ЖДТУ	Міністерство освіти і науки України Житомирський державний технологічний університет

«ЗАТВЕРДЖУЮ»
Голова
Вченої ради факультету ФІКТ

_____
_____
«_____»_____20___р.

ЗАВДАННЯ ДЛЯ САМОСТІЙНИХ РОБІТ З ДИСЦИПЛІНИ

«Стандарти ІС та сертифікація ПЗ»

для студентів освітнього рівня «магістр»
спеціальності 121«Інженерія програмного забезпечення»
освітньо-професійна програма «Інженерія програмного забезпечення»
факультет інформаційно-комп'ютерних технологій
кафедра інженерії програмного забезпечення

Робочу програму схвалено на засіданні кафедри інженерії програмного забезпечення
протокол від «28» серпня 2018р. No 1
Завідувач кафедри
інженерії програмного забезпечення
_____
А.В.Панішев

Розробники: к.т.н., доцент кафедри ІПЗ Єфремов Ю.М.
к.т.н., доцент кафедри ІПЗ Єфремов М.Ф.

Житомир 2018–2019 н.р.

№ з/п	Назва робот	Кількість годин
1	Класифікація ІС за ступенем централізації та інтеграції обробки даних	10
2	Процеси, що мають перебіг у життєвому циклі інформаційної системи	6
3	Каскадна модель життєвого циклу інформаційної системи	10
4	Парадигмі проектування інформаційних систем	8
5	Стандарти проектування інформаційних систем. Методологія CDM	8
6	Корпоративні стандарти	10
7	Канонічне проектування інформаційних систем та його нормативне забезпечення	6
Усього годин		58

Класифікація ІС за ступенем централізації та інтеграції обробки даних.

За ступенем централізації обробки даних розрізняють: централізовані ІС, децентралізовані ІС, інформаційні системи колективного використання [1–4]. До централізованих ІС належать такі, де накопичення й обробка інформації здійснюється в єдиному центрі. У цьому разі доступ до самої ІС організовано з одного або багатьох терміналів. Децентралізовані ІС є більш автономними. Кожна децентралізована ІС певного рівня обслуговує певну верству користувачів. Прикладом децентралізованої ІС може слугувати система обробки статистичних даних. Для неї характерно те, що ІС районного рівня обслуговує певний район, ІС обласного рівня обслуговує певну область тощо. Інформаційні системи колективного користування відрізняються тим, що доступ до них має велика кількість користувачів. Залежно від ступеня інтеграції функцій розрізняють: багаторівневі ІС з інтеграцією за рівнями управління (підприємство – об'єднання, об'єднання – галузь й т. і), багаторівневі ІС з інтеграцією за рівнями планування. Одноріневі інформаційні системи обслуговують переважно окремі підрозділи управління або виробництва. Одноріневою може бути система планово-фінансового відділу, автоматизована системи диспетчера виробництва, працівника складу і т. п.

Багаторівневим інформаційним системам притаманні інтеграція функцій за різними аспектами управління. Так, до багаторівневих здебільшого належать системи автоматизації бухгалтерського обліку, комплексного управління виробництвом, системи ресурсного забезпечення, загальнодержавні системи тощо.

Процеси, що мають перебіг у життєвому циклі інформаційної системи

Життєвий цикл – це безперервний процес, що починається з моменту рішення про створення ІС і закінчується після вилучення її з експлуатації [1–3]. Регламентує життєвий цикл інформаційних систем міжнародний стандарт ISO/IEC 12207. 2 Стандарт ISO/IEC 12207 визначає структуру життєвого циклу, його процеси, дії, завдання, які мають бути виконані під час створення ІС. Згідно цього стандарту, структура життєвого циклу обіймає три групи процесів:

– головні, – придбання, постачання, розроблення, експлуатація, супроводження;

– допоміжні, що забезпечують виконання головних процесів (документування, забезпечення якості, тестування, атестація, оцінкаефективності, аудит, ліцензування, сертифікація тощо);

– організаційні, – управління проектами, створення інфраструктури проекту, навчання.

