №6**. **Інфраструктура перевірки правильності програмних систем**

*Програмна система* (ПС) — група інтегрованих програмних засобів, які підтримують певний діловий процес споживача (або його частину) і використовують загальне сховище даних.

Під *інфраструктурою* перевірки правильності (доведення, верифікації і тестування) програмних систем розуміють інтегрований набір загальнодоступних технічних, технологічних і методологічних ресурсів, що знаходяться у розпорядженні команди розробників, верифікаторів і тестувальників, які виконують роботи з розроблення правильної системи за договорами із організаціями-замовниками.

**1.** **Компоненти інфраструктури розробки ПС.**

Розглядаються техніко-технологічний та кадровий аспекти з погляду на виконання розроблення як основного процесу за ISO 12207 таким чином.

**Техніко-технологічний аспект.**

1. Техніка і комунікації:

* Комп’ютери користувачів, файлові сервери
* Локальні комп’ютерні мережі (ЛКМ)
* Глобальна комп’ютерна мережа (ГКМ)
* Електронна пошта
* Техніка для тестування
* Офісна техніка
* Інші складові комплексу технічних засобів

2. Загальносистемне ПЗ та інструменти:

* Клієнт-серверні технології
* Операційні системи
* Офісні системи
* Системи документообігу
* Утиліти
* Засоби захисту інформації (антивіруси)
* CASE-інструменти, системи програмування
* СУБД
* Графічні інструменти

3. Інформаційні ресурси і стандарти розробки:

* Методології розробки
* Інструменти керування проектами, конфігураціями
* Системи підтримки використання ресурсів Інтернет
* Нормативні документи, які стосуються технічних, програмних, комунікаційних засобів, даних і захисту інформації
* Нормативні документи оформлення матеріалів
* Методичні матеріали, шаблони і заготовки документів

4. Міжпроектна програмна підтримка

* Розроблені програми (модулі), визнані здатними до загального користування, документовані та поміщені під контроль конфігурації.

**Кадровий аспект.**

1. Навчання методам і технологіям:

* Можливості організації по навчанню спеціалістів методам та прийомам розробки ПЗ
* Можливості вивчення спеціалістами техніко-технологічних компонент інфраструктури

2. Обмін позитивним та негативним досвідом:

Культура «відкритого» сприйняття/передачі набутого досвіду, знань, характерних помилок. Сприяння розповсюдженню позитивного досвіду. Не приховування власних помилок і не перекладання відповідальності за них. Бажання навчатись/навчати

3. Накопичення і закріплення позитивного досвіду:

* Визначення форматів і засобів накопичення і зберігання здобутого досвіду (опитування, семінари тощо)
* Створення бібліотек активів організації за принципом «кращий об’єкт». Включення їх у сферу керування конфігурацією. Забезпечення доступності.

4. Стандарти міжпроектної взаємодії:

* Визначення стандартів (меж компетенції, знань) по процесам життєвого циклу (ЖЦ) ПС. Уніфікація та стандартизація прийомів роботи з метою побудови і підтримки базового процесу програмної інженерії
* Профілювання знань для забезпечення замінюваності спеціалістів в проекті. Дотримання принципу «глибокі знання у вузькій сфері»

**2. Ролі спеціалістів в організаційній структурі розробки**

**Ролі на рівні організації**

1. Група техніко-технологічної підтримки:

* Вивчення ринку послуг і попиту в організації відносно техніки та загальносистемного ПЗ.
* Придбання/встановлення/підтримка техніки.
* Придбання/встановлення/підтримка загальносистемного ПЗ.
* Навчання/консультаційні послуги співробітникам.
* Рекомендації по застосуванню техніки і технологій в проектах.

2. Група захисту інформації:

* Вивчення стану справ в області захисту інформації і накопичення досвіду.
* Забезпечення захисту інформації в організації.
* Перевірка захисту інформації в організації.
* Підтримка проектів в питаннях захисту інформації.

