**Лабораторна робота №3. Проектування програмного забезпечення за допомогою каскадної (водоспадної) моделі**

**Тема:** Проектування програмного забезпечення за допомогою каскадної (водоспадної) моделі

**Мета:** Навчитись проектувати гіпотетичне програмне забезпечення, спираючись на принципи, що закладені в каскадній моделі.

**КОНТРОЛЬНІ ПИТАННЯ:**

1. Що таке модель конструювання?
2. Що таке життєвий цикл програмного забезпечення.
3. Які різновиди моделей життєвого циклу програмного забезпечення Вам відомі?
4. Назвіть основні характеристики і фази каскадної моделі.
5. Назвіть основні переваги каскадної моделі?
6. Назвіть основні недоліки каскадної моделі?

**Завдання:** Ознайомтесь з основними принципами каскадної моделі, за темою та сформуйте детальний перелік робіт щодо розробки програмного забезпечення згідно етапів каскадної моделі для власного проекту. Кожний етап повинен бути супроводжений документацією (визначити перелік документів, необхідний для кожного етапу).

**Методичні вказівки до виконання роботи:**

1. Ознайомитися з теоретичною частиною.
2. Робота повинна бути виконана згідно критеріїв оформлення документації на аркушах формату А4.
3. Студентам перед виконання лабораторної роботи потрібно ознайомитись із необхідним теоретичним матеріалом, потім обрати тему. В якості теми можна взяти тему попередніх лабораторних робіт (конструювання програмного забезпечення для оброблення результатів анкетування) або обрати власну тему, узгодивши її з викладачем.
4. Спочатку рекомендується проговорити можливі варіанти виконання.
5. Якщо потрібні пояснення – можна звернутись до викладача. Викладач сам вирішує відповідати на питання чи надати підказки, де можна знайти відповідь.
6. Протягом виконання лабораторної роботи студенти можуть використовувати конспекти або інші джерела інформації.
7. Роботу потрібно набирати на комп’ютері, що розташований в аудиторії або на власному ноутбукові, назвавши документ «Лабораторна робота № ».
8. По закінченню лабораторну роботу потрібно здати на перевірку викладачеві, роздрукувавши її або надіславши електронною поштою. Якщо викладач знаходить помилки чи неточності, він може повернути роботу на доопрацювання.

По закінченню практичну роботу потрібно здати на перевірку викладачеві, надіславши електронною поштою на адресу [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com) . Якщо викладач знаходить помилки чи неточності, він може повернути роботу на доопрацювання.

Файл з роботою повинен мати назву в такому форматі:

**KPZ <Номер групи><Номер лекції / лабораторної> [літера позначення типу роботи L – лекція, R – лабораторна]<Прізвище англійською>**.. Наприклад, **KPZ4101R**buts.doc.

Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності робіт-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

Тему в заголовку листа записати

**KPZ<Номер групи>-ЛР<Номер лабораторної>-<Прізвище англійською>**

**Строк виконання цієї роботи ІПЗ-41 – 06.10.2020**

**ІПЗ-42 – 04.10.2020**

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Класична каскадна модель, незважаючи на отриману останнім часом негативну оцінку, справно служила фахівцям з програмного інжинірингу багато років. Розуміння її сильних сторін і недоліків покращує оцінний аналіз інших, найчастіше більш ефективних моделей життєвого циклу, заснованих на даній моделі.

У перші роки практики програмування спочатку записувався програмний код, а потім відбувалася його налагодження. Загальноприйнятим вважалося правило починати роботу не з розробки плану, а з загального ознайомлення з продуктом. Без зайвих формальностей можна було спроектувати, закодувати, налагодити і протестувати ПО ще до того, як воно буде готове до випуску. Це нагадувало процес, зображений на рис. 1. У структурі такого процесу є декілька недоліків.

По-перше, оскільки спочатку не існувало офіційного проекту або аналізу, неможливо було впізнати про момент завершення процесу. Також був відсутній спосіб визначення відповідності вимогам щодо досягнення якості.