Головні процеси життєвого циклу

Життєвий цикл ІС включає три основні процеси: розробка, експлуатація та супровід (супроводження).

Розробка

Цей процес поєднує всі роботи щодо створення інформаційного та програмного забезпечення та його компонентів відповідно до вимог техзавдання. Розробка інформаційного програмного забезпечення обіймає також такі питання, як:

– оформлення проектної та експлуатаційної документації (ЕД);

– підготовку матеріалів для тестування програмних продуктів;

– розробку матеріалів для навчання персоналу.

Розробка – один із найважливіших процесів життєвого циклу ІС. Як правило, він передбачає стратегічне планування, аналіз, проектування та реалізацію програмних продуктів.

Експлуатація

Умовно роботи на стадії експлуатації можна поділити на головні та підготовчі. До головних робіт на стадії експлуатації належать:

– експлуатація;

– локалізація проблем та усунення причин їхнього виникнення;

– модифікація (або адаптація) програмного забезпечення;

– підготовка пропозицій щодо удосконалення системи;

– розвиток та подальша модернізація системи.

Підготовчі роботи зазвичай включають:

– конфігурацію бази даних і робочих місць користувачів;

– забезпечення користувачів експлуатаційною документації;

– навчання (або підвищення кваліфікації) персоналу.

Супроводження

Технічна підтримка – дуже важлива складова успішної експлуатації ІС, необхідна умова вирішення поставлених перед нею завдань. Слід зауважити, що

помилки обслуговуючого персоналу можуть призводити до фінансових втрат, які можна порівняти з вартістю самої системи.

Технічна підтримка – досить складний та довготривалий процес. Тому для його успішної реалізації дуже важлива організаційна складова, під час якої слід виконати такі дії:

– виділити ключові складові системи, оптимізувати розподіл ресурсів для їхнього технічного обслуговування (ТО);

– визначити завдання щодо ТО, розділити їх на внутрішні (що вирішуються власними силами) та зовнішні (вирішуються спеціалізованими сервісними організаціями); забезпечити чіткий розподіл функцій та відповідальності;

– провести аудит ресурсів, необхідних для організації ТО, виконати розподіл компетенцій;

– підготувати план організації ТО, в якому конкретизувати етапи дій, терміни виконання, витрати за кожним етапом, відповідальність виконавців.

Якісне технічне обслуговування потребує залучення висококваліфікованих фахівців, які здатні не тільки вирішувати поточні завдання, а й швидко відновлювати дієвість системи в аварійних ситуаціях.

Допоміжні процеси життєвого циклу

Серед допоміжних провідне місце займає процес управління конфігурацією, який підтримує головні етапи життєвого циклу ІС, зокрема – процеси розробки та супроводження. Під час розробки складних ІС кожен компонент може розроблятися незалежно, мати декілька варіантів або версій реалізації. У такому випадку виникає проблема щодо обліку зв’язків і функцій між ними, створення єдиної структури для забезпечення розвитку всієї системи. Цю проблему вирішує управління конфігурацією, що дозволяє системно контролювати внесення змін до різних компонент ІС на всіх стадіях ЖЦ.

Організаційні процеси

Управління проектом щільно пов’язано з плануванням і організацією робіт, створенням колективу проектувальників, з контролем за термінами та якістю реалізації всіх етапів розробки ІС. Вирішення цих питань покладено на технічне й організаційне забезпечення проекту, яке передбачає:

– вибір методів та інструментальних засобів для реалізації проекту;

– визначення методів щодо опису всіх проміжних станів розробки;

– розробка методів і засобів для випробувань ПЗ;

– навчання та підвищення кваліфікації персоналу.

Забезпечення якості проекту щільно пов’язане верифікації 3 , перевірки й тестування компонентів ІС.з проблемами

Перевірка – це процес, мета якого – визначення відповідності параметрів ІС вихідним вимогам. Перевірка певною мірою схожа з тестуванням, яке має за мету визначення розбіжностей між дійсними та очікуваними результатами проекту та оцінкою відповідності характеристик ІС вихідним вимогам.