3. Група інженерії процесу

* Визначення, супровід та вдосконалення базового процесу програмної інженерії. Забезпечення нормативно-методичної підтримки виконання процесів ЖЦ. Організація та поповнення сховища (бібліотеки) активів організації.
* Допомога менеджерам проектів в адаптації базового процесу до потреб проектів. Підбір або виготовлення форм (шаблонів) документів для інженерії проектів.
* Підтримка процесу документування в проектах, зокрема виконання важких графічних робіт, оформлення документів згідно стандартів оформлення. Нормоконтроль та друк документів.
* Міжпроектна координація в частині накопичення досвіду і організації навчання.
* Підтримка керування конфігурацією в проектах.

4. Незалежна група якості (SQA-група):

Планування та виконання дій по контролю і гарантії дотримання дисципліни створення програмної продукції в проектах (організація перевірок робіт в контрольних точках проектів, визначених календарними планами).

* Контроль документів і продуктів ПЗ в контрольних точках проектів на предмет дотримання діючих стандартів та інших нормативних документів, встановлених у вимогах замовника.
* Звітність безпосередньо перед керівником організації

5. Незалежна група верифікації та валідації (V&V-група):

* Виконання функції верифікації (по домовленості з групою SQA).
* Планування і проведення незалежного кваліфікаційного тестування інтегрованих компонент ПЗ або програмних продуктів з метою визначення їх відповідності потребам замовника.
* Координація планів робіт з менеджерами проектів відносно вимог до тестового середовища, строків і порядку передачі ПЗ на тестування.
* Представлення звітів (результатів) тестування менеджерам проектів для прийняття мір по виправленню ПЗ.
* Незалежність від менеджерів проектів в частині визначення об’ємів і методів тестування.
* Звітність перед керівником організації за дотримання порядку тестування і стан розроблених програмних продуктів.

6. Група підтримки замовника:

* Зв’язок із замовником з питань автоматизації ділових процесів.
* Підтримка процесів керування вимогами, навчання користувачів, супроводу (або допомога в їх виконанні на рівні окремих проектів).

**Ролі на рівні проекту**

1. Керівник проекту системи:

* Повна фінансова відповідальність за виконання проектних домовленостей перед замовником.
* Керування розробкою складових створюваної продукції – проектів ПЗ, комплексу технічних засобів, засобів захисту інформації.
* Відповідальність за дії виконавців проекту.

2. Системні аналітики:

* Дослідження умов та потреб автоматизації діяльності організації-споживача.
* Системний аналіз вимог споживача і формування концепції системи.
* Контроль обґрунтованості проектних рішень, що приймаються.

3. Група якості проекту:

* Контроль якості робочих продуктів, створених процесами ЖЦ (на відповідність стандартам, методикам тощо).
* Звітність тільки керівнику проекту.
* Може бути відсутньою, якщо на рівні організації діє незалежна група якості.

4. Група V&V проекту:

* Перевірка відповідності робочих продуктів, вироблених на певному етапі ЖЦ, вимог до них, встановлених на попередньому етапі.
* Може виконувати тестування окремих компонент ПЗ, а також системне (інтеграційне) тестування ПЗ, виробленого в проекті.
* Звітність тільки керівнику проекту.

5. Менеджер проекту ПЗ:

* Повна відповідальність за усі проектні рішення та дії, пов’язані з розробкою ПЗ в проекті
* Підбір і контроль ресурсів проекту, а також графіка робіт.
* У великих або розподілених програмних проектах може бути декілька менеджерів (по підсистемам або рівням проекту ПЗ).

6. Проектувальники:

* Прийняття і документування проектних рішень по архітектурі і функціям ПЗ. Узгодження рішень з менеджером проекту ПЗ.
* Дотримання стандартів якості (забезпечення досягнення характеристик якості).

