**Лекція 5. Етапи життєвого циклу програмного забезпечення**

**Опитування №1.**

1. **В чому сутність конструювання ПЗ?** (Конструювання ПЗ – створення ПЗ з конструкцій (блоків, операторів, функцій) і його перевірка методами верифікації і тестування.)
2. **Що таке модель конструювання?** (Моделі конструювання містять у собі набір операцій, послідовність дій і результатів. Види моделей визначаються стандартом ЖЦ, методологіями і практиками.)
3. **Які з видів діяльності, що проходять в процесі розробки ПЗ, не відносяться до конструювання?** (керування процесом розробки, виробки вимог, розробка високорівневої архітектури програми, проектування інтерфейсу користувача, тестування системи і її супроводження).
4. **Які концепції відносяться до фундаментальних** **основ конструювання** **ПЗ?** (Мінімізація складності, Очікування змін, Конструювання з можливістю перевірки, Стандарти у конструюванні)
5. **Визначте стратегії конструювання ПЗ. (**Однократний прохід, Інкрементна (заплановано покращення продукту), Еволюційна**)**
6. **В чому полягає стратегія однократного проходу? (**(водоспадна стратегія) — лінійна послідовність етапів конструювання**)**
7. **В чому полягає інкрементна стратегія? (**На початку процесу визначаються всі користувацькі та системні вимоги, решта конструювання виконується у вигляді послідовності версій. Перша версія реалізує частину запланованих можливостей, наступна версія реалізує додаткові можливості й т. д., поки не буде отримано повну систему**).**
8. **В чому полягає еволюційна стратегія? (**ПС будується у вигляді послідовності версій, але на початку процесу визначені не всі вимоги. Вони уточнюються в результаті розробки версій)
9. **Що таке модель ПЗ**?(формалізований опис системи ПЗ на певному рівні абстракції).
10. **Як можна класифікувати вимоги до ПЗ (**вимоги користувача, функціональні вимоги, програмні вимоги**)**
11. **Що таке специфікація**? (документ, що в закінченій, точній і перевіреній формі описує вимоги, проект, поведінку або інші характеристики компонента або системи, а також процедури, спрямовані на визначення того, чи задовольняються описані характеристики).
12. **В чому полягає процес декомпозиції?** (Це розбиття складного об'єкта на порівняно незалежні частини з визначенням всіх видів зв'язків частин між собою)
13. **В чому полягає метод розробки з назвою** «**покрокова деталізація»? (**багаторазове виконання процесу декомпозиції виконується: кожен блок, у свою чергу, декомпозується на частини, поки не отримують блоки, які порівняно легко розробити).
14. **Які цілі процесу декомпозиції?** (намагаються виділити аналогічні блоки, які можна було б розробляти на загальній основі, щоб забезпечити збільшення ступеня повторюваності кодів і, відповідно, зниження вартості розробки).
15. **Як зазвичай представляють результат декомпозиції?** (у вигляді схеми ієрархії, на нижньому рівні якої розміщають відносно прості блоки, а на верхньому - об'єкт, що підлягає розробці).
16. **Для чого використовується** **діаграма Ґанта і що вона собою являє? (**стрічкова діаграма, яка використовується для ілюстрації плану, графіка робіт за будь-яким проєктом. Діаграма Ґанта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі.)
17. **В чому сутність планування проекту?** (це процес розподілу й призначення ресурсів (матеріальних і людських) з урахуванням вартості й часу виконання проекту)
