
Управління проектами

ДЕНЬ 6



Тестування (продовження...)

Техніки, що базуються
на умовах
використання



Експлуатаційний профіль (Operational profile)

Базується на умовах використання системи.

Тестування для оцінки надійності системи має проводитися в такому тестовому оточенні, яке є максимально наближеним до реальних умов роботи системи

Результати таких тестів дозволяють оцінити поведінку системи в реальних умовах

Вхідні параметри тестів задаються на основі ймовірнісного розподілу відповідних параметрів або їх наборів при експлуатації (вхідні дані можуть прогнозуватися виходячи з частоти можливих сценаріїв роботи користувачів)

Тестування, яке базується на надійності інженерного процесу (Software Reliability Engineered Testing)

Базується на умовах розробки системи

Відповідні тести (що позначаються також аббревіатурою SRET від Software Reliability Engineered Testing) проектуються в контексті використовуваного процесу розробки і методик тестування

Техніки, що базуються на природі застосування

В залежності від технологічної чи архітектурної природи застосувань, можуть також застосовувати специфічні техніки, важливі саме для заданого типу додатка

Серед таких технік:

- Об'єктно-орієнтоване тестування
- Компонентно-орієнтоване тестування
- Web-орієнтоване тестування
- Тестування на відповідність протоколам
- Тестування систем реального часу

Функціональне та структурне тестування

Техніки тестування, що будуються на основі специфікацій або коду часто називають функціональними чи структурними, відповідно

Обидва підходи не повинні протиставлятися, але доповнювати один одного.

У функціональному тестуванні програма або система розглядається як чорна скринька

На входи подаються дані, а виходи перевіряються на відповідність зазначеній поведінці

Функціональне тестування здійснюється з точки зору користувача.

Функціональне та структурне тестування.2

Структурне ж тестування зазирає в деталі реалізації

Такі речі, як стиль програмування, метод управління, вихідна мова, проектування баз даних, кодування та деталі домінують у структурному тестуванні, але межа між функцією і структурою є розмитою

Хороші системи будуються по шарах, від зовнішнього до внутрішнього

Користувач бачить тільки верхній шар, шар чистої функції

Кожен шар всередині менш пов'язаний з функціями системи та більш обмежений структурою

Таким чином, те, що є структурою для одного шару, є функцією для іншого

Документування тестів

Документація - складова частина формалізації процесу тестування

Серед таких документів можуть бути:

- План тестування
- Специфікація процедури тестування
- Специфікація тестів
- Лог тестів
- Звіт з тестування

План тестування

1. Вступление

1.1 Цель

1.2 Сокращения и аббревиатуры

1.3 Ссылки

2. Требования для тестирования

2.1 Технические спецификации

2.2 Документация к продукту

3. Ресурсы

3.1 Распределение ролей в команде тестеров

4. Тестовая среда

4.1 Состав оборудования

4.1.1 Необходимое оборудование

4.1.2 Доступное оборудование

4.2 Программное обеспечение

4.2.1 Системное программное обеспечение

4.2.2 Дополнительное программное обеспечение

5. Объем тестирования

5.1 Тестовые наборы

5.2 Не тестируемые компоненты

6. Фазы тестирования проекта

7. Документы подлежащие поставке

8. Приложение

8.1 Виды тестирования

Специфікація тестів

| Версия | 1.001 | ПРОЕКТ | | | | | | | | | | | | | | | |
|----------------|-------------------|-------------------|----------------------|------------------|--------------------|--------------|----------------------|-------|--|--|--------------------|------------------------|----------------|--|------------------------|-----------|--|
| Разработчик | Ст | Тестер | Ирина Л. | | | | | | | | | | | | | | |
| Комментарий: | Test Cases | | | | | | | | | Версия 0.0.0.1 | | | Версия 0.0.0.2 | | | | |
| Case# or Bug # | Создан (Кем/Дата) | Текущее состояние | Ссылка на требования | Тип тестирования | Компонент | Конфигурация | Краткое описание | Папка | Последовательность действий | Ожидаемый результат | Реальный результат | Тестирование(Кем/Дата) | Состояние | Реальный результат | Тестирование(Кем/Дата) | Состояние | |
| FC-KU-001 | Ирина Л./27.12 | Fail | | Functional | Конструктор уроков | | Добавление фрагмента | - | Вставляем диск с ППЗ в дисковод для чтения компакт-дисков. | Инсталлятор ППЗ запускается автоматически. | как ожидалось | Ирина Л./27.12 | Pass | все падает с критической ошибкой номер бага 456745 | Ирина Л./28.12 | Fail | |

Звіт з тестування

1.Общая информация

1.1Рамки тестирования

1.2Цели тестирования

2.Детальные результаты тестирования

2.1Результаты тестирования

2.2Выявленные проблемы

3.Заключение

Впровадження

Впровадження

Впровадження програмного забезпечення - процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним

Основні задачі

Планування впровадження

Підготовка операційного середовища системи

Навчання користувачів

Міграція системи на продуктивне оточення

Підготовка та проведення випробувань

Дослідна та промислова експлуатація

Планування впровадження

Розробка та узгодження планів робіт з впровадження системи

Підготовка операційного середовища системи

Підготовка матеріальних, технічних засобів для функціонування системи

Закупівля обладнання, програмного забезпечення, матеріалів

Проведення необхідних робіт

Підготовка організаційної структури для впровадження системи

Розробка документації

Навчання користувачів

Планування навчання користувачів системи

Розробка тренінгів

Проведення тренінгів

Міграція системи на продуктивне оточення

Проведення міграції системи

Наповнення необхідними даними

Підготовка та проведення випробувань

Планування приймальних випробувань

Проведення випробувань

Доопрацювання системи

Дослідна та промислова експлуатація системи

Проведення дослідної експлуатації

Доопрацювання системи по результатах дослідної експлуатації

Проведення випробувань

Передача у промислову експлуатацію

Супровід

Супровід

Супровід програмного забезпечення як модифікація програмного продукту після передачі в експлуатацію для усунення збоїв, поліпшення показників продуктивності та / або інших характеристик (атрибутів) продукту, або адаптації продукту для використання в модифікованому оточенні

Супровід підтримує функціонування програмного продукту протягом усього операційного життєвого циклу, тобто періоду його експлуатації

У процесі супроводу фіксуються і відслідковуються запити на модифікацію (також називаються "запитами на зміни" - change requests, оцінюється вплив пропонованих змін, модифікується код та інші активи (артефакти) продукту, проводиться необхідне тестування і, нарешті, випускається оновлена версія продукту

Крім того, проводиться навчання користувачів і забезпечується їх щоденна підтримка при роботі з поточною версією продукту.

Завдання супроводу

У загальному випадку, роботи з супроводу повинні проводитися для вирішення наступних завдань:

- усунення збоїв
- поліпшення дизайну
- реалізації розширень <функціональних можливостей>
- створення інтерфейсів взаємодії з іншими (зовнішніми) системами
- адаптації (наприклад, портації) для можливості роботи на іншій апаратній платформі (або оновленій платформі), застосування нових системних можливостей, функціонування в середовищі оновленої телекомунікаційної інфраструктури і т.п.
- міграції успадкованого (legacy) програмного забезпечення
- виведення програмного забезпечення з експлуатації

Діяльність персоналу супроводу

Діяльність персоналу супроводу включає чотири ключових аспекти:

- підтримка контролю (керованості) програмного забезпечення протягом усього циклу експлуатації
- підтримка модифікацій програмних систем
- вдосконалення існуючих функцій
- запобігання падінню продуктивності програмної системи до неприйняттого рівня

Категорії супроводу (Categories of Maintenance)

Виділяють чотири категорії супроводу:

Коригувальний супровід (corrective maintenance)

Адаптивний супровід (adaptive maintenance)

Удосконалюючий супровід (perfective maintenance)

Профілактичний супровід (preventive maintenance): модифікація програмного продукту на етапі експлуатації для ідентифікації та запобігання прихованих дефектів до того, коли вони приведуть до реальних збоїв.

Коригувальний супровід

Коригувальний супровід – це "реактивна" модифікація програмного продукту, виконувана вже після передачі в експлуатацію для усунення збоїв

Адаптивний супровід

Адаптивний супровід – це модифікація програмного продукту на етапі експлуатації для забезпечення продовження його використання з заданою ефективністю (з точки зору задоволення потреб користувачів) у зміненому оточенні або оточенні, що знаходиться в процесі змін; в першу чергу, мається на увазі зміна бізнес-оточення, що породжує нові вимоги до системи

Удосконалюючий супровід

Удосконалюючий супровід - це модифікація програмного продукту на етапі експлуатації для підвищення характеристик продуктивності і зручності супроводу

Профілактичний супровід

Профілактичний супровід - це модифікація програмного продукту на етапі експлуатації для ідентифікації та запобігання прихованих дефектів до того, коли вони приведуть до реальних збоїв.

Проактивний та реактивний підхід

| | Корректирующие работы | Работы по расширению |
|----------------------|--------------------------------|--------------------------------|
| “Проактивный” подход | Профилактическое сопровождение | Совершенствующее сопровождение |
| “Реактивный” подход | Корректирующее сопровождение | Адаптирующее сопровождение |

Роботи з супроводу (Maintenance Activities)

Багато робіт з супроводу схожі на аспекти діяльності з розробки

В обох випадках необхідно проводити аналіз, проектування, кодування, тестування і документування

Фахівці з супроводу повинні відслідковувати вимоги так само, як і інженери-розробники, та оновлювати документацію по мірі розробки та / або випуску оновлених або нових релізів продукту

У той же час, діяльність з супроводу має певні унікальні риси, що призводить до необхідності використання спеціалізованих процесів.

