

Тестування програмного забезпечення

Викладач: к.т.н. Шитікова Олена Вікторівна

Лекція 10

ТЕСТ-ПЛАНИ

ТЕСТ-ПЛАНИ

Тест-план (Test Plan) – це документ, що описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладів, критеріїв початку та закінчення тестування, вимог до необхідного в процесі роботи обладнання, спеціальних знань, а також оцінки ризиків з варіантами їх вирішення.

Тест-план є важливою складовою будь-якого грамотно організованого процесу тестування, оскільки містить в собі всю необхідну інформацію, що описує даний процес.

Цілі створення тест-плану

Переваги тест-плану:

- узгодження обсягів та стратегії тестування різних складових тестованого ПЗ з іншими учасниками проєктної команди;
- пріоритезація завдань з тестування;
- своєчасне планування ресурсовитрат на тестування;
- облік ресурсів (ПЗ, обладнання), необхідних для тестування;
- завчасний облік ризиків, які можуть виникнути в процесі реалізації плану, і впровадження попереджувальної стратегії.

Рекомендації з написання тест-плану

Хороший тест-план повинен, як мінімум, описувати:

1. Об'єкт тестування: системи, додатки, обладнання.
2. Список функцій та опис тестованої системи та окремо її компоненти.
3. Стратегія тестування: види тестування та їх застосування по відношенню до об'єкта тестування.
4. Послідовність проведення робіт: підготовка (Test Preparation), тестування (Testing), аналіз результатів (Test Result Analysis) у розрізі запланованих фаз розробки.

Рекомендації з написання тест-плану

5. Критерії початку тестування:

- готовність тестової платформи (тестового стенду);
- завершеність розробки необхідного функціоналу;
- наявність всієї необхідної документації.

6. Критерії завершення тестування – результати тестування задовольняють критерії якості продукту:

- вимоги до кількості відкритих багів виконані;
- витримка певного періоду без зміни початкового коду додатка Code Freeze (CF);
- витримка певного періоду без відкриття нових багів Zero Bug Bounce (ZBB).

Рекомендації з написання тест-плану

7. Оточення тестованої системи (опис програмно-апаратних засобів);

8. Необхідне для тестування обладнання та програмні засоби (тестовий стенд та конфігурація, програми для автоматизованого тестування тощо);

9. Ризики та шляхи їх вирішення.

Види тест-планів

- **майстер тест-план (Master Plan or Master Test Plan)**, який створюється у двох випадках:
 - якщо продукт має безліч релізів або ітерацій, між якими зберігається загальна інформація, яку немає сенсу повторювати;
 - якщо різні тестові команди працюють над одним продуктом, виконуючи різні завдання, які необхідно об'єднати в рамках одного документа.

Види тест-планів

При цьому в майстер тест-плані міститься інформація:

- загальна інформація про продукт, посилання на документацію, багтрекер та інші проєктні ресурси;
- загальні правила тестування: вимоги до дефектів, які заводяться, умови прийняття збірки на тестування;
- критерії готовності продукту до випуску, метрики якості;
- інструменти та техніки, які були використані.

Види тест-планів

- **тест-план (Test Plan)**, або детальний тест-план, який створюється до кожного релізу/ітерації або для кожної команди в рамках проєкту.

Основна мета – коротко та доступно відобразити завдання тестування. При цьому в детальному тест-плані міститься інформація:

- перелік областей тестування з пріоритетами;
- стратегія тестування;
- проєктні ризики;
- ресурси, необхідні для виконання завдань;
- проєктний план (терміни готовності основних завдань).

Види тест-планів

- **план приймального тестування (*Product Acceptance Plan*)** – документ, що описує набір дій, пов'язаних з приймальним тестуванням (стратегія, дата проведення, відповідальні працівники тощо).

У випадку з невеликим проектом створюється один тест-план, що містить у собі зведену інформацію з майстер тест-плану та детального тест-плану.

Стандарти тест-планів

Кожна методологія або процес пропонує свої формати представлення планів тестування.