18. **Що таке життєвий цикл програмного забезпечення?** (ЖЦ ПЗ — сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення програмного забезпечення і закінчується припиненням використання програмного забезпечення.)
19. **Які різновиди моделей ЖЦ ПЗ Вам відомі?** (каскадна, еволюційна, ітеративна та інкрементна модель, спіральна, V-подібної модель, адаптивні моделі).
20. **В чому сутність каскадної моделі?** (послідовне виконання етапів створення ПЗ.)
21. **Назвіть основні характеристики і фази каскадної моделі** (дослідження концепції, процеси системного розподілу, визначення вимог, розробки проекту, реалізації, установки, експлуатації та підтримки, супроводження, виведення з експлуатації, інтегральні завдання).
22. **Назвіть основні переваги каскадної моделі?** (отримання в кінці кожної стадії закінченого набору проектної документації, що відповідає вимогам повноти і узгодженості; - простота планування процесу розробки)
23. **Назвіть основні недоліки каскадної моделі?** (негнучкість; фаза повинна бути завершена до переходу до наступної; набір фаз фіксований; важко реагувати на зміни вимог)
24. **Де краще використовувати каскадну модель** (там, де вимоги добре зрозумілі і стабільні).
25. **Які фази ЖЦ ПЗ безпосередньо стосуються розроблення ПЗ**? (проектування, реалізація, тестування, інколи-супровід)
26. **Які моделі життєвого циклу дозволяють оперативно реагувати на зміну вимог до ПЗ?** (спіральна, адаптивна)
27. **В чому сутність ітераційної спіральної модель ЖЦ ПЗ?** (розробка ПЗ здійснюється по спіралі, кожен виток (ітерація) якої передбачає реалізацію певного функціоналу програмної системи)
28. **Які характеристики спіральної моделі життєвого циклу є її перевагою?** Поясніть.( починаючи з деякої ітерації, на якій забезпечена певна функціональна повнота, продукт можна надавати користувачеві)
29. **В** **чому сутність інкрементної ітераційної модель ЖЦПЗ** (розробка ПЗ реалізується декількома ітераціями з поступовим нарощуванням функціональності системи)
30. **В** **чому сутність V-подібної моделі? (**В цій моделі тестування розглядається як неперервний процес, інтегрований в процес розробки ПС. Він включає два взаємопов’язаних підпроцеса: планування тестування в рамках процесів розробки системи (ліва гілка) і проведення тестування відповідних об’єктів (права гілка)**)**
31. **Дайте визначення прототипу програмного продукту** (Прототипом називають діючий програмний продукт, який реалізує окремі функції і зовнішні інтерфейси розроблюваного програмного забезпечення)
32. **Визначте цілі прототипування (**перевірка концепції та моделювання процесів, керування інвестиціями та мінімізація ризиків**)**
33. **Де прототипування дає найбільші переваги?** (при розробленні систем, які мають розвинутий інтерфейс користувача)
34. **В** **чому сутність CASE технології**? (CASE (англ. Computer-Aided Software Engineering) — набір інструментів і методів програмної інженерії для проектування ПЗ, що допомагає забезпечити високу якість програм, відсутність помилок і простоту в обслуговуванні програмних продуктів.)
35. **Які адаптивні моделі Вам відомі**? (Scrum, XP (Extreme Programming), Adaptive Software development (ASD), Dynamic System Development Model (DSDM), Feature Driven Development (FDD) )