Каскадна модель життєвого циклу інформаційної системи

Згідно каскадної моделі всі роботи виконуються. При цьому вся розробка розбивається на етапи, перехід до наступного етапу відбувається після повного завершення всіх робіт попереднього етапу.

Кожен етап завершується оформленням повного комплекту документації. Склад і зміст цієї документації передбачає, що реалізація проекту може бути продовжена іншою командою розробників.

Головні етапи розроблення згідно з каскадною моделлю Враховуючи досвід реальних розробок, у каскадній моделі прийнято виділяти декілька сталих етапів, що майже не залежать від предметної області , а саме:

– аналіз вимог замовника;
– проектування;
– розробка;
– тестування і дослідна експлуатація;
– здавання готового продукту.

Результатом першого етапу є ТЗ (завдання на розробку). Воно має бути узгоджене з усіма сторонами. Результат другого етапу – комплект проектної документації (ПД), що містить необхідні дані для реалізації проекту. Результат третього етапу – готовий програмний продукт.

На четвертому етапі виявляють приховані недоліки, які виявились за умов експлуатації, а також вносять відповідні коригування до програмного забезпечення.

Головне завдання п’ятого етапу – переконати замовника у тому, що всі вимоги виконані повною мірою. На практиці ЖЦ реальної системи може істотно відрізнятись, бути складнішим і довшим. Він може мати довільну кількість циклів уточнення, змін і доповнень вже реалізованих проектних рішень. до того ж саме у таких циклах відбувається розвиток ІС й модернізація її компонентів.

Переваги каскадної моделі

Серед переваг каскадної моделі передусім можна вказати на такі:

– на кожному етапі формується повний та узгоджений набір ПД. На завершальних етапах також розробляється документація, що охоплює всі передбачені стандартами різновиди забезпечення ІС (організаційне, методичне, інформаційне, програмне, апаратне).
– чітка послідовність етапів дозволяє планувати терміни виконання робіт та відповідні витрати.

Каскадний підхід добре проявив себе під час розробки певного класу ІС. В першу чергу це системи, де з самого початку можна окреслити всі вимоги, що дає розробникам свободу щодо шляхів реалізації.

Недоліки каскадної моделі

Кількість недоліків каскадної моделі досить значна, а саме:

– повільність щодо отримання результатів;
– помилки на будь-якому етапі впливають на подальшу роботу;
– у межах каскадної моделі складно організувати паралельне ведення робіт;

– має місце певна інформаційна перенасиченість на всіх етапіх проекту;
– ускладнено управління проектом;
– значний рівень ризику та ненадійність інвестицій.

Щоб зрозуміти область можливого використання каскадної моделі розглянемо вказані недоліки біль детально.

Затримка в отриманні результатів, – це головний недолік каскадної моделі, що є наслідком послідовного підходу. Адже у цьому випадку узгодження результатів проводиться тільки після завершення чергового етапу робіт.

Крім того, узгодження результатів часто проводиться вибірково, після завершення кожного етапу, вимоги до ІС «заморожені» у вигляді технічного завдання на весь час її створення. Тому користувачі можуть надати свої пропозиції та зауваження тільки після завершення робіт над системою. У разі неточного формулювання вимог або їхньої зміни за період створення ПО користувачі отримують систему, що не задовольняє їхнім потребам.

До того ж задіяні на початок розробки моделі об’єктів можуть застаріти (унаслідок змін у законодавстві, в організаційній структурі об’єкта тощо). Це стосується всіх складових проекту: функціональної, інформаційної моделей, інтерфейсу користувача й документації. Повернення до попередніх стадій. Цей недолік каскадної моделі є наслідком попереднього. Повернення може стати причиною порушення графіку робіт та ускладнити взаємовідносини між розробниками ІС, що працюють над різними етапами тощо. Найгіршим є той факт, що недоліки попереднього етапу можуть проявити себе набагато пізніше. Унаслідок цього роботу требу припиняти і повертатись з поточного до попереднього етапу.