Існує два міжнародних стандарти опису планів тестування:

- стандарт IEEE 829 (Institute of Electrical and Electronics Engineers);
- стандарт RUP (Rational Unified Process).

Стандарт IEEE 829

1. Ідентифікатор тест-плану (*Test Plan Identifier*)

– унікальний номер для ідентифікації документа всередині компанії.

Цей номер може вказувати на те, чи є документ майстер тест-планом або детальним тест-планом, а також він вказує і на ітерацію (етап) тестування, на якому застосовується.

Це дозволяє полегшити та систематизувати організацію процесу тестування.



Стандарт IEEE 829

2. Посилання (*Preferences*) – список усіх документів, які підтримують тест-план. Цей розділ призначений для того, щоб не копіювати частини тексту з перерахованих документів та не створювати надлишок інформації в тест-плані. Можуть бути згадані такі документи:

- план проєкту;
- технічні вимоги;
- дизайн-документ високого рівня;
- деталізований дизайн-документ;
- стандарти розробки і тестування;
- принципи та методології тестування;
- корпоративні стандарти і принципи.

Стандарт IEEE 829

3. Вступ (Introduction) – містить опис мети тест-плану та його рівня (майстер тест-план або детальний тест-план). По суті, це узагальнювальна частина всього документу, в якій у загальних рисах описується, що саме буде протестовано й які методи будуть для цього використані.

4. Об'єкти тестування (Test Items) – ті речі, які будуть протестовані в рамках даного тест-плану. По суті, це список того, що необхідно протестувати. Такий список можна скласти, виходячи з компонентів програмного продукту або ж з документації. Інформація даного розділу також містить номери версій тестованого продукту та вимоги до конфігурації (якщо необхідно).

•

•

Стандарт IEEE 829

5. Ризики в програмному забезпеченні (Software Risk Issues). Необхідно визначити критичні аспекти у тестуванні даного програмного забезпечення, такі як:

- надання продукта третьою стороною;
- нова версія пов'язаного програмного забезпечення;
- можливість використання нових інструментів;
- надзвичайно складні функції;
- зміни в компонентах, які раніше містили критичні помилки;
- безпека;
- кілька інтерфейсів;
- вплив на клієнта;
- державне регулювання та правила.

Стандарт ІЕЕЕ 829

6. Функції, які будуть протестовані – список того, що буде перевірено, що з точки зору користувача виконує система. Це не технічний опис програмного забезпечення, а опис тестованих функцій з боку кінцевого користувача системи.

Для кожної функції слід встановити рівень ризику (використовуючи, наприклад, просту шкалу High, Medium, Low). При формуванні тест-плану необхідно бути готовим пояснити, чому саме цей рівень ризику був обраний для конкретної функції.

Стандарт IEEE 829

7. Функції, які не будуть протестовані – список того, що не буде протестовано як з точки зору користувачів, так і з точки зору управління конфігураціями.

Необхідно визначити, чому ті чи інші функції виключені з тестування – цьому може бути кілька причин:

- функція не буде включена до поточного релізу ПЗ;
- низький рівень ризику, функція була протестована раніше та вважається стабільною;
- функція увійде до релізу, але не буде задокументована як частина релізу нової версії ПЗ.

Стандарт ІЕЕЕ 829

8. Підхід (Стратегія) – загальна стратегія тестування тест-плану, яка повинна відповідати рівню (виду) плану (майстер, приймальний та ін.) й повинна бути узгоджена з усіма вищими та нижчими рівнями плану.

Повинні бути зазначені загальні правила й процеси:

- Чи будуть використовуватися спеціальні інструменти й які саме?
- Чи потребують інструменти додаткового вивчення?
- Які метрики (показники) будуть зібрані в ході тестування?
- На якому з рівнів буде зібраний кожен показник?
- Як буде здійснюватися управління конфігурацією?

Стандарт ІЕЕЕ 829

- Яка кількість конфігурацій буде протестована?
- Апаратне забезпечення.
- Програмне забезпечення.
- Комбінації програмного та апаратного забезпечення.
- Які рівні регресійного тестування будуть проведені й в якій кількості?
 - Регресійне тестування буде базуватися на серйозності чи на знайдених дефектах?
 - Яким чином слід обробляти елементи вимог та дизайну у випадку, якщо їх неможливо протестувати?