7. Програмісти:

* Програмування або моделювання компонентів ПЗ по проектним специфікаціям, підготованих проектувальниками.
* Дотримання стандартів якості при програмуванні (по зручності супроводу коду, зручності застосування програм).
* Відладка та автономне тестування розроблених компонент.

8. Група керування конфігурацією:

Виконання процесу конфігураційного керування версій ПЗ і робочих продуктів проекту ПЗ.

*9. Група супроводу:*

* Виконання процесу супроводу версій ПЗ і робочих продуктів проекту ПЗ під час дослідної експлуатації і під час встановленого періоду супроводу.
* Навчання користувачів.
* Виконання процесу розв’язання проблем.
* Можуть бути членами групи підтримки замовника.

10. Група проекту ЛКМ:

* При розробці системи «під ключ» проектування і монтаж ЛКМ для встановлення в організації споживача.
* Закупівля і встановлення КТЗ і загальносистемного ПЗ, пуско-налагоджувальні дії.

**Класифікація інструментів по SWEBOK**

**Інструменти роботи з вимогами:**

* Засоби моделювання.
* Засоби трасування.

**Інструменти проектування:**

* UML.
* Бізнес-проектування.
* Проектування БД.

**Інструменти конструювання:**

* Редактори програм.
* Компілятори і генератори коду.
* Інтерпретатори.
* Дебаггери.

**Інструменти тестування:**

* Генератори тестів.
* Засоби виконання тестів.
* Інструменти оцінки тестів.
* Засоби керування тестами.
* Інструменти аналізу продуктивності.

**Інструменти супроводу:**

* Засоби візуалізації.
* Інструменти реінженерії.

**Інструменти управління конфігурацією:**

* Інструменти відслідковування дефектів і проблем.
* Інструменти управління версіями.
* Інструменти зборки та випуску.

**Управління інженерією:**

* Інструменти планування та відстежування , прогнозування вартості.
* Інструменти керування ризиками.
* Засоби кількісної оцінки.

**Інструменти підтримки процесів:**

* Інструменти моделювання процесів.
* Засоби керування процесами.
* Інтегровані CASE-середовища і рольові платформи розробки.
* Процес-орієнтовані середовища розробки.

**Інструменти забезпечення якості:**

* Інструменти інспекції, підтримка оглядів та аудитів.
* Інструменти статичного аналізу.

**Додаткові аспекти:**

* Засоби інтеграції інструментів: програмні платформи (Java, .NET Framework), платформи розподілених обчислень (CORBA, WebServices).
* Мета інструменти: засоби генерації інших інструментів, компілятор компіляторів тощо.
* Засоби оцінки інструментів.

Під *інфраструктурою* перевірки правильності (доведення, верифікації і тестування) програмних систем розуміють інтегрований набір загальнодоступних технічних, технологічних і методологічних ресурсів, що знаходяться у розпорядженні команди розробників, верифікаторів і тестувальників, які виконують роботи з розроблення правильної системи за договорами із організаціями-замовниками.

Вище ми розглядали частково розглядали ці питання, зокрема, стратегічну мету тестування, об'єкти процесу тестування, дії команди розробників, верифікаторів і тестувальників, класифікацію помилок (помилка, дефект, відмова, які розподіляють на класи: 1) логічні і функціональні помилки; 2) помилки обчислень і часу виконання; 3) помилки вводу-виводу і маніпулювання даними; 4) помилки інтерфейсів; 5) помилки обсягу даних і ін.).

Наведені типи помилок розподіляються за процесами ЖЦ і їм відповідають такі джерела їхнього виникнення:

– ненавмисне відхилення розробників від робочих стандартів або планів реалізації;

– специфікації функціональних і інтерфейсних вимог виконані без дотримання стандартів розробки, що призводить до порушення функціонування програм;

– організації процесу розробки – недосконале або недостатнє управління керівником проекту ресурсами (людськими, технічними, програмними і т.д.) і питаннями тестування й інтеграції елементів проекту.

