**Лекція 7. Процес розробки програмного забезпечення**

**На період** карантину в дистанційній формі навчання на надані в кінці запитання потрібно надати письмові відповіді, надіславши їх на електронну адресу викладача. Файл надавати з іменем у форматі

**ОPI<Номер групи><Номер лекції / практичної / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**. Наприклад, О**PI3104L**buts.doc. Кожна відповідь оцінюється в 0,5 балів. Відповіді повинні бути лаконічними і змістовними. Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності відповідей -"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк виконання цієї роботи ІПЗ-31,32,33 20.11.2024**

**Процес створення ПЗ**

Центральним об'єктом вивчення *програмної інженерії* є ***процес створення ПЗ*** – певна сукупність різних видів діяльності, методів, методик і кроків, що використовуються для розробки і *еволюції ПЗ* і пов'язаних з ним продуктів (проектних планів, документації, програмного коду, тестів, призначеної для користувача документації та ін.). Проте, на сьогоднішній день не існує **універсального процесу** розробки ПЗ як набору методик, правил і розпоряджень, які б відповідали б ПЗ будь-якого виду, для будь-яких компаній, для будь-якої національної команди. Кожен **поточний процес** розробки, який здійснюється деякою командою в межах *певного проекту*, має велику кількість особливостей і індивідуальностей. Це відкидає доцільності планування процесу роботи перед початком проекту із визначенням ролей і обов'язків в команді, *робочих продуктів* (проміжних і фінальних), порядку участі в їх розробці членів команди і так далі. Називатимемо цей попередній опис **конкретним процесом**, відрізняючи його від плану *робіт*, проектних специфікацій і ін.

В межах окремої компанії можлива і корисна стандартизація всіх поточних процесів, яку називатимемо **стандартним процесом**. Останній, таким чином, виявляється деякою базою даних, що містить такі дані:

* інформацію, правила використання, документацію і інсталяційні пакети *засобів розробки*, які використовуються в проектах компанії (систем версійного контролю, засобів контролю помилок, засобів програмування – різних *IDE*, *СУБД* і т.д.);
* опис практик розробки – проектного *менеджменту*, правил роботи із замовником і т.д.;
* шаблони проектних документів – *технічних завдань*, проектних специфікацій, планів тестування тощо.

Також можлива стандартизація процедури розробки конкретного процесу як частки стандартного. Основна ідея стандартного процесу – впровадження усередині компанії передового досвіду, а також *уніфікація засобів розробки*. Це особливо важливо, коли в компаніях різні департаменти і проекти сильно відрізняються *за зрілістю процесу розробки*, що ускладнює повторне використання передового досвіду, а також коли компанія використовує декілька засобів паралельних інструментів розробки. Такі підходи істотно ускладнюють міграцію фахівців з проекту в проект, а також використання результатів одного проекту в іншому. Проте, у разі організації стандартного процесу необхідно стежити, щоб стандартний процес не виявився лише формальним, бюрократичним інструментом.

**Вдосконалення процесу створення ПЗ**

Вдосконалення процесу (*software process improvement*) – це *діяльність щодо* змін існуючого процесу (як поточного, в межах одного проекту, так і стандартного, для всієї компанії) з метою поліпшення якості створюваних продуктів і/або зниження ціни і часу їх розробки. Причини актуальності цієї діяльності для компаній-виробників полягає в наступному.

1. Відбувається швидка зміна технологій розробки ПЗ, потрібні вивчення і впровадження нових засобів розробки.

2. Спостерігається швидке зростання компаній і їх вихід на нові ринки, що вимагає нової організації робіт.

3. Має місце висока конкуренція, яка вимагає пошуку ефективніших, більш економічних способів розробки.

Предмети і методи удосконалення (що і яким чином можна покращувати) можна в узагальненому вигляді визначити таким чином.

1. Перехід на нові засоби розробки, *мови програмування* та інші інноваційні заходи.