Унікальні роботи та практики супроводу

Передача (Transition): контрольована і координована діяльність з передачі програмного забезпечення від розробників групі, службі або організації, що відповідає за подальшу підтримку;

Ухвалення / відхилення запитів на модифікацію (Modification Request Acceptance / Rejection): запити на зміни можуть як прийматися і передаватися в роботу, так і відхилятися з різних обґрунтованих причин - обсягу та / або складності необхідних змін, а також необхідних для цього зусиль

Відповідні рішення можуть також прийматися на основі пріоритетності, оцінці обґрунтованості, відсутності ресурсів (в тому числі, відсутності можливості залучення розробників до вирішення завдань з модифікації, при реальній наявності такої потреби), затвердженого плану реалізації в наступному релізі і т.п.

Унікальні роботи та практики супроводу.2

Засоби фіксації, комунікації та відстеження статусу запитів на модифікацію або інцидентів (Help Desk): функція підтримки кінцевих користувачів, яка підтримує роботи з оцінки (assessment), аналізу пріоритетності та вартості модифікацій, пов'язаних з запитом, що надійшов, або повідомленою проблемою

Аналіз впливу (Impact Analysis): аналіз можливих наслідків змін, що вносяться в існуючу систему.

Унікальні роботи та практики супроводу.3

Підтримка програмного забезпечення (Software Support): роботи з консультування користувачів, що проводяться у відповідь на їх інформаційні запити (request for information), наприклад, що стосуються відповідних бізнес-правил, перевірки, змісту даних і специфічних питань користувачів і їх повідомлень про проблеми (помилки, збої, непередбачених поведінки, нерозуміння аспектів роботи з системою і т.п.)

Контракти і зобов'язання: до них відносяться класична угода про рівень сервісу - Service Level Agreement (SLA), а також інші договірні аспекти, на підставі яких, група / служба / організація з супроводу виконує відповідні роботи

Техніки супроводу (Techniques for Maintenance)

Розуміння програмних систем (Program Comprehension)

Реінжиніринг (Reengineering)

Зворотній інжиніринг (Reverse engineering)

Розуміння програмних систем

Для реалізації змін програмісти витрачають значну частину часу на читання і формування розуміння програмного продукту

Засоби роботи з кодом є ключовим інструментом для вирішення цього завдання

Чітка, однозначна і лаконічна документація забезпечує адекватне розуміння програмних систем

Реінжиніринг

Реінжиніринг визначається як детальна оцінка (examination) і перебудова програмного забезпечення для формування розуміння, відтворення (на рівні моделі і, в ряді випадків, вимог) та подальшої реалізації його функцій в новій формі (наприклад, з використанням нових технологій і платформ, при збереженні існуючої і розширенням та полегшенням можливостей додавань нової функціональності)

В індустрії існують різні позиції щодо реінжинірингу - одні вважають, що реінжиніринг є найбільш радикальною і витратною формою змін програмних систем, інші, що такий підхід може застосовуватися й для не настільки кардинальних змін (наприклад, як зміна платформи чи архітектури)

Реінжиніринг, зазвичай, проводиться не стільки для покращення можливостей супроводу (maintainability), скільки для заміни застарілого програмного забезпечення

В принципі, реінжиніринг можна розглядати як самостійний проект, що включає в себе формування концепції, застосування відповідних інструментів і технік, аналіз і застосування досвіду проведення реінжинірингу, а також оцінку ризиків і переваг, пов'язаних з такими роботами

Реінжиніринг.2

Необхідно зазначити, що реалізація продукту в новій якості (формі) при збереженні основної функціональності оригінального продукту, є невід'ємною і визначальною частиною реінжинірингу, на відміну від зворотного інжинірингу, розглянутого нижче, та є важливою складовою частиною реінжинірингу

Зворотній інжиніринг

"Зворотній" інжиніринг - це процес аналізу програмного забезпечення з метою ідентифікації програмних компонент і зв'язків між ними, а також формування уявлення про програмне забезпечення, з подальшою перебудовою в новій формі (вже в процесі реінжинірингу)

Зворотний інжиніринг є пасивним, припускаючи відсутність діяльності щодо зміни або створення нового програмного забезпечення

Зазвичай, в результаті зусиль з зворотного інжинірингу створюються моделі викликів (call graphs) і потоків управління (control flow graphs) на основі вихідного коду системи

Один з типів зворотного інжинірингу - створення нової документації на існуючу систему (redocumentation). Інший з поширених типів - відновлення проекту системи (design recovery).

Документація супроводу

SLA

Інструкції

Процеси