Выходы

Входы

Кодирование и тестирование

Делать, пока не будет сделано

***Рис. 1. Модель процесса "делать, пока, не будет сделано”***

У 1970 році каскадна модель була вперше визначена як альтернативний варіант методу розробки ПЗ за принципом кодування-усунення помилок, який був широко поширений в той час. Це була перша модель, яка формалізувала структуру етапів розробки ПЗ, надаючи особливого значення вихідним вимогам і проектування, а також створенню документації на ранніх етапах процесу розробки.

Початковий етап виконання каскадної моделі показаний у лівій верхній частині рис. 2. Продовження процесу виконання реалізується за допомогою впорядкованої послідовності кроків. У моделі передбачено, що кожна наступна фаза починається лише тоді, коли повністю завершено виконання попередньої фази Кожна фаза має певні критерії входу і виходу: вхідні і вихідні дані.

В результаті виконання генеруються внутрішні чи зовнішні дані проекту, включай документацію та ПЗ. Документи по аналізу вимог згодом передаються системним фахівцям, які в свою чергу передають їх розробника програмних систем більш високого рівня. Програмісти передають детальні технічні характеристики програмістам, які вже представляють готовий код тестерам.

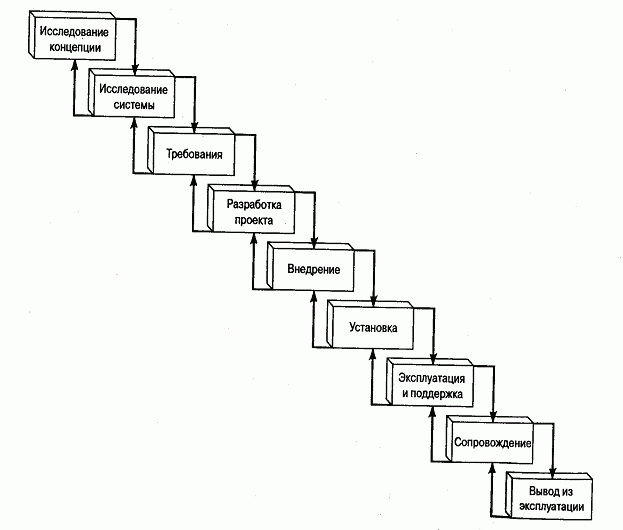
Перехід від однієї фази до іншої здійснюється за допомогою формального огляду. Таким чином, клієнт отримує загальне уявлення про процес розробки, крім того відбувається перевірка якості програмного продукту. Як правило, проходження стадії огляду вказує на домовленість між командою розробників і клієнтом про те, що поточна фаза завершена і можна перейти до виконання наступної фази. Закінчення фази зручно приймати за стадію в процесі виконання проекту.

В результаті завершення певних фаз формується базова лінія, яка в даній точці "заморожує" продукти розробки. Якщо виникає потреба в їх зміну, тоді для внесення змін використовується формальний процес змін.

У критичних точках каскадної моделі формуються базові лінії, остання з яких є базовою лінією продукту. Після формування заключній базової лінії проводиться огляд приймання.

***Короткий опис фаз каскадної моделі***

Наведена нижче характеристика являє собою короткий опис кожної фази каскадної моделі (включаючи фази інтеграції):

* *дослідження концепції* - відбувається дослідження вимог на системному рівні з метою визначення можливості реалізації концепції;
* *процес системного розподілу* - може бути пропущений для систем з розробки виключно ПЗ. Для систем, в яких необхідна розробка як апаратного, так і програмного забезпечення, необхідні функції застосовуються до ПЗ та обладнанню відповідно із загальною архітектурою системи;
* *процес визначення вимог* - визначаються програмні вимоги для інформаційної предметної області системи, призначення, лінії поведінки, продуктивність і інтерфейси. (В разі необхідності в процес також включено 