2. *Поліпшення* окремих управлінських і інженерних практик, зокрема, тестування, *управління вимогами*, конфігурацією.

3. Повна, комплексна перебудова всіх процесів в проекті, департаменті, компанії (у відповідності, наприклад, до вимог *CMM* – див. лекцію №2).

4. Сертифікація компанії (за стандартами *CMM*, *ISO 9000* та ін.).

Потрібно зауважити, що діяльність із сертифікація компанії не завжди зв’язана *з* поліпшення процесів розробки *ПЗ,* оскільки може зводитися до підтримки відповідного документообігу, який є необхідним для отримання сертифікації. Сертифікат потім використовується як засіб, козир в боротьбі за замовлення.

Головна важкість реального вдосконалення процесів в компанії полягає в тому, що вона при проведенні вдосконалення повинна працювати і створювати ПЗ. Саме це є причиною появи ідеї безперервного поліпшення процесу, так би мовити, малими порціями, щоб процедура була не такою хворобливою. Така стратегія відповідає стандарту вдосконалення процесів розробки *CMM*.

**Стратегії впровадження інновацій**

**Pull/push стратегії**. У контексті впровадження *інновацій* у виробничі процеси бізнес-компаній (не обов'язково компанії для створення *ПЗ*) існують дві *парадигми*.

1. *Organization pull – інновації*, які націлені на вирішення конкретних проблем компанії.

2. *Technology push –* широкомасштабне впровадження *інновацій* із стратегічних міркувань. Замість конкретних проблем, які будуть вирішені після впровадження *інновації*, в цьому випадку розглядаються показники компанії (ефективність, *продуктивність*, річний оборот засобів, збільшення *вартості акцій* публічної компанії), які будуть збільшені, покращені після впровадження *інновації*. При цьому передбачається, що будуть автоматично вирішені численні окремі проблеми організації, у тому числі і ті, про які в даний момент нічого не відомо.

***Приклад використання*** ***стратегії*** *organization pull –* впровадження нових засобів тестування за ситуації, коли є високі вимоги *до якості* в проекті, або коли якість програмної системи не задовольняє замовника. Приклад використання стратегії *technology push –* перехід компанії із засобів структурної розробки на об’єктно-орієнтоване програмування. Ще один приклад використання тієї ж стратегії – впровадження стандартів якості ISO *9000* або CMMI. У обох цих випадках компанія не вирішує якусь одну проблему або ряд проблем – вона хоче радикально змінити ситуацію, вийти на нові позиції.

***Проблеми застосування стратегії*** *technology push* полягають в тому, що потрібна глобальна перебудова процесу. Але компанію не можна "закрити на реконструкцію" – за цей час місце компанії на ринку може бути зайняте конкурентами, акції компанії можуть впасти і так далі. Таким чином, впровадження *інновацій*, як правило, відбувається паралельно із звичайною діяльністю компанії, поетапно, що у випадку з *technology push* зв'язано з великими труднощами і ризиками. Використання стратегії *organization pull* менш ризиковане, зміни, що вносяться нею в процес, менш глобальні, локальні. Але і переваг такі *інновації* приносять менше, *в порівнянні з* вдалими впровадженням відповідно до стратегії *technologypush*.

Необхідно також відзначити, що існують проблеми, які неможливо усунути точковими переробками процесу, тобто необхідно застосовувати стратегію *technology push*. Наприклад, процес супроводу і розвитку сімейства *програмних* зайшов у *безвихідь, –* компанія стає збитковою, якщо і далі буде супроводжувати встановлені раніш продукти, інструментальні засоби проекту безнадійно застаріли і знаходяться в жалюгідному стані, *менеджмент* засмучений, всі спроби керівництва змінити процес натрапляють на нерозуміння колективу, сварки і *конфлікти*. Можливо, що у такому разі без "революції" не обійтися.