Стандарт ІЕЕЕ 829

При складанні майстер тест-плану також повинна бути позначена загальна стратегія тестування всього проєкту та покриття вимог. Вкажіть, чи є особливі вимоги до тестування:

- будуть протестовані тільки цілі компоненти;
- зазначена група елементів (компонентів) повинна бути протестована разом.

Стандарт ІЕЕЕ 829

9. Критерій «Пройдено/Провалено» (*Item Pass/Fail Criteria*) – кожен тест-план повинен чітко визначати критерії завершення тестування:

- на рівні компонентного тестування (unit test level) критерії можуть бути наступними:

- всі тест-кейси виконані;
- вказану кількість (у відсотках) тест-кейсів виконано;
- вказану кількість (у відсотках) тест-кейсів виконано із зазначеним допустимим відсотком дрібних (незначних) дефектів;
- інструмент покриття коду (code coverage tool) визначає, що весь код покритий.



Стандарт ІЕЕЕ 829

- на рівні майстер тест-плану критерії можуть бути наступними:

- всі плани нижніх рівнів виконані;
- вказану кількість планів виконано без помилок і певний відсоток – з дрібними дефектами.

Дефект та збій в рамках тест-плану можна визначити наступним чином:

- дефектом є те, що може викликати збій і може бути прийнятним, щоб залишатися в додатку;
- збій є результатом дефекту, видимим для користувача.



Стандарт ІЕЕЕ 829

10. Критерії припинення та поновлення тестування, коли необхідно вказати, за яких умов та в яких ситуаціях слід припинити або продовжити виконання тестування.

Найчастіше таким критерієм може бути загальна кількість дефектів (або кількість дефектів певного типу), після досягнення якого, подальше тестування не має сенсу. Таким чином, необхідно визначити певну допустиму кількість дефектів, незважаючи на яку, тестування слід продовжувати. Продовження тестування після виникнення критичної помилки може створити умови, які можуть бути ідентифіковані як дефекти, які насправді є наслідком проігнорованої раніше помилки.

Стандарт ІЕЕЕ 829

11. Що надається в ході тестування (*Test deliverables*), а саме: «що буде надано як частина тест-плану?»:

- документ тест-плану;
- тест-кейси;
- специфікації тест-дизайну;
- інструменти та результат роботи;
- емулятори;
- статичні та динамічні генератори;
- логи помилок та логи виконання програм;
- звіти про проблеми та коригувальні дії.

Стандарт ІЕЕЕ 829

12. Майбутні тестові завдання, якщо процес тестування проходить у кілька етапів або розробка продукту ведеться шляхом інкрементного додавання версій, тест-план може не охоплювати деякі частини програмного продукту.

Ці частини повинні бути описані в тест-плані, щоб уникнути тестування ще не готових частин програми, й, як наслідок, створення дефектів по функціоналу, який ще не готовий до тестування. Якщо програмне забезпечення розробляється третьою стороною, у цьому розділі може міститися опис завдань як для зовнішніх, так і для внутрішніх груп тестувальників.

Стандарт ІЕЕЕ 829

13. Потреби середовища. У цьому розділі перераховуються специфічні вимоги для даного тест-плану:

- спеціальне програмне забезпечення (симулятори, статичні генератори тощо);
- яким чином надаються тестові дані та чи повинні надаватися специфічні набори даних або специфічний діапазон значень?
- який обсяг тестування буде виконаний для кожного компонента системи?
- спеціальні вимоги до потужності;
- специфічні версії допоміжного програмного забезпечення;
- обмежене використання системи під час тестування.

Стандарт ІЕЕЕ 829

14. Потреби у персоналі та навчанні.

- Навчання роботі з програмою або системою.
- Навчання роботі з інструментами для тестування.
- Хто несе відповідальність за навчання персоналу.