Складність паралельного виконання робіт. За каскадної моделі робота над проектом будується як низка послідовних кроків. Навіть у тому разі, коли розробку окремих частин (підсистем) можна виконувати паралельно, зробити це у рамках каскадної схеми досить важко. Ці складнощі пов’язані з необхідністю постійного узгодження різних частин проекту. Парадокс полягає у тому, що чим більш незалежними є частини, тим ретельніше має виконуватись синхронізація. А внаслідок цього – тим сильніше залежать одна від одної групи розробників.

Інформаційна перенасиченість. Цей недолік є наслідком суттєвої залежності між різними групами розробників. Проблема полягає в тому, що при внесенні змін до будь-якої частини проекту треба повідомити про них всіх розробників, що пов’язані з цією частиною. Як наслідок – швидко зростають витрати на документування змін, опрацювання цих змін, погіршується оперативність узгоджень й т. і. А в цілому, – збільшується обсяг інформації, що є наслідком особливостей схеми управління проектом, що має в певному розумінні «паразитну» природу. Питання перенасиченості різко загостюється за умов ротації груп розробників. Адже нові спеціалісти крім вивчення нового матеріалу повинні опанувати велику кількість старої інформації, що пов’язана з

історією змін, коригувань та узгоджень. Це факт вкрай негативно впливає на ефективність проектування.

Складність управління проектом під час використання каскадної схеми здебільшого обумовлена строгою послідовністю стадій розроблення й наявністю складних взаємозв'язків між різними частинами проекту. Послідовність розроблення проекту призводить до того, що одні групи розробників повинні очікувати результати роботи інших команд. Отже, потрібно адміністративне втручання для узгодження термінів роботи та складу переданої документації. Значний рівень ризику. Чим складнішим є проект, тим довшою буде тривалість кожного з його етапів, тим складнішими будуть взаємозв'язки між частинами проекту. Внаслідок специфіки каскадної моделі така ситуація може призвести до великої кількості повернень до попередніх етапів після виявлення змін та їх погоджень. Нарешті, Фактично це означає, що існує ймовірність значно збільшити терміни виконання проекту, а таке збільшення (з фінансової точки зору) означає високий рівень ризику інвестицій). Нарешті, занадто велика кількість змін (особливо в в предметній області або у вимогах замовника) можуть взагалі призупинити проект, не довівши його до остаточної реалізації. Тому можна стверджувати, що складні проекти, що розробляються за каскадною схемою, мають підвищений рівень ризику. Область застосування. Каскадний підхід добре зарекомендував себе при розробці:

1) однорідних ІС, де кожен додаток становить єдине ціле;

2) ІС, для яких на самому початку розроблення можна досить точно й повно сформулювати всі вимоги, щоб надати розробникам свободу реалізувати їх якнайкраще з технічного погляду. До цієї категорії потрапляють ІС зі складними обчисленнями, системи, що працюють у реальному часі тощо.

На базі каскадної моделі життєвого циклу побудовано один із найвідоміших засобів автоматизації процесів розроблення складних інформаційних систем від компанії Oracle, що має назву Oracle Designer.

Методика (в літературі вона має назву CRM), що використовується під час роботи цього комплексу, базується на таких положеннях:

1) проектування має структурне походження, весь процес розроблення системи подається у вигляді послідовності чітко визначених етапів;

2) підтримка здійснюється на всіх етапах життєвого циклу системи, починаючи від загальних пропозицій і закінчуючи супроводом готового продукту;

3) перевага віддається архітектурі «клієнт-сервер», зокрема складним структурам розподілених баз даних;

4) під час розроблення всі специфікації проекту зберігаються в спеціальній базі даних (репозиторії), який працює під управлінням СУБД Oracle. До репозиторію може підключатись велика кількість користувачів (розробників), унаслідок чого їхні дії є узгодженими;

5) послідовний перехід від одного етапу до іншого автоматизований унаслідок використання спеціальних утиліт. За допомогою них за

специфікаціями на концептуальній стадії можна отримати початковий варіант специфікації рівня проектування. Надалі генерація доповнень значно спрощується;
6) усі етапи проектування та розробки автоматизовані; у будь-який момент може бути згенерований довільний обсяг звітів, які забезпечують документування поточної версії системи на всіх етапах її розробки.