За рекомендацією стандарту ISO/IEC–12207 процес тестування кожного процесу ЖЦ і типи помилок, що виявляються під час, можна визначити так.

*Процес розробки вимог*. При визначенні вихідної концепції системи і вихідних вимог до системи виникають помилки аналітиків при специфікації вищого рівня системи і побудові концептуальної моделі предметної області.

Характерними помилками цього процесу є:

– неадекватність специфікації вимогам кінцевих користувачів;

– некоректність специфікації взаємодії ПС із середовищем функціонування або з користувачами;

– невідповідність вимог замовника окремим і загальним властивостям ПС;

– некоректність опису функціональних характеристик;

– незабезпеченість інструментальними засобами всіх аспектів реалізації вимог замовника й ін.

*Процес проектування.* Помилки при проектуванні компонентів можуть бути наслідком недоліків в описі алгоритмів, логіки керування, структур даних, інтерфейсів, логіки моделювання потоків даних, форматів вводу-виводу та ін. В основі цих помилок лежать дефекти специфікацій заданих аналітиками і недоробки проектувальників. До них належать помилки, пов'язані з:

– погодженістю інтерфейсу користувача із середовищем;

– описом функцій (неадекватність цілей і задач компонентів, що виявляються при перевірці комплексу компонентів);

– визначенням процесу обробки інформації і взаємодії між процесами (результат некоректного визначення взаємозв'язків компонентів і процесів);

– некоректним завданням даних і їхніх структур при описі окремих компонентів і ПС у цілому;

– некоректним описом алгоритмів модулів;

– визначенням умов виникнення можливих помилок у програмі;

– порушенням прийнятих для проекту стандартів і технологій.

*Процес кодування.* На даному процесі виникають помилки, що є результатом дефектів проектування, помилок програмістів і менеджерів у процесі розроблення і налагодження системи. Причиною помилок є:

– безконтрольність значень вхідних параметрів, індексів масивів, параметрів циклів, вихідних результатів та ін.;

– неправильна обробка нерегулярних ситуацій при аналізі кодів повернення від викликуваних підпрограм, функцій і ін.;

– порушення стандартів кодування (погані коментарі, нераціональне виділення модулів і компонентів та ін.);

– використання одного імені для позначення різних об'єктів або різних імен одного об'єкта, погана мнемоніка імен;

– непогоджене внесення змін у програму різними розробниками та ін.

*Процес тестування****.*** На цьому процесі помилки допускаються програмістами і тестувальниками при виконанні технології збирання і тестування, вибору тестових наборів і сценаріїв тестування та ін. Відмови в програмному забезпеченні, викликані такого роду помилками, повинні виявлятися, усуватися і не впливають на статистику помилок компонентів і на програмне забезпечення в цілому.

*Процес супроводу.* На процесі супроводу виявляються помилки, причиною яких є недоробки і дефекти експлуатаційної документації, недостатні показники кодифікованості й легкості читання, а також некомпетентність осіб, відповідальних за супровід і/або удосконалення ПС. Залежно від сутності внесених змін на цьому процесі можуть виникати практично будь-які помилки, аналогічні раніше перерахованим помилкам на попередніх процесах.

**Джерела помилок.**

Помилки можуть бути виникнути в процесі розроблення проекту, компонентів, коду і документації. Як правило, вони виявляються при виконанні або супроводі програмного забезпечення в найбільш несподіваних і різних її точках.

Причиною появи помилок є – нерозуміння вимог замовника; неточна специфікація вимог у документах проекту та ін. Це приводить до того, що реалізуються деякі функції системи, що будуть працювати не так, як пропонує замовник. У зв'язку з цим проводиться спільне обговорення замовником і розробником деяких деталей вимог для їхнього уточнення.

Команда розробників системи може також змінити мову опису системи.

Деякі помилки можуть бути не виявлені (наприклад, неправильно задані індекси або значення змінних цих операторів).