Стандарт ІЕЕЕ 829

15. Відповідальність. Хто за що відповідає? Цей розділ включає всі аспекти тест-плану, наприклад:

- оцінка ризиків;
- вибір функцій, які будуть і не будуть протестовані;
- вибір загальної стратегії тестування для тест-плану даного рівня;
- перевірка наявності всіх необхідних для тестування елементів;
- регулювання конфліктів у розкладі, особливо, коли тестування виконується на виробничій системі;
- Хто забезпечує необхідне навчання?
- Хто приймає критичні рішення по об'єктах, які не покриває даний тест-план?

Стандарт IEEE 829

16. Розклад, формування якого повинно ґрунтуватися на реальних та затверджених термінах.

Неточна оцінка часу тестування може змістити терміни виконання загального плану розробки програмного забезпечення, оскільки тестування є його невід'ємною частиною.

Рекомендується вказувати терміни проведення тестування з прив'язкою до дати закінчення етапу розробки ПЗ. Це захистить команду тестувальників від сприйняття їх як винуватців несвоєчасного випуску ПЗ у тому випадку, якщо затримка сталася з вини розробників.

Стандарт ІЕЕЕ 829

17. Планування ризиків та непередбачених обставин.

Які загальні ризики на проєкті з акцентом на процес тестування:

- відсутність кадрових ресурсів перед початком тестування;
- відсутність доступу до необхідного апаратного або програмного забезпечення, даних або інструментів;
- несвоєчасне надання програмного, апаратного забезпечення або інструментів;
- затримки в навчанні роботі з додатком і/або інструментами;
- зміна оригінальних вимог або дизайну.

Стандарт IEEE 829

Зазначають, що необхідно зробити у разі виникнення різних ситуацій в процесі тестування, наприклад:

- до команди тестувальників будуть додані допоміжні ресурси;
- команда тестувальників буде працювати понаднормово;
- мета плану може бути змінена;
- може бути проведена оптимізація ресурсів.
- визначення вимог буде завершено 1.01.20XX. Якщо вимоги будуть змінені після цієї дати:
 - розклад тестування та розклад розробки буде зміщено на відповідну кількість днів – відбувається рідко, оскільки більшість проєктів має фіксовані терміни випуску;

Стандарт ІЕЕЕ 829

– кількість проведених тестів буде зменшено – може знизити загальний рівень якості продукту, що випускається;

– кількість допустимих дефектів буде збільшено – може знизити загальний рівень якості продукту, що випускається.

Слід зауважити, що, якщо взагалі нічого не вжити з наведеного списку, то тестування програмного продукту буде скорочено або взагалі опущено, що, звичайно ж, негативно позначиться на якості розробленого ПЗ.

Стандарт IEEE 829

18. Затвердження: хто має право підтвердити, що процес тестування завершений, і дозволити перехід на наступний рівень у розробці проєкту (залежно від рівня плану).

На рівні майстер тест-плану це можуть бути всі зацікавлені сторони. При визначенні процесу затвердження необхідно враховувати хто входить до цільової аудиторії:

- аудиторія плану рівня модульного тестування відрізняється від плану рівня інтеграційного, системного тестування та рівня майстер тестплану;
- рівні та типи знань також будуть відрізнятися на різних рівнях тестпланів;

Стандарт IEEE 829

- програмісти дуже технічні, але часто можуть не мати чіткого уявлення про бізнес-процес та управління проектом;
- люди можуть мати різний рівень ділової хватки та дуже низький технічний рівень;
- необхідно бути обережними з користувачами, які заявляють про високий рівень технічної підготовки, й з програмістами, які заявляють про повне розуміння бізнес-процесу.

19. Глосарій: використовується для визначення термінів та скорочень, що використовуються у документі й тестуванні в цілому, для того, щоб не допустити плутанини та сприяти полегшенню комунікацій на проекті.



Стандарт RUP

1. Вступ.

1.1. Мета.

Опишіть існуючу інформацію про проєкт та програмні компоненти, які повинні бути перевірені, рекомендовані вимоги для тестування й загальну стратегію тестування. Також повинні бути визначені необхідні ресурси й надані елементи для проєкту.