Ще одна відмінність обох стратегій: у випадку з *organization pull*, як правило, повернення інвестицій від впровадження відбувається швидше, ніж у випадку з *technology push*.

**Визначення моделі процесу**.

Процес створення *програмного забезпечення* не є однорідним. Той або інший метод розробки *ПЗ,* як правило, визначає деяку динаміку *розгортання* тих або інших видів діяльності, тобто, визначає модель *процесу* (*process model*).

Модель є *абстракцією* різних методів розробки *ПЗ,* яка дозволяє лаконічно, стисло і інформативно їх унаочнити. Сама ідея *моделі процесу* є однією з найбільш ранніх в *програмній інженерії*, коли вважалося, що вдала модель – це найголовніше, що сприяє успіху розробки. Пізніше прийшло усвідомлення, що існує безліч інших аспектів (принципи управління і розробки, структура команди і так далі), які повинні бути узгоджені один з одним. В результаті були започатковані інтегральні *методології розробки*. Класичні моделі процесу ми розглядали в лекції №2.

**Фази і види діяльності**. Кажучи про моделі процесів, необхідно розрізняти фази і види діяльності.

**Фаза** (*phase*) – це певний етап процесу, який має початок, кінець і вихідний результат. Наприклад, фаза перевірки здійсненності проекту, здача проекту і так далі. Фази слідують одна за одною в лінійному порядку, характеризуються наданням звітності замовникові і, часто, виплатою грошей за виконану частину роботи. Поодинокий замовник погодиться перший раз побачити результати лише після завершення проекту. З іншого боку, підрядчики вважають за краще отримувати гроші поступово, *у міру того, як* виконуються окремі частини роботи. Таким чином, з'являються фази, що дозволяють створювати і пред'являти проміжні результати проекту. Фази корисні також безвідносно взаємодії із замовником – з їх допомогою можна синхронізувати *діяльність* різних *робочих груп*, а також відслідковувати перебіг проекту. Прикладами фаз може служити узгодження із замовником технічного завдання, реалізація певної функціональності *ПЗ,* етап розробки, що закінчується здачею системи на тестування або випуском альфа-версії.

**Вид діяльності** (*activity*) – це певний тип роботи, що виконується в процесі розробки *ПЗ.* Різні види діяльності часто вимагають різних професійних навичок і виконуються різними фахівцями. Наприклад, *управління проектом виконується менеджером проекту*, *кодування – програмістом, тестування – тестувальником*. Є види діяльності, які можуть виконуватися одними і тими самими фахівцями, – наприклад, *кодування* і *проектування* (особливо в невеликому проекті) часто виконують одні і ті ж самі люди.

В межах однієї фази може виконуватися багато різних видів діяльності. Крім того, один вид діяльності може виконуватися на різних фазах наприклад, тестування: на фазі аналізу і проектування можна писати тести і налагоджувати тестове оточення, а саме тестування проводити під час розробки і перед здачею. На поточний момент для складного *програмного забезпечення* використовуються *багатовимірні моделі* процесу, в яких відокремлення фаз від видів діяльності істотно полегшує управління розробкою *ПЗ.* Види діяльності, фактично, присутні, під різними назвами, в кожному методі розробки *ПЗ.* У *RUP* вони мають назву робочі *процеси* (*work flow*), в *CMM –* ключовими галузями процесу (key *process area*). Потрібно зберігати традиційні назви, які прийняті в тому або іншому методі, щоб не створювати плутанини.