**Етапи життєвого циклу програмного забезпечення**

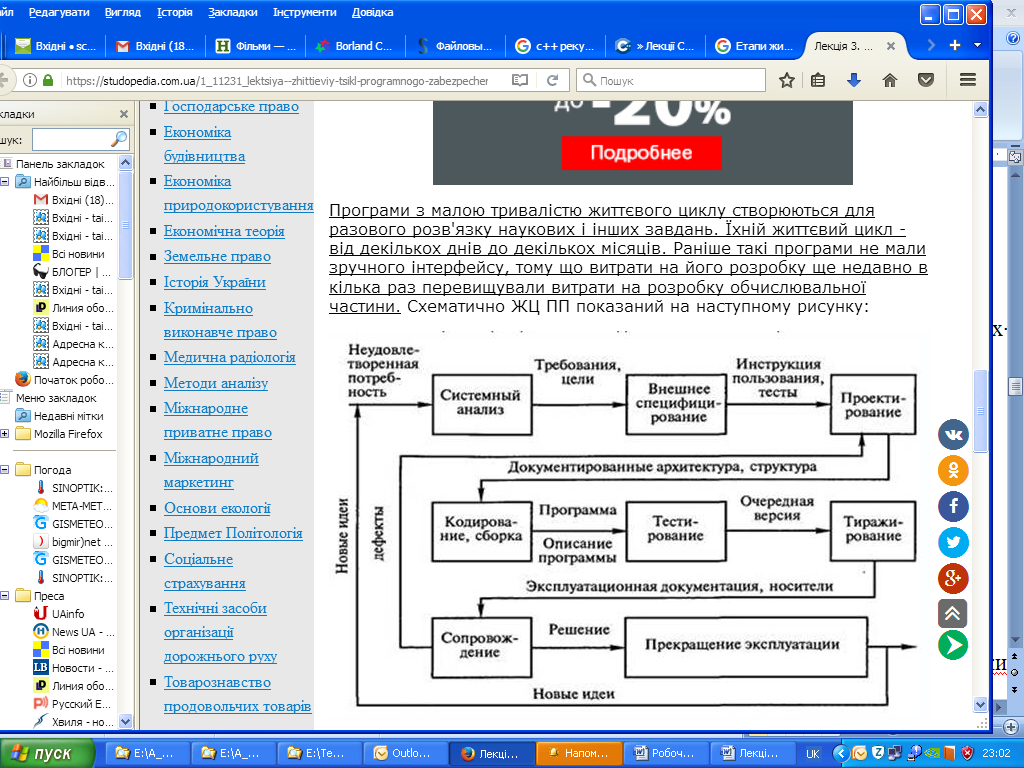
Поняття життєвого циклу займає центральне місце в технології програмування. Потреба в цім понятті виникла у зв'язку з перетворенням процесу розробки програм на інженерну дисципліну. Уперше про життєвий цикл заговорили в 1968 році в Лондоні, де відбулася зустріч 22- х керівників проектів по розробці програмного забезпечення. На зустрічі аналізувалися проблеми й перспективи проектування, розробки, поширення й підтримки програм. Було констатовано, що методи, що застосовуються у розробці ПЗ вимагають постійного вдосконалення. Саме на цій зустрічі була запропонована концепція життєвого циклу ПЗ (SLC - Software Lіfetіme Cycle) як послідовності кроків- стадій, які необхідно виконати в процесі створення й експлуатації програмного засобу.

**Програми** створюються, експлуатуються й розбудовуються в часі.

**Життєвий цикл ПЗ** - це період часу, який починається з моменту ухвалення рішення про необхідність створення програмного забезпечення й закінчується в момент його повного вилучення з експлуатації (ІEEE Std. 610.12 - 19990 Standard Glossary of Software Engіneerіng Termіnology).

Це поняття не є специфічним для програмування. Воно виникло й розвивалося спочатку стосовно до технічних систем. Зокрема, ще кілька десятиліть назад наші економісти виражали своє занепокоєння із приводу того, що закордонний споживач порівняно дешевим радянським тракторам віддає перевагу канадським, ціна яких у кілька разів вище. Виявилося, що повна вартість останніх з урахуванням витрат усього періоду існування машин (включаючи їх технічне обслуговування й ремонт) виходить в кілька раз менше. Не випадково питання технологічності не тільки з погляду виготовлення, але й наступної експлуатації, має в техніці першорядне значення.

