

Лекція 6. Парадигма якості в програмній інженерії

Основні поняття в області якості

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

Термін **програмне забезпечення** (ПЗ) використовується застосовано до сукупності програмних засобів, які розробляються з метою експлуатації у складі системи.

Спеціалісти-практики в області програмних систем і користувачі сходяться в поглядах на поняття поганого програмного продукту як такого, що:

- ⦿ Не забезпечує підтримку стратегії бізнесу або потреб користувача
- ⦿ Недостатньо надійний, гнучкий, ефективний і погано супроводжується
- ⦿ Є коштовним і занадто довго розробляється

Інфраструктура програмної інженерії –

інтегрований набір загальнодоступних технічних, технологічних і методологічних ресурсів організації розробника, які роблять можливим виконання процесу програмної інженерії колективами проектів, які відкриваються по договорах із замовниками.

Тут ***проект*** – це обмежена часовими рамками діяльність, мета якої полягає в створенні унікального програмного продукту.

Процес програмної інженерії – множина логічно пов'язаних видів діяльності по визначенню, проектуванню і побудові програмних продуктів (прикладних програмних систем).

Техніко-технологічний
аспект

Техніка і комунікації

Загальносистемне ПЗ та
інструменти

Інформаційні ресурси і
стандарти розробки ПЗ

Міжпроектна
програмна підтримка

Рівень організації

Інфраструктура
програмної
інженерії

Кадровий аспект

Навчання методам і
технологіям

Обмін позитивним та
негативним досвідом

Накопичення і
розповсюдження
позитивного досвіду

Стандарти між
проектної взаємодії

Базовий процес програмної інженерії

Базовий процес програмної інженерії



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 ЖЦ створюваної 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»

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 груп 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.

Архітектура процесів життєвого циклу

Процес програмної інженерії має ієрархічну структуру і включає множину процесів ЖЦ програмної системи.

Вимоги до процесів ЖЦ ПС визначає міжнародний стандарт ISO/IEC 12207, а в Україні йому відповідає ДСТУ 3918-99.

Процеси ЖЦ розподілені по трьом групам, які відображають функціональну направленість видів діяльності, які ці процеси регламентують:

- Основні процеси
- Процеси підтримки
- Організаційні процеси

Основні процеси ЖЦ:

◎ Придбання

- Підготовка придбання
- Вибір постачальника
- Моніторинг діяльності постачальника
- Прийом споживачем

◎ Поставка

- Участь в тендері
- Укладення договору
- Випуск продукту (релізу)
- Підтримка приймання продукту

◎ Розробка

- Виявлення вимог
- Аналіз вимог до системи
- Проектування архітектури системи
- Аналіз вимог до ПЗ системи
- Проектування ПЗ
- Програмування ПЗ
- Інтеграція ПЗ
- Тестування ПЗ
- Системна інтеграція
- Системне тестування
- Інсталяція ПЗ

◎ Експлуатація

- Функціональне використання
- Підтримка споживача

◎ Супровід

Процеси підтримки інтегруються з будь-якими іншими процесами, розв'язуючи задачі, допоміжні по відношенню до задач цих процесів, і забезпечують якість їх розв'язання в конкретних проектах.

- ⦿ Документування
- ⦿ Керування конфігурацією
- ⦿ Забезпечення гарантії якості
- ⦿ Верифікація
- ⦿ Валідація
- ⦿ Сумісний перегляд
- ⦿ Аудит
- ⦿ Керування розв'язанням проблем
- ⦿ Керування запитами на зміну
- ⦿ Забезпечення застосовуваності продукту (підтримка користувача)
- ⦿ Оцінювання проекту

Організаційні процеси ЖЦ

◎ Керування

- Організаційне будівництво (формування корпоративної політики, цілей, процесів, стандартів, етики)
- Управління організацією
- Управління проектом
- Управління якістю
- Управління ризиком
- Вимірювання

◎ Підтримка інфраструктури

- ◎ Вдосконалення
 - Установлення процесів
 - Оцінювання процесів
 - Покращення процесів
- ◎ Забезпечення трудовими ресурсами
 - Управління кадрами
 - Навчання
 - Управління знаннями (розповсюдження знань)
- ◎ Управління активами організації
- ◎ Управління програмою повторного використання
- ◎ Доменна інженерія

Базовий процес організації

Сучасною концепцією процесу програмної інженерії є побудова базового процесу організації (БПО), який розробляється, супроводжується, оцінюється і покращується подібно до того як розробляються програмні продукти.

БПО є основою для визначення процесів усіх програмних проектів. Процеси на рівні проектів розробляються шляхом адаптації БПО до характеристик конкретного проекту.

Визначення БПО – це формалізований опис складу процесів ЖЦ, з яких мають побудуватись процеси розробки в програмних проектах, а також взаємозв'язків між елементами цих процесів.