Oracle пропонує свій варіант структури життєвого циклу інформаційної системи, а методологія та програмні засоби щодо її реалізації орієнтовані переважно на продукти цієї компанії. Незважаючи на певні недоліки та обмеження, технологія CDM лишається одною з провідних під час розробки складних інформаційних систем. Докладніше до особливостей CDM ми повернемось у розділі 3, коли будемо розглядати головні методики й технології проектування.

Парадигмі проектування інформаційних систем

Існують дві головні парадигми проектування, що реалізують два різних підходи до опису систем [1,2,4]:
1) структурна (процесно-орієнтована);
2) об’єктно-орієнтована.
Перша концепція має в основі на каскадну модель ЖЦ інформаційної системи. Друга концепція заснована на ітеративній моделі ЖЦ. Розглянемо головні відмінності та особливості використання цих підходів.

Структурний аналіз
Структурний аналіз (від англ. Structured Analysis або SA) і структурне проектування (або SD, від англ. Structured Design) бере свій початок від структурного програмування і класичного системного аналізу [5]. Сутність структурного підходу до розробки ІС полягає в її декомпозиції (розбитті) на функції, що підлягають подальшої автоматизації, а саме: система розбивається на функціональні підсистеми, ті зі свого боку розподіляються на підфункції, завдання тощо. Процес розбиття триває до рівня деталізації процедур. До того ж автоматизована система зберігає цілісне уявлення, коли всі складові компоненти взаємопов’язані та мають загальне підґрунтя. Під час розробки системи у напряму «знизу-догори», тобто від окремих завдань до всієї системи цілісність втрачається, виникають проблеми у разі інформаційного стикування окремих компонентів. Усі найпоширеніші методології структурного підходу [6] базуються на певних загальних принципах [7], головними з яких є такі:

– принцип «розділяй і володій» –підхід до вирішення складнихпроблем шляхом їхнього розбиття на низку менших та незалежних завдань, шо є легкими для сприйняття та практичної реалізації;

– принцип ієрархічного упорядкування – принцип організації складових частин проблеми та упорядкування їх у вигляді ієрархічної деревоподібної структури з додаванням нових деталей на кожному рівні.

Виділення цих двох базових принципів не означає, що інші принципи є другорядними, оскільки ігнорування кожного з них може призвести до непередбачуваних наслідків, зокрема до провалу проекту загалом. Головними з додаткових принципів є такі:

– принцип абстрагування, що полягає у виділенні ключових аспектів системи і абстрагування від несуттєвих;

– принцип формалізації, що полягає методичного підходу до вирішення проблеми;

– принцип несуперечності, узгодженості елементів;

що полягає в необхідності в чіткого обґрунтованостій

– принцип структурування даних, він полягає у тому, що всі дані мають бути структуровані та ієрархічно впорядковані.

Методології структурного аналізу використовують каскадну модель життєвого циклу інформаційної системи.

У структурному аналізі використовуються дві головні групи засобів, що ілюструють функції системи та співвідношення між даними. Кожній групі засобів відповідають певні моделі (діаграми). Найпоширенішими серед них є такі:

– SADT (Structured Analysis and Design Technique), – моделі й відповідні функціональні діаграми;

– DFD (Data Flow Diagrams) – діаграми потоків даних;

– ERD (Entity-Relationship Diagrams) – діаграми «сутність-зв’язок».

На стадії проектування ІС моделі розширюються, уточнюються і доповнюються діаграмами, що відображають структуру програмного забезпечення: архітектуру ПЗ, структурні схеми програм і діаграми екранних форм.

Вказані моделі в сукупності дають повне описання ІС незалежно від того, чи вона є існуючою чи тільки розробляється. Склад діаграм у кожному конкретному випадку залежить від рівня повноти описання системи. У подальших розділам ми повернемось до особливостей складання та використання діаграм DFD та ERD. Зараз розглянемо перший тип моделей, що є типовим для структурного підходу до проектування систем. Мова буде йти про методології структурного аналізу SADT (від англ. Structured Analysis and Design Technique) та SSADM. Вони базуються на структурному аналізі систем та графічному поданні інформації у вигляді системи функцій, що мають три класи структурних моделей:

– функціональна модель;

– інформаційна модель;

– динамічна модель.

