# Лабораторна робота № 8.

## Техніки управління якістю ПЗ.

#### Короткі відомості

**Техніки управління якістю.** The TickIT Guide - «Використання ISO 9001:2000 для побудови систем менеджменту якості програмних продуктів, сертифікації та безперервного поліпшення». Цей стандарт розроблений професіоналами галузі з Європи та США, запрошеними для роботи над стандартом у складі спеціального комітету BRD/3/1 Британським Інститутом Стандартизації (BSI). Призначення стандарту - підвищити спроможність оцінок систем менеджменту підприємств - розробників програмних продуктів (IT-сектор) органами сертифікації. Якщо оцінка відповідності системи менеджменту проводиться фахівцями, не достатньо компетентними галузі розробки програмного забезпечення, то їх висновки щодо відповідності стандарту ISO 9001:2000 можуть виявитися невірними. Свою компетентність органи сертифікації систем менеджменту можуть підтвердити, демонструючи акредитацію послуг в IT-секторі за правилами «TickIT».

Крім додаткових вимог до органів сертифікації, «TickIT» містить керівництво щодо застосування ISO 9001:2000 при розробці програмного забезпечення, допомагаючи визначити:

* що є якість в контексті розробки програмних продуктів;
* як можна досягти якості;
* як система менеджменту може безперервно поліпшуватися.

«TickIT» гармонійно поєднує типові процеси розробників програмного забезпечення, детально описані в стандарті ISO / IEC 12207 і вимоги ISO 9001:2000. Результат цього об'єднання - ефективні технології менеджменту, описані з урахуванням специфічних процесів учасників IT-сектору.

Визначаючи дії, які споживачі повинні зробити, щоб забезпечити можливість постачання очікуваного програмного продукту, «TickIT» містить також керівництво з придбання програмних продуктів.

Наступні області IT-сектора виключені з розгляду в «TickIT»:

* Складування програмних продуктів,
* Продаж програмних продуктів через мережу роздрібної торгівлі,
* Установка програмних додатків на персональних комп'ютерах,
* Копіювання дисків та дискет, якщо це ізольований бізнес. У перерахованих випадках розробники стандарту вважають,
* що перевірка відповідності стандарту ISO 9001:2000 може бути проведена кваліфіковано органом сертифікації, які не мають акредитації «TickIT».

**Статичні техніки.** Техніки SQM можуть бути розподілені по декількох категоріях:

* статичні;
* техніки, що вимагають інтенсивного використання людських ресурсів;
* аналітичні;
* динамічні.

Статичні техніки передбачають детальне дослідження (examination) проектної документації, програмного забезпечення та іншої інформації про програмний продукт без його виконання. Ці техніки можуть включати інші, а саме дії з "колективної" оцінки або "індивідуального" аналізу, незалежно від ступеня використання засобів автоматизації.

Cтатіческіе методи використовуються при проведенні інспекцій та розгляді специфікацій компонентів без їхнього виконання. Техніка статичного аналізу полягає у методичному перегляді (або рецензуванні) і аналізі структури програм, а також у доведенні правильності. Статичний аналіз спрямований на аналіз документів, що розробляються на всіх етапах ЖЦ і полягає в інспекції вихідного коду та наскрізного контролю програми.

**Техніки колективної оцінки *(People-intensive techniques)***

"Техніки, які потребують інтенсивного використання людських ресурсів". Їх можна назвати і техніками "очних оцінок", тому що їхня ідея полягає саме у формі безпосередньої взаємодії фахівців. Ці техніки в перекладі названі "техніками колективної оцінки".

Форма такого роду техніки, включаючи оцінку і аудит, може варіюватися від формальних зборів до неформальних зустрічей або обговорення продукту навіть без звернення до його коду. Звичайно, такого роду техніки припускають взаємодію мінімум двох, а в більшості випадків, і більше фахівців.

При цьому, такі зустрічі можуть вимагати попередньої підготовки (практично завжди стосується визначення змісту зустрічей, тобто переліку питань, що виносяться на обговорення). До ресурсів, які використовуються в таких техніках, нарівні з досліджуваними артефактами (продуктом, документацією, моделями тощо,) можуть належати різного роду **листи перевірки** ***(checklists)*** і результати аналітичних технік та робіт з тестування.