**Визначення тесту.**

Для перевірки правильності програм спеціально розробляються тести і тестові дані. Під *тестом* розуміється деяка програма, призначена для перевірки працездатності іншої програми і виявлення в ній помилкових ситуацій. Тестову перевірку можна провести також шляхом введення в програму, які перевіряється, операторів, які будуть сигналізувати про хід її виконання й отримання результатів.

*Тестові дані* слугують для перевірки роботи системи і складаються різними способами: генератором тестових даних, проектною групою на основі документів або наявних файлів, користувачем з специфікаціях вимог та ін. Дуже часто розробляються спеціальні форми вхідних документів, у яких відображається процес виконання програми за допомогою тестових даних.

Створюються тести, що перевіряють:

– повноту функцій;

– погодженість інтерфейсів;

– коректність виконання функцій і правильність функціонування системи в заданих умовах;

– надійність виконання системи;

– захист від збоїв апаратури і не виявлених помилок та ін.

Тестові дані готуються як для перевірки окремих програмних елементів, так і для груп програм або комплексів на різних стадіях процесу розроблення.

Багато типів тестів готуються замовником для перевірки роботи програмної системи. Структура і зміст тестів залежать від виду елемента тестування, яким може бути модуль, компонент, група компонентів, підсистема або система. Деякі тести залежать від мети і необхідності знати: чи працює система відповідно до її проекту, чи задоволені вимоги і чи бере участь замовник у перевірці роботи тестів тощо.

Залежно від задач, що ставляться перед тестуванням програм, складаються тести перевірки проміжних результатів проектування елементів на процесах ЖЦ, а також створюються тести іспитів остаточного коду системи.

**Тестування інтегрованої системи.**

Тести для перевірки окремих елементів системи і тести інтегрованої системи мають загальні і відмінні риси.

***Приклад.*** *Схема інтеграції компонентів*.

На першому рівні схеми знаходяться, наприклад, компоненти А, B, D, на другому рівні – E, C, G. Вони пов'язані між собою інтерфейсом (рис.1).

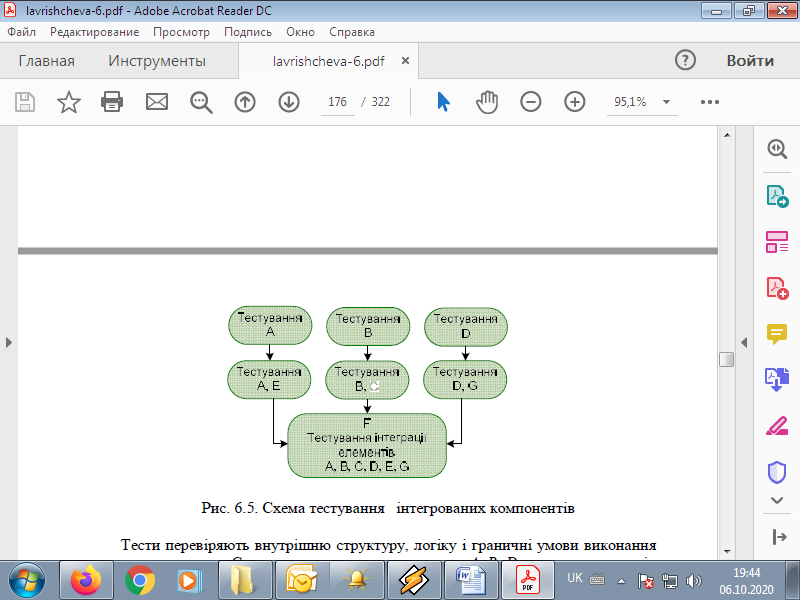


Рис. 1. Схема тестування інтегрованих компонентів

Кожен компонент схеми тестується окремо від інших компонентів тестами, що містять у собі набори даних і сценаріїв, складені відповідно до їхніх типів і функцій, специфікованих у вимогах до системи. Тестування проводиться в контрольному операційному середовищі на заданій безлічі тестових даних і операцій, розроблених з ними.