Програми з малою тривалістю життєвого циклу створюються для разового розв'язку наукових і інших завдань. Їхній життєвий цикл - від декількох днів до декількох місяців. Раніше такі програми не мали зручного інтерфейсу, тому що витрати на його розробку ще недавно в кілька раз перевищували витрати на розробку обчислювальної частини. Схематично ЖЦ ПП показаний на наступному рисунку:



Кожна програма починається з якої-небудь незадовільненої потреби й, усвідомивши її, необхідно провести **системний аналіз** для виявлення цілей майбутнього програмного виробу й вимог до нього. Наступним етапом буде зовнішнє **специфікування**, призначене для створення "ідеології" програми - загальної спрямованості в наступному проектуванні, аж до зовнішнього вигляду програми й інструкції користування програмою. На етапі **проектування** програмний виріб специфікується в повному обсязі від постановки завдання до робочого проекту з описом внутрішньої структури програми й плану розробки частин програми. Потім відбувається **кодування й тестування**, у результаті чого виходить готова версія програми. Програма випускається в тираж і супроводжується виробником. **Супровід** полягає як в усуненні помилок, що виявляються в процесі експлуатації, і випуску виправлених версій, так і в удосконаленні базової версії програми, що найчастіше призводить до перепроектування програми й випуску радикально оновлених версій. Закінчення життєвого циклу – це **припинення експлуатації** розробки. Однак ідеї, висунуті в процесі експлуатації програми, звичайно використовуються при розробці наступного, більш досконалого й сучасного виробу.

Припинення експлуатації - звичайно не одномоментний акт знищення програми в комп'ютері, а період часу, коли деякі організації або деякі користувачі ще продовжують використовувати стару розробку.

**Основні етапи розробки програмного забезпечення:**

- Постановка задачі (стадія «Технічне завдання»);

- Аналіз вимог і розробка специфікацій (стадія «Ескізний проект»);

- Проектування (стадія «Технічний проект»);

- Реалізація (стадія «Робочий проект»).

Традиційно розробка також включала етап супроводу (початку цього етапу відповідає стадія «Впровадження» по ГОСТ). Однак за міжнародним стандартом відповідно до змін, що відбулися в індустрії розробки програмного забезпечення, цей процес тепер розглядається окремо.

Умовність виділення етапів пов'язана з тим, що на будь-якому етапі можливе прийняття рішень, які зажадають перегляду рішень, прийнятих раніше .

**Традиційний погляд.**

**Постановка задачі.** У процесі постановки завдання чітко формулюють призначення програмного забезпечення і визначають основні вимоги до нього. Кожна вимога є опис необхідного або бажаного властивості програмного забезпечення. Розрізняють функціональні вимоги, що визначають функції, які має виконувати розробляється програмне забезпечення, і експлуатаційні вимоги, що визначають особливості його функціонування.

**Аналіз вимог і визначення специфікацій.** Специфікаціями називають точну формалізоване опис функцій і обмежень розроблюваного програмного забезпечення. Відповідно розрізняють функціональні і експлуатаційні специфікації. Сукупність специфікацій є загальну логічну модель проектованого програмного забезпечення.

На цьому етапі також доцільно сформувати тести для пошуку помилок в проектованому програмному забезпеченні, обов'язково вказавши очікувані результати.

**Проектування.** Основним завданням цього етапу є визначення докладних специфікацій розроблюваного програмного забезпечення. Процес проектування складного програмного забезпечення зазвичай включає:

- Проектування загальної структури - визначення основних компонентів і їх взаємозв'язків;

- Декомпозицію компонентів і побудова структурних ієрархій відповідно до рекомендацій блочно-ієрархічного підходу;

- Проектування компонентів.

Результатом проектування є детальна модель розроблюваного програмного забезпечення разом зі специфікаціями його компонентів всіх рівнів. Тип моделі залежить від обраного підходу (структурний, об'єктний або компонентний) і конкретної технології проектування. Процес проектування охоплює як проектування програм (підпрограм) і визначення взаємозв'язків між ними, так і проектування даних, з якими взаємодіють ці програми або підпрограми. Прийнято розрізняти також два аспекти проектування:

- Логічне проектування, яке включає ті проектні операції, які безпосередньо не залежать від наявних технічних і програмних засобів, що становлять середовище функціонування майбутнього програмного продукту;

- Фізичне проектування - прив'язка до конкретних технічних і програмних засобів середовища функціонування, тобто облік обмежень, визначених у специфікаціях.