**Загальносистемні принципи та етапи створення програмного продукту**

При створенні й розвитку ПЗ рекомендується застосовувати такі загальносистемні принципи:

* *принцип включення*, який передбачає, що вимоги до створення, функціонування та розвитку ПЗ визначаються з боку більш складної системи, що включає його в себе;
* *принцип системної єдності*, який полягає в тому, що на всіх стадіях створення, функціонування та розвитку ПЗ його цілісність буде забезпечуватися зв'язками між підсистемами, а також функціонуванням підсистеми управління;
* *принцип розвитку*, який передбачає в ПЗ можливість його нарощування та вдосконалення компонентів і зв'язків між ними;
* *принцип комплексності*, який полягає в тому, що ПЗ забезпечує зв'язаність обробки інформації, як окремих елементів, так і для всього обсягу даних в цілому на всіх стадіях обробки;
* *принцип інформаційної єдності*, тобто у всіх підсистемах, засобах забезпечення і компонентах ПЗ використовуються єдині терміни, символи, умовні позначення і способи подання;
* *принцип сумісності* полягає в тому, що мова, символи, коди та засоби програмного забезпечення узгоджені, забезпечують спільне функціонування всіх підсистем і зберігають відкритою структуру системи в цілому;
* *принцип інваріантності* визначає інваріантність підсистем і компонентів ПЗ до оброблюваної інформації, тобто вони є універсальними або типовими.

**Основні стадії типового процесу створення програмного продукту**

Процес створення програмного забезпечення - сукупність заходів, метою яких є створення або модернізація програмного забезпечення.

Виділяють 4 основні стадії процесу:

1. Специфікація: формулювання основних вимог до ПЗ (що повинна робити система).

2. Розробка: створення ПЗ відповідно до специфікацій.

3. Атестація: перевірка ПЗ на відповідність потребам замовника.

4. Модернізація: розвиток ПЗ відповідно до потреб замовника, що змінилися.



Всі стадії засновані на спеціальних технологіях, наприклад, модульне, структурне, об'єктно-орієнтоване, компонентне програмування відносяться до стадії Розробки. Всі ці стадії можуть бути деталізовані за етапами.

**Етапи створення програмного продукту**

1. *Системний аналіз*. У рамках цього етапу здійснюється аналіз вимог, що пред'являються до програмної системи та програмного продукту зокрема. Він проводиться на основі первинного дослідження всіх потоків інформації при традиційному проведенні робіт і здійснюється в наступній послідовності:
   1. уточнення видів і послідовності всіх робіт;
   2. визначення цілей, які повинні бути досягнуті програмним продуктом, що розробляється;
   3. виявлення аналогів, що забезпечують досягнення подібних цілей, їх переваг та недоліків.
2. *Зовнішнє специфікування*. Полягає у визначенні зовнішніх специфікацій, тобто описів вхідної та вихідної інформації, форм її подання і способів обробки інформації. Реалізується у такій послідовності:
   1. постановка завдання на розробку нової програмного продукту або програмної системи;
   2. оцінка цілей розроблюваного програмного продукту.

Далі, при необхідності, етапи 1-2 можуть бути повторені до досягнення задовільного вигляду програмного продукту або програмної системи з описом виконуваних нею функцій і деякої ясністю реалізації її функціонування.