***Рис.2. Каскадна модель проектування програмного забезпечення***

* функціональний розподіл системних вимог до апаратному і програмному забезпеченню.);
* *процес розробки проекту* - розробляється і формулюється логічно послідовна технічна характеристика програмної системи, включаючи структури даних, архітектуру ПО, інтерфейсні уявлення і процесуальну (алгоритмічну) деталізацію;
* *процес реалізації* - в результаті його виконання ескізне опис ПО перетворюється на повноцінний програмний продукт. При цьому створюється вихідний код, база даних і документація, які лежать в основі фізичного перетворення проекту. Якщо програмний продукт являє собою придбаний пакет прикладних програм, основними діями по його реалізації будуть установка і тестування пакета програм. Якщо програмний продукт розробляється на замовлення, основними діями є програмування та код-тестування;
* *процес установки* - включає установку ПО, його перевірку і офіційну приймання замовником для операційного середовища;
* *процес експлуатації та підтримки* - має на увазі запуск користувачем системи і поточне забезпечення, включаючи надання технічної допомоги, обговорення питань, що виникли з користувачем, реєстрацію запитів користувача на модернізацію та внесення змін, а також коригування або усунення помилок;
* *процес супроводження* - пов'язаний з дозволом програмних помилок, несправностей, збоїв, модернізацією та внесенням змін, що генеруються процесом підтримки. Складається з ітерацій розробки і передбачає зворотний зв'язок з надання інформації про аномалії;
* *процес виведення з експлуата*ції - висновок існуючої системи з її активного використання або шляхом припинення її роботи, або завдяки її заміні новій системою або модернізованою версією існуючої системи;
* *інтегральні завдання* - включають початок роботи над проектом, моніторинг проекту та його управління, управління якістю, верифікацію та атестацію, менеджмент конфігурації, розробку документації та професійну підготовку протягом усього життєвого циклу.

**Основні документи проекту по розробці ПЗ**

* специфікація - перелік і призначення всіх файлів програмного виробу, включаючи файли документації;
* відомість власників оригіналів - список підприємств, які зберігають оригінали програмних документів, складається тільки для складних програмних виробів;
* текст програми - запис кодів програми та коментарі до них;
* опис програми - інформація про логічну структуру та функціонування програми;
* програма і методика випробувань - перелік і опис вимог, які повинні бути перевірені в ході випробування програми, методи контролю;
* технічне завдання - документ, в якому викладаються призначення і область застосування програми, вимоги до програмного виробу, стадії і терміни розробки, види випробувань;
* пояснювальна записка - обґрунтування прийнятих і застосованих технічних і техніко-економічних рішень, схеми та опис алгоритмів, загальний опис роботи програмного виробу;
* відомість експлуатаційних документів - містить список експлуатаційних документів на програмний виріб, до яких відносяться формуляр, опис застосування, керівництво системного програміста, керівництво програміста, керівництво оператора, опис мови, керівництво з технічного обслуговування;
* формуляр - містить основні характеристики програмного виробу, склад і відомості про експлуатацію програми;
* опис застосування - містить інформацію про призначення та галузі застосування програмного виробу, обмеження при застосуванні, клас і методи вирішуваних завдань, конфігурацію технічних засобів;
* керівництво системного програміста - містить відомості для перевірки, настроювання і функціонування програми при конкретному застосуванні;
* керівництво програміста - містить відомості для експлуатації програмного виробу;
* керівництво оператора - містить докладну інформацію для користувача, який забезпечує його спілкування з ЕОМ у процесі виконання програми;
* опис мови - містить синтаксис і семантику мови;
* керівництво з технічного обслуговування - містить відомості для застосування тестових і діагностичних програм при обслуговуванні технічних засобів.

**Склад і зміст робіт на стадії проектування**

На цій стадії щодо розроблення робочої документації ведуться роботи щодо практичної реалізації положень, закладених в проекті. Робоча документація розробляється на основі затверджених технічного завдання і технічного проекту і затвердженню не підлягає.

Мета робочого проектування – складання технічної документації для налагодження і впровадження ІС, для проведення приймально здавальних досліджень, а також для забезпечення нормального забезпечення функціонування ІС.

Згідно з ГОСТ 34.601–90 «Стадии создания» стадію можна поділити на два етапи.

1. Розробка робочої документації на систему та її частини.

2. Розробка чи адаптація програм.