1.2. Передумови (Background).

У цьому розділі необхідно дуже коротко описати основний функціонал та завдання. Необхідно включити таку інформацію, як головні функції системи, її архітектуру та історію проєкту.



Стандарт RUP

1.3. Тестова область (Test Scope).

- опишіть стадії тестування (компонентне, інтеграційне, системне) та типи тестування (наприклад, функціональне, навантажувальне тощо);
- наведіть короткий список тих функцій системи, що будуть / не будуть протестовані;
- перерахуйте будь-які пропозиції, що виникли в процесі написання документа;
- перерахуйте ризики та непередбачені обставини, які можуть виникнути в ході розробки та реалізації тестів;
- перерахуйте обмеження, які можуть вплинути на проєктування, розробку та реалізацію тестів.

Стандарт RUP

1.4. Ідентифікація проєкту.

Необхідно перерахувати наявну документацію по проєкту та доступність на момент написання даного тест-плану:

- специфікація вимог;
- специфікація функціоналу;
- звіти про використання;
- план проєкту;
- специфікація дизайну;
- прототип;
- посібники користувача;
- бізнес-модель;
- бізнес-функції та правила;
- оцінювання ризиків.

Стандарт RUP

2. Вимоги для тестування – список, який представляє перелік того, що необхідно протестувати, в якому необхідно описати функціональні та нефункціональні вимоги, які були позначені як цілі тестування.

3. Стратегія тестування – рекомендований підхід до тестування. У попередньому розділі пояснено, що буде протестовано, а у цьому розділі – якими методами тестування буде виконуватися.

Для кожного типу тестування необхідно навести докладний опис й позначити, для чого даний вид тестування виконується. Основними складовими стратегії є техніки тестування, які будуть використовуватися, й критерії завершення тестування.

Стандарт RUP

3.1 Типи тестування. У кожному з пунктів необхідно визначити об'єкт тестування, техніку (методи), за допомогою яких буде проводитися даний вид тестування, критерій проходження тесту. Також опис може включати інформацію про особливості проведення даного виду тестування (якщо такі є).

3.2 Тестування цілісності даних та баз даних. Бази даних та процеси мають бути протестовані, як підсистема у рамках проєкту без взаємодії з інтерфейсом користувача. Додаткові дослідження в системі управління базами даних необхідні для того, щоб визначити засоби та методи, за допомогою яких буде виконуватися даний вид тестування.

Стандарт RUP

3.3 Тестування бізнес-циклу передбачає імітацію діяльності, здійснюваної на проєкті протягом довгого часу. Повинен бути визначений період, наприклад, один рік, і всі транзакції та дії, які виконуються над проєктом протягом даного проміжку часу, повинні бути виконані.

3.4 Тестування інтерфейсу користувача (UI Testing) виконується перевірка взаємодії користувача з програмним забезпеченням. Мета – переконатися, що інтерфейс надає користувачеві необхідний доступ та функції навігації в системі. Крім того, тестування інтерфейсу користувача гарантує, що всі об'єкти працюють згідно з очікуваннями та відповідають корпоративним стандартам та стандартам галузі.

-
-

Стандарт RUP

3.5 Профілювання продуктивності – тестування продуктивності, при якому вимірюються часово-залежні вимоги такі, як час відгуку, швидкість виконання операцій та ін. Метою даного виду тестування є підтвердження того, що всі вимоги продуктивності виконані. Профілювання продуктивності реалізується та виконується для окремих профілів з налаштовуваною поведінкою та умовами середовища такими, як конфігурація устаткування й робоче навантаження.

Стандарт RUP

3.6 Навантажувальне тестування – тест

продуктивності, який піддає систему різним навантаженням для оцінки продуктивності та здатності системи продовжувати коректну роботу під даними навантаженнями. Мета такого тестування полягає в тому, щоб визначити та гарантувати коректну роботу системи при навантаженні понад максимально очікуваного рівня.