З кожним процесом розробки пов'язуються:

- ⦿ *Вимоги* до процесу, які вказують, «що» собою являє процес (що він буде робити)
- ⦿ *Архітектура* і проект процесу, які описують, «як» процес буде визначений (які будуть елементи процесу і як будуть пов'язані)
- ⦿ *Реалізація* опису процесу в рамках організації або програмного проекту (створення елементів процесу і встановлення інтерфейсу)
- ⦿ *Перевірка* і затвердження визначення процесу
- ⦿ *Впровадження* процесу в середовище конкретного проекту.

В мінімальній конфігурації процесів ЖЦ, крім процесу управління проектом, в БПО обов'язково має знайтись місце для процесу перевірки, а також процесу управління конфігурацією.

Побудований БПО має підтримувати використання моделей ЖЦ, застосування яких допускається в проектах організації.

Широко розповсюджені такі основні класи моделей ЖЦ:

- ◎ Каскадні
 - Стандартна
 - Із зворотнім зв'язком
 - Пилоподібна
- ◎ Ітераційні
 - З прирощуваннями
 - Еволюційні
 - Спіральна
 - Швидкої розробки програм (RAD)

Вибір моделі суттєво залежить від двох факторів:

А) чи можна спочатку визначити практично повний набір функцій, які необхідно реалізувати в програмному продукті

Б) чи мають усі жадані функції поставлятися замовнику одночасно

- ⦿ Якщо А і Б, то вибираємо каскадні моделі
- ⦿ А і не Б – вибирається ітераційна модель з прирощуваннями
- ⦿ Не А і Б, а також бажана розробка прототипів для моделювання вимог – спіральна модель
- ⦿ Не А і не Б – модель швидкої розробки програм, при умові, що строки розробки не будуть чітко встановлені.

Керування проектами

Застосовано до програмної інженерії види діяльності по управлінню утворюють дворівневу структуру:

- загальне організаційне управління,
- керування виконанням програмних проектів.

Керування проектом (в будь-якій галузі) – це область знань, навиків, інструментарію та прийомів для досягнення цілей проектів в рамках узгоджених параметрів якості, бюджету, строків та інших обмежень.

Сучасна концепція керування проектом основана на принципі інформативного вимірювання процесів ЖЦ проекту, його ресурсів і створюваних робочих продуктів.

Керування проектом включає наступні кроки:

1. **Ініціація проекту** і визначення його меж. Визначення вимог до проекту за допомогою застосування методів оцінювання вимог з різних точок зору: технічної, технологічної, фінансової, соціально-політичної

2. **Планування**. Передбачає:

- ⦿ Вибір моделі ЖЦ проекту і оцінювання процесів ЖЦ в контексті придатності для задовільнення вимог до проекту, адаптацію БПО.
- ⦿ Ієрархічну декомпозицію задач проекту, їх специфікацію поряд з встановленими вимогами, асоційованими робочими продуктами.
- ⦿ Оцінки об'ємів робіт, трудомісткості і вартості реалізації проекту
- ⦿ Розподіл ресурсів по задачам з врахуванням графіку проекту і з позицій раціонального використання персоналу проекту, обладнання та матеріалів.
- ⦿ Керування ризиком проекту по критеріям вартості розробки, тривалості проекту і якості програмних продуктів
- ⦿ Керування якістю - застосування процедур планування та контролю якості виконання процесів і будь-яких робочих продуктів цих процесів, а також верифікація та валідація продуктів.
- ⦿ Керування планом проекту. Необхідність такого керування обумовлена часто змінюваними вимогами замовника і умовами виконання проекту. Це робить процес планування проекту ітеративним

3. **Введення в дію.** Передбачає виконання обраних процесів ЖЦ у відповідності з планом поряд із змінами, моніторингом та регулюванням процесів і складанням звітів для зацікавлених сторін (керівництва організації, замовників, співвиконавців)
4. **Огляд і оцінка виконання проекту.** Передбачає виконання всеосяжної перевірки діяльності по проекту і оцінки його руху до встановлених цілей. Проводиться у встановлених критичних точках проекту. Оцінюються не тільки результати виконання процесів, але й ефективність застосованих методів та інструментів, а також продуктивність роботи колективу проекту і можливі труднощі. При необхідності здійснюється регулювання
5. **Закриття проекту.** Передбачає припинення проекту після завершення процесів та досягнення цілей. Оцінюється успішність проекту по відношенню до встановлених критеріїв. ПС передається в експлуатацію.

Процес вимірювання при керуванні проектами

Сучасний рівень розвитку програмної інженерії дозволяє вивести проблему якості ПС за рамки простого питання *«працює система чи ні?»* формулюючи її так: *«наскільки точно створена система задовольняє встановленим вимогам до її якості»*. Відкладувати пошук відповіді на це питання до моменту завершення розробки – занадто великий ризик як для замовника, так і для розробника. Вимірювання мають стати невід'ємною частиною ЖЦ проекту.

Вимірювання – отримання об'єктивних даних про стан продуктів, процесів та ресурсів розробки ПС з метою побудови прогнозуючих та оціночних моделей, які застосовуються для керування проектом і вдосконалення процесів організації.