**Реалізація.** Реалізація являє собою процес поетапного написання кодів програми на обраною мовою програмування (кодування), їх тестування і налагодження.

**Супровід.** Супровід - це процес створення і впровадження нових версій програмного продукту. Причинами випуску нових версій можуть служити:

- Необхідність виправлення помилок, виявлених в процесі експлуатації попередніх версій;

- Необхідність вдосконалення попередніх версій, наприклад, поліпшення інтерфейсу, розширення складу виконуваних функцій або підвищення його продуктивності;

- Зміна середовища функціонування, наприклад, поява нових технічних засобів та / або програмних продуктів, з якими взаємодіє супроводжуване програмне забезпечення.

На цьому етапі в програмний продукт вносять необхідні зміни, які так само, як в інших випадках, можуть вимагати перегляду проектних рішень, прийнятих на будь-якому попередньому етапі. Зі зміною моделі життєвого циклу програмного забезпечення роль цього етапу істотно зросла, так як продукти тепер створюються ітераційно: спочатку випускається порівняно проста версія, потім наступна з великими можливостями, потім наступна і т. Д. Саме це і послужило причиною виділення етапу супроводу в окремий процес життєвого циклу відповідно до стандартом 1SO / IEC 12207.

**Модель ЖЦ ПЗ у вигляді базових етапів - процесів**

Необхідність визначення етапів життєвого циклу (ЖЦ) ПЗ обумовлена прагненням розробників до підвищення якості ПЗ за рахунок оптимального управління розробкою і використання різноманітних механізмів контролю якості на кожному етапі, починаючи від постановки завдання і закінчуючи авторським супроводом ПЗ. Найбільш загальним представленням життєвого циклу ПЗ є модель у вигляді базових етапів - процесів, до яких відносяться:

* системний аналіз і обгрунтування вимог до ПЗ;
* попереднє (ескізне) і детальне (технічне) проектування ПЗ;
* розробка програмних компонент, їх комплексування і відлагодження ПЗ в цілому;
* випробовування, дослідна експлуатація та тиражування ПЗ;
* регулярна експлуатація ПЗ, підтримка експлуатації та аналіз результатів;
* супровід ПЗ, його модифікація і вдосконалення, створення нових версій.

Дана модель є загальноприйнятою і відповідає як вітчизняним нормативним документам у області розробки програмного забезпечення, так і зарубіжним.

Спочатку була створена каскадна модель ЖЦ, в якій крупні етапи починалися один за одним з використанням результатів попередніх робіт. Найбільш специфічною є спіралевидна модель ЖЦ. У цій моделі увага концентрується на ітераційному процесі початкових етапів проектування. На цих етапах послідовно створюються концепції, специфікації вимог, попередній і детальний проект. На кожному витку уточнюється зміст робіт і концентрується зовнішність створюваного ПЗ.

Стандартизація ЖЦ ПЗ проводиться по трьох напрямах. Перший напрям організовується і стимулюється Міжнародною організацією зі стандартизації (ISO - International Standard Organization) і Міжнародною комісією з електротехніки (IEC - International Electro-technical Commission). На цьому рівні здійснюється стандартизація найбільш загальних технологічних процесів, що мають значення для міжнародної кооперації. Другий напрям активно розвивається в США Інститутом інженерів електротехніки і радіоелектроніки (IEEE - Institute of Electrotechnical and Electronics Engineers) спільно з Американським національним інститутом стандартизації (American National Standards Institute-ANSI). Стандарти ISO/IEC і ANSI/IEEE в основному мають рекомендаційний характер. Третій напрям стимулюється Міністерством оборони США (Department of Defense-DOD). Стандарти DOD мають обов'язковий характер для фірм, що працюють за замовленням Міністерства оборони США.