На першому етапі здійснюється розробка робочої документації, яка містить всі необхідні й достатні відомості для забезпечення виконання робіт з введення інформаційної системи в дію та її експлуатації, а також для підтримання рівня експлуатаційних характеристик (якості) системи згідно з прийнятими проектними рішеннями, її оформлення, погодження і затвердження. Це технологічні інструкції з обробки даних, інструкції для роботи в умовах функціонування розробленої системи, порадник користувача.

Можна виділити такі роботи:

1) прийняття рішень щодо організації розробки робочої документації;

2) розробка загальносистемних проектних рішень;

3) розробка проектної документації з видів забезпечення;

4) оформлення, погодження, затвердження в установленому порядку.

З інформаційного забезпечення:

1) розробка експлуатаційної документації з інформаційного забезпечення, до якої входить розробка уніфікованих форм документів і підготовка класифікаторів;

2) перевірка інформаційно-логічної структури бази даних;

3) розробка інструкції для створення і ведення бази даних.

З організаційного забезпечення:

1) уточнення функцій і конкретизація складу робіт персоналу інформаційної системи, розробка положень та інструкцій усіх видів, формуляру системи;

2) розробка технологічного процесу обробки даних, який складається з технологічного процесу отримання і обробки даних. Розробка

порадника користувача.

На другому етапі здійснюється адаптація чи прив’язка пакетів прикладних програм (ППП) і окремих програм, розробка програм, програмних та інформаційних виробів, програмної і експлуатаційної документації.

У ході розробки робочої документації замовник:

1) забезпечує приймання до дослідної експлуатації;

2) завершує під методичним керівництвом розробника формування інформаційної бази і організовує її експлуатацію;

3) передає на вимогу розробника дані, необхідні для перевірки задач на контрольних приладах;

4) організовує приймання програм, надаючи розробникові магнітні носії та машинний час;

5) виконує основні організаційно-технічні заходи з підготовки підприємства до введення інформаційної системи в дію.

Розробник розробляє та оформлює технічну документацію з усіх видів забезпечення і перевіряє програми.

**Склад проектної документації на стадіях «Технічний проект» і «Робоча документація» (робочий проект)**

Склад і структура проектної документації на стадіях «Технічний проект» і «Робоча документація» визначаються ГОСТ 34.201–89 «Виды, комплектность и обозначение документов при создании автоматизированных систем».

Ці документи можна поділити на три частини.

1. Документація на інформаційну систему – комплекс взаємопов’язаних документів, у яких повністю описано всі рішення щодо створення і функціонування системи, а також документи, що підтверджують відповідність системи вимогам технічного завдання і готовність її до експлуатації.

2. Проектно-кошторисна документація на інформаційну систему – частина документації на ІС, яка розробляється для виконання будівельних і монтажних робіт, пов’язаних зі створенням інформаційної системи.

3. Робоча документація на суміжні частини ІС – частина документації на ІС, яка необхідна для виготовлення, будівництва, монтажу і налагодження інформаційної системи в цілому, а також та документація, яка входить до системи програмно-технічних, програмно-методичних комплексів і компонентів технічного, програмного та інформаційного забезпечення.

Перелік найменувань розроблюваних документів і їх комплектність на систему та її частини мають бути визначені в технічному завданні на створення ІС чи її компонента.

Документацію технічного проекту поділено на такі частини.

**Загальносистемні рішення**

1. Пояснювальна записка до технічного проекту.

2. Відомість технічного проекту.

3. Схема організаційної структури.

4. Схема функціональної структури.

5. Опис функцій, що автоматизуються.

6. Опис постановки задач.

7. Локальний кошторисний розрахунок.

8. Проектна оцінка надійності системи.

За видами забезпечення:

**Технічне забезпечення**

1. Схема структурна комплексу технічних засобів.

2. Схема автоматизації.

3. Опис комплексу технічних засобів.

4. План розміщення.

5. Перелік завдань на розробку будівельних, електромонтажних та інших робіт, пов’язаних зі створенням системи.

**Організаційне забезпечення**

1. Опис організаційної структури.

**Інформаційне забезпечення**

1. Опис інформаційного забезпечення системи.

2. Опис організації інформаційної бази.

3. Перелік вхідних сигналів і даних.

4. Перелік вихідних сигналів.

5. Опис системи класифікації та кодування.