Дані техніки розглядаються в стандарті 12207 при обговоренні рецензування (review) і аудиту (audit).

**Аналітичні техніки**

Інженери, які займаються програмним забезпеченням, застосовують аналітичні техніки. Іноді, кілька інженерів використовують одну й ту саму техніку, але стосовно різних частин продукту. Деякі техніки базуються на специфіці інструментальних засобів,які застосовуються, інші - припускають "ручну" роботу.

Ряд аналітичних технік також включає різного роду **експертизу *(assessment)*** як складовий елемент загального аналізуякості. Приклади таких технік:

* + аналіз складності (complexity analysis),
  + аналіз керуючої логіки (чи аналіз контролю потоків управління - control flow analysis) і
  + алгоритмічний аналіз (algorithmic analysis).

Кожен тип аналізу має конкретне призначення і не всі типи можна застосовувати до будь-якого проекту.

Прикладом **техніки підтримки** є аналіз складності, корисний для визначення фрагментів дизайну системи. Результат аналізу складності може також застосовуватися для розробки **тестових сценаріїв *(test cases)***.Такі техніки пошуку дефектів,як аналіз керуючої логіки, може також використовуватися і в інших випадках.

Для програмного забезпечення з обширною алгоритмічної логікою вкрай важливо застосовувати алгоритмічні техніки, особливо в тих випадках, коли некоректний алгоритм (не його реалізація, а саме логіка) може призвести до катастрофічних результатів (наприклад, програмне забезпечення авіоніки, для якої питання безпеки використання - safety грають вирішальну роль).

Більш формальні типи аналітичних технік, відомі як формальні методи. Вони застосовуються для перевірки вимог та дизайну. Перевірка коректності застосовується до критичних фрагментів програмного забезпечення (що мало пов'язане з формальними методами - це природний шлях досягнення прийнятної якості при мінімізації витрат). Найчастіше вони використовуються для верифікації особливо важливих частин критично-важливих систем, наприклад, конкректних вимог інформаційної безпеки та надійності.

**Динамічні техніки *(Dynamic techniques)***

1. В процесі розробки та супроводження програмного забезпечення доводиться звертатися до різних видів динамічних технік. В основному, це **техніки тестування*.*** Однак, в якості динамічних технік можуть розглядатися техніки симуляції, перевірки моделей та "посимвольного" виконання (symbolic execution, часто передбачає використання «пустих» (з точки зору виконуваної логіки) модулів з емульованим входом і виходом при розгляді загального сценарію поведінки багатомодульних систем).

Перегляд коду зазвичай розглядається як статична техніка, але досвідчений інженер може виконувати код безпосередньо "в процесі" його читання (наприклад, використовуючи діалогові засоби покрокового налагодження для ознайомлення або оцінки чужого коду).

Таким чином, ця техніка цілком може називатися і як динамічна. Такі розбіжності в класифікації технік ясно показують, що в залежності від ролі людини в організації, він може приймати і застосовувати одні й ті ж техніки по-різному.

* 1. залежності від організації ведення проекту, певні роботи з тестування можуть виконуватися при розробці програмних систем

1. SQA і V & V процесах.

**Робоче завдання**

1. Відкрити створений програмний код адресної книги, що був згенерований на попередній роботі.
2. Розробити план якості програмного продукту.
3. Виконати за допомогою статичної техніки аналіз якості ПЗ.
4. Розглянути динамічну техніку по символьного виконання.
5. Розглянути управління якістю програмного виробу.
6. Надати визначення якості процесу програмного забезпечення (Software engineering process quality).

Контрольні питання:

1. Описати процеси управління якістю програмного забезпечення (Software Quality Processes), навести приклади ефективного та неефективного управління.
2. Описати підхід всебічного управління якістю (Total Quality Management).
3. Пояснити, навіщо і в яких випадках використовують техніки колективної оцінки програмних продуктів (People-intensive techniques).
4. Надати визначення статичним технікам (Static techniques) забезпечення якості програмних систем, пояснити в чому полягає різниця між статичними і динамічними техніками (Dynamic techniques).
5. Проілюструвати відмінність між поняттями якість програмного продукту (Software product quality) та якість процесу (Software engineering process quality).
6. Описати типові процеси по управлінню якістю ПЗ (Software Quality Processes).
7. Описати типовий план якості (SQA-план), навести приклад.