Для проектування ПЗ складної системи, особливо системи реального часу, доцільно використовувати загальносистемну модель ЖЦ, засновану на об'єднанні всіх відомих робіт в рамках розглянутих базових процесів. Ця модель призначена для використання при плануванні, складанні робочих графіків, управлінні різними програмними проектами.

Сукупність етапів даної моделі ЖЦ доцільно ділити на дві частини, що істотно розрізняються особливостями процесів, техніко-економічними характеристиками і впливаючими на них чинниками.

У першій частині ЖЦ проводиться системний аналіз, проектування, розробка, тестування і випробовування ПЗ. Номенклатура робіт, їх трудомісткість, тривалість та інші характеристики на цих етапах істотно залежать від об'єкту і середовища розробки. Вивчення подібних залежностей для різних класів ПЗ дозволяє прогнозувати склад і основні характеристики графіків робіт для нових версій ПЗ.

Друга частина ЖЦ, що відображає підтримку експлуатації і супроводу ПЗ, відносно слабо пов'язана з характеристиками об'єкту і середовища розробки. Номенклатура робіт на цих етапах стабільніша, а їх трудомісткість і тривалість можуть істотно змінюватися, і залежать від масовості застосування ПЗ. Для будь-якої моделі ЖЦ забезпечення високої якості програмних комплексів можливо лише при використанні регламентованого технологічного процесу на кожному з цих етапів. Такий процес підтримується CASE засобами автоматизації розробки, які доцільно вибирати з тих, що є або створювати з урахуванням об'єкту розробки і адекватного йому переліку робіт.

**Технологічний цикл конструювання ПЗ**

Відомо, що технологічний цикл конструювання ПЗ містить три процеси:

1) аналіз;

2) синтез;

3) супроводження.

Протягом аналізу виявляються цілі створюваної системи, тобто вказуються функції та зміни стану системи залежно від оточення та керованих параметрів.

У процесі синтезу вказуються способи реалізації запропонованих на першому етапі функцій системи, тобто виконується програмна реалізація системи. Виокремлюють три етапи синтезу:

1) проектування ПЗ;

2) кодування ПЗ;

3) тестування ПЗ



**Етап проектування** доповнює вимоги до ПЗ, які подані моделями аналізу: *інформаційною*, *функціональною моделями* та *моделлю поведінки*. ***Інформаційна модель*** описує інформацію, яку, за задумом, має обробляти ПЗ. ***Функціональна модель*** визначає перелік функцій оброблення. ***Модель поведінки*** фіксує бажану динаміку системи (режими її роботи). На виході з етапу проектування виконується *розроблення даних, архітектури* та *процедурне розроблення ПЗ.*

***Розроблення даних*** – це результат перетворення інформаційної моделі аналізу в структури даних для реалізації програмної системи.

***Розроблення архітектури*** – виокремлення основних структурних компонентів та фіксація зв’язків між ними.

***Процедурне розроблення*** – опис послідовності дій у структурних компонентах, тобто визначення їх умісту.

Далі створюють тести програмних модулів, виконують ***тестування*** та ***перевірку системи програмування***. На проектування, кодування і тестування відводиться 75 % вартості конструювання системи програмування.

Рішення, які приймаються протягом проектування, роблять цей процес провідним етапом процесу синтезу. Важливість проектування визначається ще і якістю. ***Проектування*** – етап, на якому «вирощують» якість розроблення системи програмування, це єдиний шлях, який забезпечує правильну трансляцію вимог замовника в кінцевий програмний продукт.

**Розроблення програмних модулів**

Методології розроблення ПЗ корисні для розроблення великих складних продуктів або розподілених інформаційних систем.

Розробляючи відносно невеликі програми або реалізуючи конкретний програмний модуль, достатньо притримуватися такої послідовності кроків:

1. Постановка завдання.

2. Обґрунтований вибір засобів розроблення (програмування).

3. Вибір методу розв’язання завдання.

4. Розроблення алгоритму рішення завдання.