6. Опис масиву інформації.

7. Креслення форми документа (відеокадру).

**Програмне забезпечення**

1. Опис програмного забезпечення.

**Математичне забезпечення**

1. Опис алгоритму.

Робочу документацію також можна поділити на дві частини.

**Загальносистемні рішення**

1. Формуляр.

2. Паспорт.

3. Загальний опис системи.

4. Програма і методика випробувань.

5. Відомість тримачів оригіналів.

6. Відомість експлуатаційних документів.

7. Проектна оцінка надійності системи.

8. Локальний кошторис.

За видами забезпечення:

**Технічне забезпечення**

1. Специфікація обладнання.

2. Відомість потреб у матеріалах.

3. Інструкція з експлуатації КТЗ.

4. Схема структурна КТЗ.

5. Схема поділу системи.

**Організаційне забезпечення**

1. Методика (технологія) автоматизованого проектування.

2. Технологічні інструкції.

3. Керівництво користувача.

4. Опис технологічного процесу обробки даних.

**Інформаційне забезпечення**

1. Каталог бази даних.

2. Відомість машинних носіїв інформації.

3. Склад вхідних даних (повідомлень).

4. Масив вхідних даних.

5. Креслення форм документів (відеокадрів).

6. Інструкція з формування і ведення бази даних (набору даних).

До цього переліку не включено проектно-кошторисну докумен-тацію.

**Визначення структури інформаційної системи**

Основу системної побудови інформаційної системи становить її структура, яка має бути досить повною. Засобами структуризації є процедури декомпозиції (аналізу) і композиції (синтезу) системи. Тому важливим етапом проектування інформаційної системи є структуризація системи – локалізація її меж і виділення структурних складових частин.

Кожний економічний об’єкт не існує повністю автономно, він безпосередньо входить до складу зовнішнього середовища, всього того, що не входить до складу об’єкта, але тією чи іншою мірою впливає на його існування.

Також практично всі традиційні системи побудовані за ієрархічною схемою, але завжди в екстремальних ситуаціях керівник вищого рівня може в певній ситуації взяти керівництво на себе, тобто на час ситуації змінити кількість рівнів ієрархії, тоді як у інших випадках усе може залишатися без змін.

Автоматизація інформаційної системи робить структуру управління жорсткою, особливо в умовах використання середніх і великих ЕОМ. Нині застосування персоналізації з установленням персональної ЕОМ чи АРМ у мережах може призвести до змінної структури, яка має властивості гнучкості традиційних систем.

1. З позиції ресурсної моделі багаторівнева структура системи управління має переваги над однорівневою, оскільки дозволяє зменшити ресурси забезпечення неперервного управління.

2. За складністю побудови і реалізації алгоритмів управління багаторівнева система має переваги перед однорівневою і не лише спрощує систему управління, а й зменшує інформаційне навантаження на елементи системи; на практиці для економічних об’єктів (підприємств) найкращою виявляється дво-трирівнева система.

3. Багаторівнева структура системи керує роботою каналів зв’язку в системі управління і полегшує її, зменшуючи при цьому надмірність інформації і забезпечуючи вибіркове інформаційне обслуговування осіб, які приймають рішення.

4. У ході реалізації системи управління ієрархічність її моделі не висуває додаткових жорстких вимог до технічних засобів системи і каналів зв’язку.

Тому багаторівнева система є все ж досить гнучкою і допускає створення різної кількості рівнів для різних управлінських задач. Для цього потрібно побудувати чи вдосконалити модель управління економічним об’єктом і визначити структуру інформаційної системи. При побудові моделі виявляється:

1) повна множина циклів управління (довгострокове, поточне, оперативне і т.п.);

2) структура управління ( одно-, дво-, трирівнева і т. д.);

3) система економічних показників, що відповідають діючій методології управління і характеризують усе різноманіття станів, які об’єктивно має об’єкт управління.

Визначення структури і поділ системи на складові можна здійснити за чотири етапи.

1. Як зазначалося, за основу поділу системи можна взяти один, чи декілька поділів або їх поєднання: ресурсний, функціональний, адміністра­тивний і т.д.