Стандарт RUP

3.7 Стрес-тестування – тестування продуктивності, яке реалізується та виконується з метою виявлення помилок, пов'язаних з недостатністю ресурсів. Недостатня кількість пам'яті або місця на жорсткому диску може спричинити виникнення помилок, які не проявляються в умовах, коли системних ресурсів достатньо. Також дефекти можуть виникати при використанні загальних ресурсів таких, як бази даних або пропускна здатність мережі.

Стрессове тестування може також проводитися з метою визначення рівня максимального навантаження, при якому система продовжує виконувати встановлені функції.



Стандарт RUP

3.8 Об'ємне тестування: завдання якого полягає в отриманні оцінки продуктивності при збільшенні обсягів даних у базі даних програми. При цьому відбувається:

- вимірювання часу виконання обраних операцій при певній інтенсивності виконання цих операцій;
- може здійснюватися визначення кількості користувачів, що одночасно працюють з додатком.

3.9 Тестування безпеки та контролю доступу зосереджено на двох основних сферах безпеки:

- безпека рівня додатків, включаючи доступ до даних або бізнес-функцій;
- безпека рівня системи, включаючи авторизацію або віддалений вхід у систему.

Стандарт RUP

3.10 Тестування відмовостійкості та відновлення.

У ході проведення даного виду тестування перевіряється стійкість системи до програмних та апаратних збоїв та здатність до відновлення після таких збоїв без втрати інформації.

Прикладами таких збоїв можуть бути:

- відключення живлення клієнта;
- відключення живлення сервера;
- порушення мережевого з'єднання;
- пошкоджений елемент у базі даних та ін.

Стандарт RUP

3.11 Тестування конфігурації. Перевіряє роботу системи, яка підлягає тестуванню на різних конфігураціях програмного та апаратного забезпечення.

3.12 Тестування установки. По-перше, даний вид тестування виконується для того, щоб переконатися, що ПЗ коректно встановлюється за різних умов: нова установка, оновлення, повна або вибіркова установка. Також перевіряється установка ПЗ в умовах нестачі системних ресурсів (наприклад, місця на жорсткому диску).

По-друге, необхідно переконатися, що одного разу встановлене ПЗ коректно функціонує при наступних запусках.



Стандарт RUP

3.13 Інструменти.

У цьому розділі необхідно перелічити всі інструменти, за допомогою яких буде виконуватися тестування, наприклад:

- система адміністративного управління тестуванням;
- система відстеження дефектів;
- інструменти для функціонального тестування;
- інструменти для тестування продуктивності;
- засіб моніторингу тестового покриття;
- засіб профілювання (профайлер);
- інструменти СУБД.

Стандарт RUP

4. Ресурси

У цьому розділі представлені рекомендовані ресурси для проєкту, основні обов'язки персоналу і набір знань та навичок.

4.1. Ролі. Цей пункт містить перелік посад усіх, хто задіяний на проєкті, та основні обов'язки. Найчастіше це:

- тест-менеджер;
- тест-дизайнер;
- тестувальникі;
- адміністратор тестової системи;
- адміністратор баз даних;
- дизайнер.

Стандарт RUP

4.2. Система.

Даний розділ містить опис системних ресурсів для тестування та може містити наступні елементи:

- сервер бази даних;
- тестові ПК клієнта;
- тестовий репозиторій;
- тестові ПК розробників.

Стандарт RUP

5. Основні етапи проєкту містять основні етапи тестування проєкту та терміни виконання, як правило, виділяють наступні етапи:

- планування тестування;
- тест-дизайн;
- реалізація тестів;
- виконання тестів;
- оцінка тестів.

Стандарт RUP

6. Практичні результати тестування. У цьому розділі описують різні документи, інструменти та звіти, які будуть створені, а також для кого, ким й коли будуть створені:

- модель тестування описує звіти та інші артефакти, які будуть створені та розподілені в рамках тестової моделі;
- тестові журнали (логи) описують методи та інструменти, які використовуються для запису та створення звітів;
- звіти про помилки описують методи та інструменти для написання звітів про помилки, управління та моніторингу статусу.