5. Кодування засобами обраної мови програмування.

6. Верифікація й перевірка коректності.

7. Тестування програми.

8. Налагодження програми у випадку виявлення помилок.

9. Розроблення документації.

10. Експериментальна експлуатація.

11. Промислова експлуатація.

**ЖЦ ПЗ за ISO 12207**

Базовим стандартом розроблення ПЗ є ISO 12207. Systems and software engineering – Software Life Cycle Processes, в якому усі процеси ЖЦ ПЗ розподілені на три групи (рис. 3.5).



Рисунок 3.5 – Процеси ЖЦ ІС відповідно до стандарту ISO 12207

***Допоміжні процеси*** призначені для підтримки виконання основних процесів, забезпечення якості проекту, організації верифікації та тестування ПЗ. ***Організаційні процеси*** визначають дії та завдання замовників та розробників для керування процесами у ході проекту.

Для підтримки практичного використання стандарту ISO 12207 розроблені такі технологічні документи: *Керівництво для ISO/IEC 12207* (ISO/IEC TR 24748-3:2011 Systems and software engineering - Life cycle management - Part 3: Guide to the application of ISO/IEC 12207 (Software life cycle processes)) та *Керівництво з використання ISO/IEC 12207 в керуванні проектами* (ISO/IEC TR 16326:2009 Systems and software engineering - Life cycle processes - Project management).

**ЖЦ ПЗ за ISO/IEC 15288**

У 2002 р. був опублікований стандарт на процеси життєвого циклу систем ISO/IEC 15288 Systems and software engineering - System life cycle processes, у розробленні якого брали участь фахівці різних галузей: системної інженерії, програмування, управління якістю, людськими ресурсами, безпекою та ін. Даний документ враховує практичний досвід створення систем в урядових, комерційних, військових та академічних організаціях і може бути застосований для широкого класу систем, але його основне призначення – підтримка створення комп'ютеризованих систем. На цей час діє версія стандарту 2008 р. У стандарті ISO/IEC 15288:2008 у структурі ЖЦ виділені групи процесів за видами діяльності (рис. 3.6).



|  |  |  |
| --- | --- | --- |
| Таблиця. Стадії створення систем (ISO/IEC 15288) | | |
| № п/п | Стадія | Опис |
| 1 | ***Формування концепції*** | Аналіз потреб, вибір концепції та проектних рішень |
| 2 | ***Розробка*** | проектування системи |
| 3 | ***Реалізація*** | виготовлення системи |
| 4 | **Експлуатація** | Введення в експлуатацію і використання системи |
| 5 | **Підтримка** | Забезпечення функціонування системи |
| 6 | **Зняття з експлуатації** | Припинення використання, демонтаж, архівування системи |

**Проектування як стратегічний етап**

*Мета стратегічного етапу* полягає в тому, щоб описати проект з точки зору клієнта. Всі цілі можуть здаватися іноді очевидними. Проте, спроба визначити їх зазвичай дозволяє уникати можливого непорозуміння між клієнтом і виробником. Крім того, навіть якщо цілі ясні для всіх людей, залучених до стратегічного етапу, вони не будуть очевидні для людей, які працюють на подальших стадіях і людям, які не мають прямого контакту з клієнтом. *Метою проекту повинен бути докладний опис вимог на випадок, якщо виникають будь-які сумніви в подальших етапах.*

Наступне завдання, яке виконується на стратегічному етапі полягає в тому, щоб *описати можливості проекту*. Зазвичай розвинена система підтримуватиме певні аспекти діяльності клієнта. Відповідний опис можливостей необхідний для хорошої вартісної оцінки проекту. Проте, клієнт не знає можливостей проекту на даному етапі. Може бути незрозумілим, які функції будуть здійснені в ПЗ і які функції будуть здійснені користувачами, іншими системами і апаратними засобами.