1. *Проектування програмного продукту.* На цьому етапі проводиться комплекс робіт із формування опису програмного продукту. Вихідними даними для цієї фази є вимоги, викладені у специфікації, розробленої на попередньому етапі. Приймаються рішення, що стосуються способів задоволення вимогам специфікації. Цю фазу розробки програмного продукту поділяють на два етапи:
   1. архітектурне проектування. Являє собою розробку опису програмного продукту у найзагальнішому вигляді. Цей опис містить відомості про можливі варіанти структурної побудови програмного продукту (або у вигляді кількох програм, або у вигляді кількох частин однієї програми), а також про основні алгоритми, і структури даних. Результатом цієї роботи є остаточний варіант архітектури програмної системи, вимоги до структури окремих програмних компонентів і організації файлів для міжпрограмного обміну даними;
   2. робоче проектування. На цьому етапі архітектурний опис програмного продукту деталізується до такого рівня, який робить можливими роботи з її реалізації (кодування і збірці). Для цього здійснюється складання і перевірка специфікацій модулів, складання описів логіки модулів, складання остаточного плану реалізації програми.
2. Кодування і тестування. Ці види діяльності здійснюються для окремих модулів і сукупності готових модулів до отримання готової програми.
3. Комплексне тестування, яке виконується при об'єднанні окремих компонентів у підсистеми або системи з метою перевірки правильної спільної роботи складових частин . При комплексному тестуванні особливу увагу зазвичай приділяється взаємодії компонентів.
4. Розробка експлуатаційної документації.
5. Приймально-здавальні та інші види випробувань.
6. Коригування програмного продукту. Проводиться за результатами попередніх випробувань.
7. Здавання замовнику. Здійснюється остаточна здача програмного продукту замовнику.
8. Тиражування.
9. Супровід програмного продукту. До поняття "супровід" входять усі технічні операції, необхідні для використання програмного продукту у робочому режимі. Сюди входить не тільки виправлення помилок. На цьому етапі також здійснюється модифікація програмного продукту, внесення виправлень у робочу документацію, вдосконалення програмного продукту та інше. Внаслідок широких масштабів подібних операцій супровід є ітеративним процесом, який бажано здійснювати не стільки після, скільки до випуску програмного продукту для широкого використання. Роботи із супроводу часто поглинають більше половини витрат, що припадають на весь життєвий цикл програмної системи у вартісному вираженні.

**"Каркас" процесу розробки програмного забезпечення**

Сучасні технології проектування програмного забезпечення спрямовані на часткову автоматизацію описаних вище етапів і на суміщення їх у часі з метою скорочення термінів виконання проектів.

Кожна організація може організувати процес створення програмного забезпечення так, як їй це представляється розумним. Цей процес може мати різний ступінь формалізації. Не дивлячись на природні відмінності в детальних описах процесів, як правило, в усіх присутні розглянуті стадії. Вони можуть інакше називатися, можуть деталізуватися в той чи інший спосіб, але всі вони присутні.

Існують відомі схеми процесів розробки ПЗ, що досить добре себе зарекомендували, напр. Microsoft Solutions Framework (MSF), Rational Unified Process (RUP) та інші. Ці процеси (часто до них застосовують термін *методологія*) мають свої різновиди і можуть застосовуватися як для малих і середніх компаній й проектів, так і для великих. Однак, всі ці процеси базуються на певних моделях процесу розробки програмних продуктів відповідно до обраного ***ЖЦ ПЗ.*** Отже, зазвичай "каркас" процесу розробки програмного забезпечення виглядає відповідно до переліку стадій: Специфікація → Розробка → Атестація → Модернізація.

Сам цей "каркас" можна приводити в життя по-різному. Існують загальні моделі процесу, які визначають, як організувати роботу по "каркасу" на практиці. Фактично, модель процесу - якесь абстрактне представлення процесу на верхньому рівні. Так, модель не розглядає детально вміст кожного з етапів. Модель розглядає склад етапів і способи їх запровадження.

**Контрольні точки процесу розробки програмного забезпечення**

Очевидно, що функції, виконувані розробниками проекту, в ході його розвитку зазнають змін, як і сам проект. Спочатку він існує у вигляді заявки на розробку, потім - як функціональні й технічні вимоги, далі - як специфікації розроблюваного виробу, набір програмних модулів, скомпонована з модулів система тощо. Цей перелік можна розглядати як один із прикладів моделі життєвого циклу програмного продукту, тобто подання еволюції розробки та подальшого використання програмної системи.

ЖЦ ПЗ потрібно розглядати як основу діяльності менеджера програмного проекту: з ним пов'язано і цілі проекту - остаточні й проміжні, розподіл і контроль витрачання ресурсів, а також усі інші аспекти управління розвитком проекту. Перш за все цю прив'язку обумовлено розбивкою виробництва будь-якого програмного продукту на етапи, які асоціюються з певними видами робіт або функцій, виконуваних розробниками у той чи інший момент розвитку проекту. Етапи характеризуються спрямованістю виконуваних функцій на досягнення локальних (для етапу) цілей проекту. Необхідність відстеження цілей призводить до поняття контрольних точок - моментів розробки, коли здійснюється підведення проміжних підсумків, осмислення досягнутого і ревізія зроблених раніше припущень.