Тести перевіряють внутрішню структуру, логіку і граничні умови виконання кожного компонента. Спочатку тестуються компоненти *А, B*, *D* незалежно один від одного і кожний з окремим тестом. Після їхньої перевірки виконується перевірка інтерфейсів для зв'язку компонентів другого рівня: *Аº*→º*E*, *B* → *C*, *D* → *G,* а потім вже компоненти *Е*, C, *G*.

Компоненти й інтерфейси інтегруються і утворюють компонент *F,* він перевіряється на правильність інтеграції і функцій.

При тестуванні можуть виникати помилки. Вони, зазвичай, – результат неправильного завдання параметрів в операторах виклику або помилок в алгоритмі обчислення процедур або функцій. Помилки, що виникають у в зв'язках, усуваються, а потім повторно перевіряється зв'язок з компонентом *F* у вигляді трійки: компонент – інтерфейс – компонент.

Наступний крок тестування комплексної системи – перевірка функціонування системи за допомогою тестів перевірки функцій і вимог до них. Після цього перевіряється комплекс на виконавчих і іспитових тестах відповідно до вимог до ПС, апаратури і виконуваних функцій. Іспит системи проводиться в реальному середовищі, у якому система буде функціонувати надалі.

***Контрольні запитання***.

1. Надайте визначення *інфраструктури* перевірки правильності ПС?
2. Яким чином типи помилок розподіляються за процесами ЖЦ?
3. Наведіть приклад помилок на етапі виявлення вимог.
4. Які помилки виникають на процесі проектування?
5. Які помилки виникають на процесі кодування?
6. В чому причина появи помилок на етапі тестування?
7. В чому причина появи помилок на етапі супроводу?
8. Які кваліфікаційні вимоги висуваються до команди тестувальників?
9. Які дії виконуються на етапі планування тестуванням?
10. В чому полягає керування тестуванням?
11. Назвіть сутність інфраструктури організації робіт з тестування.
12. Надайте визначення *інфраструктури* перевірки правильності ПС?
13. Яким чином типи помилок розподіляються за процесами ЖЦ?
14. Наведіть приклад помилок на етапі виявлення вимог.
15. Які помилки виникають на процесі проектування?
16. Які помилки виникають на процесі кодування?
17. В чому причина появи помилок на етапі тестування?
18. В чому причина появи помилок на етапі супроводу?
19. Які кваліфікаційні вимоги висуваються до команди тестувальників?
20. Які дії виконуються на етапі планування тестуванням?
21. В чому полягає керування тестуванням?
22. Назвіть сутність інфраструктури організації робіт з тестування.

***Для самостійного вивчення***: Поглибити матеріал лекції за наданою літературою. Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

**Література**

1. Бабенко Л.П., Лавріщева К.М*.* Основи програмної інженерії.– Навч. посібник.–К.: Знання, 2001.– 269с.
2. И. Соммервиль. Инженерия программного обеспечения,  
    6 изд. – И.д. "Вильямс", 2002.
3. Лавріщева К.М. Програмна інженерія. Електронний підручник: http://csc.knu.ua/uk/library/books/lavrishcheva-6.pdf
4. Лавріщева К.М. Програмна інженерія. – Підручник.–К.:Академперіодика, 2008.–415с.
5. Тамре Л. Введение в тестирование программного обеспечения / Л. Тамре. –М.: Издательский дом «Вильямс», 2003. –368 с.
6. Сайкс Д. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие / Д.Сайкс, Д.Макгрегор. –К.: Диасофт, 2002. –432 с.
7. Фолк Д. Тестирование программного обеспечения / Д.Фолк, Е.К. Нгуен, С.Канер. – К.: Диасофт, 2003. –400 с.