Визначення можливостей проекту призводить до контекстного опису, тобто до опису зовнішніх систем, з якими взаємодіяла б система. У багатьох випадках система використовується одним або декількома користувачами. Цих користувачів розглядають як зовнішню систему.

Визначення мети, можливостей і контексту проекту недостатньо для вартісної оцінки виробництва і складних систем. Точна оцінка можлива тільки якщо всі етапи, такі як формулювання вимог, аналіз і проектування, були виконані. Практично фінансові ресурси доступні в стратегічному етапі, обмежені, тому рекомендується робити тільки загальну оцінку. Важливо враховувати всі можливості описаної системи.

Стратегічний етап називають також етапом проектування. Він виконується перш, ніж ухвалені остаточні рішення. У випадку компанії, яка виробляє ПЗ для клієнта, це етап ведення переговорів про продукт з клієнтом. У випадку, коли компанія розробляє ПЗ для ринку, це - етап обговорення і планування продукту.

Стратегічний етап також називають техніко-економічним вивченням. На цьому етапі виконуються наступні дії:

· обговорення проекту з представниками клієнта

· визначення мети проекту з точки зору клієнта

· визначення можливостей і контексту проекту

· приблизне формулювання вимог, аналіз і проект системи

· формулювання альтернативних рішень

· аналіз

· представлення результатів представникам клієнта і здійснення виправлень

· попереднє планування і вибір структури команди

· стандартні визначення

У випадку розробки системи ПЗ для клієнта ми повинні розрізняти людину-замовника і людей, які використовуватимуть і застосовуватимуть систему. В загальному наш проект повинен відповідати вимогам замовників, так як продукт буде оцінюватися ними. Дуже часто вони не є користувачами системи.

**На цьому етапі існує декілька стратегічних рішень, які повинні бути прийняті:**

· вибір моделі проекту

· вибір методів, що будуть використовуватися для аналізу і проектування

· вибір програмного середовища

· вибір CASE-інструментів

· рішення про використання наборів інструментів

· рішення про можливу співпрацю

**Як правило, існують декілька можливих рішень по системі і ці варіанти рішень підпорядковані певним обмеженням. Обмеження можуть стосуватись:**

· максимальна допустима вартість

· доступні професіонали і персонал

· доступні інструменти

· обмеження в часі

На стратегічному етапі виконується попереднє планування ( визначаються підзадачі і оцінюється час, необхідний для їх виконання). Тому деталі проекту ще не визначені і графік роботи є дуже загальним.

**На стратегічному етапі повинні бути визначені стандарти. Вони включають:**

· використання інструментів і понять

· методи документування

Оцінка рішення може бути заснована на таких критеріях, як вартість, затрати часу на проект, надійність, повторне використання компонентів системи, мобільності і спосіб виконання.

**До ключових чинників успіху на стратегічному етапі належить:**

· Ефективність роботи ( особливо для компаній, що працюють на конкретного клієнта на ринку конкуруючих компаній, будь-які затримки можуть призвести до низьких шансів на нові замовлення).

· Незалежно від системи, що розробляється, компанія не може витратити значні кошти на стратегічному етапі ( таким чином, етап повинен бути організований невеликою групою людей і в короткі терміни).

· Нерозуміння ключових моментів клієнтом ( цей чинник робить успіх проекту неможливим).

· Осмислення всієї системи ( більшість невдач, викликані в результаті приділення дуже великої уваги деяким фрагментам системи, тобто відсутнє розуміння частин, які можуть бути найважчими в розробці).

**До основних результатів стратегічного етапу відносять:**

· складання звіту, який охоплює

визначення мети

опис можливостей

опис зовнішніх систем

опис загальних вимог

загальна модель системи

запропоноване рішення по системі

вартісна оцінка

попереднє планування

· звіт про відносну оцінку рішення, що містить інформацію про всі можливі рішення і обгрунтування ухвалених рішень

· представлення необхідних ресурсів - штату, апаратних засобів, програмного забезпечення і т.д.

· стандартні визначення

· планування аналізу