Звідси випливає, що контрольні точки є постійною турботою менеджера проекту і моментами, коли інтенсивність його роботи зростає. Разом з тим визначення контрольних точок - це елемент планування, який знаходиться в компетенції менеджера. У першу чергу планування часу, а на базі його - розподілу інших ресурсів. Є певна свобода у виборі етапів і контрольних точок, обмежена зобов'язаннями перед замовниками, розробниками, а також планувальниками компанії. Це означає доцільність пристосування етапів розвитку проекту до його специфіки та до специфіки умов виконання завдання. У рамках обговорення менеджменту програмних проектів питання життєвого циклу повинні розглядатися як першорядні.

**Конструювання та проектування програмного забезпечення**

Термін конструювання програмного забезпечення (software construction) описує детальне створення робочої програмної системи за допомогою комбінації кодування, верифікації (перевірки), модульного тестування (unit testing), інтеграційного тестування і налагодження. Конструювання ПЗ – створення ПЗ з конструкцій (блоків, операторів, функцій) і його перевірка методами верифікації і тестування.

Проектування програмного забезпечення починається, власне, з його конструювання, яке визначає стратегію для його внутрішнього проектування - для етапу програмування. Цей етап виконується без використання мови програмування, але з орієнтацією на певний програмний інструмент розробки ПЗ.

**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.

**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, що включає
  + звіт про відносну оцінку рішення, що містить інформацію про всі можливі рішення і обґрунтування ухвалених рішень
  + представлення необхідних ресурсів - штату, апаратних засобів, програмного забезпечення і т.д.
  + стандартні визначення
  + планування аналізу

*Для самостійного вивчення* *(2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література*

1. Лавріщева К.М. Програмна інженерія. Електронний підручник. URL:  http://csc.knu.ua/uk/library/books/lavrishcheva-6.pdf.

2.  Лавріщева К.М. Програмна інженерія. – Підручник.–К.:Академперіодика, 2008.–415с.

3. Бабенко Л.П., Лавріщева К.М*.* Основи програмної інженерії.– Навч. посібник.–К.: Знання, 2001.– 269с.

4. Проектування інформаційних систем: Посібник // За редакцією Пономаренка В.С. – К.: Видавничий центр "Академія". 2002. ­ 488 с. URL: [http://www.dut.edu.ua/uploads/l\_874\_10304054.pdf](https://www.blogger.com/).

5.  И. Соммервиль. Инженерия программного обеспечения, 6 изд. – И.д. "Вильямс", 2002. URL: <https://www.studmed.ru/download/sommervill-ian-inzheneriya-programmnogo-obespecheniya_4935164f089.html>

6. Боэм Б. Инженерное проектирование программного обеспечения. — М.: Радио и связь, 1985.

*Запитання для самоперевірки.*

1. В чому полягає процес створення ПЗ?
2. Які причини вдосконалення процесу створення ПЗ?
3. Визначтестратегії впровадження інновацій в розробці ПЗ.
4. Визначте загальносистемні принципи розробки програмного забезпечення.
5. Визначте основністадії типового процесу створення програмного продукту та його основні етапи.
6. Для чого потрібні контрольні точки процесу розробки програмного забезпечення?
7. З яких дій складається етап проектування ПЗ?
8. Чому проектування **ПЗ визначається як стратегічний етап?**

*Запитання для письмової відповіді.*

1. Поясніть для чого потрібно проводити стандартизацію процедур процесів розробки ПЗ та які від цього вигоди отримує організація?
2. Чи можна об’єднувати етапи проектування та конструювання? Відповідь поясніть.