Виконання вимірювань в проекті це багатокроковий процес, який включає наступні кроки:

1. *Визначення цілей програми вимірювання*
2. *Побудова процесу вимірювань.* Модель розроблюється таким чином, щоб забезпечити побудову точних і аргументованих відповідей на питання, підкріплених кількісними оцінками, оснований на вимірюваних величинах.
3. *Вибір базових мір.* В їх число як мінімум мають входити:
 - ⦿ Розмір і складність програмного продукту, на основі яких може бути оцінена трудомісткість та вартість розробки
 - ⦿ Продуктивність праці окремих спеціалістів і колективу проекту в цілому
 - ⦿ Міри вимірювань атрибутів якості

4. Організація збору даних для виконання вимірювань.
Дані, що збираються мають забезпечувати не тільки високі передбачувальні можливості вимірювань. Але й бути достатньо «дешевими» з точки зору їх отримання. Збір і обробка даних для вимірювань є трудомістким процесом, який вимагає підтримку керівників організації і додаткових інвестицій в розробку ПС.

5. Застосування моделей.

Підходи до підвищення якості програмних систем.

Перший крок полягає у впровадженні спеціально призначених для цього підтримуючих процесів, а саме процесів гарантування якості, верифікації, валідації, сумісного перегляду та аудиту.

- *Процес гарантування якості* (процес SQA) забезпечує гарантії. Що програмні продукти і процеси в життєвому циклі проекту відповідають вимогам. Ці гарантії основані на тому, що SQA планує і здійснює контроль і здійснення допомоги в тій діяльності по проекту, що безпосередньо пов'язана із «вбудовою» в програмні продукти тих властивостей, що забезпечують якість. SQA встановлює стандарти та контролює їх дотримання в ході ЖЦ. Мета SQA – встановити чому допускаються помилки і як вони можуть бути виправлені. Об'єктом досліджень SQA є процеси ЖЦ ПС, а не програмні продукти.

- *Процеси верифікації та валідації* визначають, чи дійсно продукти певного етапу процесу розробки і супроводу ПС відповідають вимогам, які до них висуваються. Задача цих процесів – перевірити та підтвердити, що кінцевий програмний продукт буде відповідати своєму призначенню і задовольняти користувачів.

Методи які використовують як в цілях SQA так і в цілях V&V ділять на статичні та динамічні.

- *Статичні методи* - дослідження документації, інспекції, ревізії, аналіз потоків даних, аналіз дерев подій і дерев відмов, аналіз складності алгоритмів і т. д.
- До *динамічних методів* відносять тестування. Імітаційне моделювання та символічне виконання.

Керування ризиком в проекті.

Ризик виникає там, де є невизначеність, пов'язана з настанням якої-небудь небажаної події і є можливість потерпіти збитки внаслідок цієї події.

Ризик проекту ПС можна визначити як можливість зниження якості кінцевого продукту, перевищення вартості його розробки, затримки завершення розробки або зриву проекту із-за неефективності і недосконалості процесів ЖЦ ПС.

Величина ризику - це добуток серйозності наслідків небажаної події в проекті на ймовірність появи цієї події.

Ефективне керування ризиком полягає в прийнятті компромісних рішень по оцінюванню трудомісткості позбавлення від визначеного ризику з однієї сторони, і величини негативного впливу цього ризику на якість робочих продуктів і процесів – з іншої.

Підвищення зрілості організації.

Зрілість організації можна охарактеризувати як ступінь чіткості (ясності) визначення, управління, вимірювання, контролю і виконання процесу розробки ПС в організації.

Модель зрілості являє собою шаблон для оформлення маленьких еволюційних кроків для покращення процесу у вигляді 5-и рівнів зрілості.

- ◎ Рівень 1 – початковий. Включений в модель тільки з метою утворення точки підрахунку для оцінювання наступних покращень процесу . Характеризується тим, що процес розробки ПС неструктурований та хаотичний, а бюджет, графік і якість розробки непередбачувані.
- ◎ Рівень 2 – повторюваний. На цьому рівні керування проектом націлено на контроль дотримання планів по вартості, тривалості і функціональності розробки. Дисципліна розробки дозволяє застосовувати відпрацьовані засоби управління неодноразово у схожих проектах. Розробка нових проектів ведеться на основі накопиченого досвіду і у відповідності з основними стандартами в області програмування

- Рівень 3 – фіксований. Процес програмної інженерії в організації затверджений, стандартизований і документований. Впроваджена програма навчання штату розробників ПС і менеджерів. Колективи окремих проектів слідує базовому процесу розробки в організації і налаштовують його для досягнення цілей конкретного проекту.
- Рівень 4 – керований. Досягається мета кількісної оцінки якості програмних продуктів і процесу розробки в рамках єдиної програми вимірювань. Здійснюється збір і наліз даних по проектам, що дає можливість управляти ризиком проекту і при необхідності повертати процес у встановлені рамки.
- Рівень 5 – оптимізований. Забезпечується неперервне покращення процесу завдяки наявності засобів кількісної оцінки його слабких і сильних сторін. Дані про ефективність процесу розробки використовуються для проведення аналізу в цілях переходу на нові технології і вдосконалення процесу розробки в організації. Дані про нові засоби інженерії вивчаються і розповсюджуються по організації.