2. Надалі визначають необхідний перелік задач різного ступеня деталізації. Задача інформаційної системи: функція чи частина функції інформаційної системи, що являє собою формалізовану сукупність автоматичних дій, виконання яких приводить до результату заданого виду (ГОСТ 34.003–90). Різноманітність розв'язуваних у комп'ютерних інформацій­них системах задач потребує їхньої класифікації.

За функціями управління розрізняють планові, облікові, кон­трольні задачі, задачі нормування показників, складання звітності і т.ін.

За характером перетворення інформації задачі поді­ляються на обчислювальні, імітаційні, прийняття рішень.

За роллю в процесі управління розрізнюють інженерно-технічні, економічні та інформаційно-довідкові задачі.

За математичною суттю задачі поділяються на оптимізаційні, прямого розрахунку та інформаційно-пошукові.

Оптимізаційні задачі розв'язуються шляхом пошуку одного рішення із великої кількості можливих варіантів. Вони характери­зуються складною методикою розрахунків (що зумовлює необхід­ність використання різноманітних моделей), а також відносно не­великими розмірами вхідних даних.

В основній своїй масі задачі сучасної комп'ютерної ІС належать до задач прямого розрахунку. Для них характерні великі розміри і складність вхідних даних, проста методика розрахунку і одноваріантність розв'язання-інформаційно-пошукові задачі, тобто задачі типу «запитан­ня — відповідь» характеризуються складною методикою розрахунку та значними розмірами вхідної інформації.

За можливістю формалізованого опису задачі ІС поділяються на формалізовні та неформалізовні. Розв'язування перших можна описати у вигляді математичних формул та залежностей, щодо других — цього зробити не можна.

За регулярністю розв'язування задачі ІС поділяються на сис­тематичні, епізодичні та випадкові.

3. На основі моделі функціональної частини визначають необхідність розв’язання тих чи інших прикладних задач ІС різного ступеня деталізації. Кількість задач може бути різною. Визначають можливий перелік задач (надмірний), складають таблицю-специфікацію задач і синтезують їх.

4. Метою аналізу системи прикладних задач є погодження їх за виходами, входами та інформаційними переходами для усунення в системі повторних перетворень інформації, її дублювання в різних задачах.

За такого погодження з’являється можливість вилучити деякі задачі із розробки, об’єднати пари задач чи їх комплекси в один об’єкт проектування чи спростити задачу вилученням із неї обчислювальних процедур, які повторюються в інших задачах, вихідних і вхідних даних, задачу можна усунути тоді, коли вихідні показники деякої сукупності задач збігаються або множина вихідних показників однієї задачі містить множину вихідних показників іншої задачі цієї сукупності.

Об’єднувати задачі можна в тих чи інших випадках, коли:

1) серед вихідних показників двох чи більше задач є як спільні, так і різні показники, причому різних значно менше;

2) у деякій сукупності задач усі вихідні показники чи більшість із них і періодичність розв’язання цих задач збігаються;

3) у двох чи кількох задачах вихідні показники одних задач збігаються з вихідними інших.

Спростити дві задачі чи більше можна в тому випадку, коли вони мають як спільні, так і різні вихідні показники, але різних більше, ніж спільних.

При великій кількості задач здійснити ефективне погодження можна лише за допомогою ЕОМ.

Із викладеного випливає, що ні локальний, ні глобальний підходи не забезпечують тотожності проекту ІС її об’єктивним властивостям. Для забезпечення тотожності в ході системного проектування ІС здійснюють спільно і в органічній єдності впродовж розвитку системи процеси її макро- і мікропроектування з урахуванням системного принципу.

Макропроектування – це моделювання мети, призначення, побудови, функціонування, розвитку та інших несистемних атрибутів ІС і її складових частин. У результаті формування моделі визначають мету, призначення, сфера використання, вимоги до кінцевих проектних рішень. Моделі визначають створення, функціонування і розвиток об’єкта проектування.

Мікропроектування – це перетворення перелічених моделей у кінцеві проектні рішення, що впроваджуються на об’єкті, але спочатку потрібно з’ясувати, які задачі виконувала людина, а які – ЕОМ.