Процес моделювання за методологією SADT складається з таких етапів:

– збір інформації та аналіз інформації щодо предметної області;

– документування отриманої інформації;

– моделювання в термінах IDEF0;

– коригування моделі у процесі ітеративного рецензування.

На сьогодні ця методологія (більш відома як IDEF0, [5]) використовує формалізований процес моделювання ІС та передбачає такі стадії:

– аналіз;
– проектування;
– реалізація;
– об’єднання;
– тестування;
– установка;
– функціонування.

Проектування інформаційних систем за стандартом IDEF0 полягає у послідовній декомпозиції основних функцій організації на окремі бізнес-процеси, роботи або дії з подальшим об’єднанням їх у єдину ієрархічну модель організації. Декомпозицію можна робити багаторазово, аж до чіткого й вичерпного опису всіх процесів.

Діаграми IDEF0 верхнього рівня прийнято називати батьківськими, нижнього рівня – дочірніми. Процес, що аналізується, виглядає як прямокутник. Зліва знаходяться вхідні дані, праворуч – вихідні, зверху – керуючі або регламентуючі дії, знизу – об’єкти управління. У діаграмі IDEF0 спочатку описують усі зовнішні зв’язки процесу. Після цього роблять декомпозицію цього процесу й описують внутрішні підпроцеси з позначенням усіх зв’язків. До того ж раніше позначені стрілочками зовнішні зв’язки не повинні загубитися. Вони переносяться до діаграми декомпозиції у відповідні підпроцеси. Далі можна зробити подальшу декомпозицію кожного підпроцесу, після чого докладно описати усі зв’язки та функції. Головною перевагою цієї методології є простота й наочність. За недолік можна вказати неможливість описати реакцію процесу на мінливі зовнішні фактори. Для вирішення таких завдань є інші методології.

Методи структурного аналізу удосконалювалися та використовувалися впродовж багатьох років. З часом з’явилися та набули популярності об’єктно-орієнтовані мови програмування. Завдяки цій популярності було розроблено методологію, що має допомагати програмісту розробляти додатки з використанням об’єктно-орієнтованих мов. Ця методологія стала відома як об’єктно-орієнтований аналіз та проектування (від англ. Object-oriented analysis and design або OOAD).

Об’єктно-орієнтований аналіз та програмні засоби щодо його реалізації Об’єктно-орієнтований аналіз або OOAD – це підхід до інженерії програмного забезпечення, що розглядає систему як групу взаємодіючих об’єктів. Об’єктно-орієнтований аналіз (Object-oriented analysis, OOA) використовує методи об’єктного моделювання для аналізу функціональних вимог до системи.

Об’єктно-орієнтоване проектування, ООП (Object-oriented design, OOD) має за мету розробити аналітичні моделі процесів для подальшогостворення специфікацій для їхньої реалізації. Однією з найважливіших серед таких специфікацій є технічне завдання (ТЗ). Концептуальною основою ООП єоб’єктна модель, яка будується з урахуванням принципів абстрагування,інкапсуляції, модульності, ієрархії, типізації, паралелізму, стійкості. Головними поняттями об’єктно-орієнтованого підходу є об’єкт і клас. Об’єкт – це визначена сутність, що відповідає певному предмету або явищу й характеризується класом, станом і поведінкою. Для цих взаємодіючих між собою об’єктів можна створити різні моделі,

що характеризують їхню статичну структуру або динамічну поведінку й розгортання в реальній роботі (run-time deployment). Клас – це множина об'єктів, що пов'язані спільною структурою та поведінкою. Наступну групу важливих понять об'єктного підходу складають поліморфізм (здатність класу належати більш ніж одному типу) та успадкування (побудова нових класів на основі існуючих із можливістю додавання або коригування даних та методів). У сучасній практиці використовують велику кількість об'єктно- орієнтованих методів проектування. Один із найпотужніших серед них – це IDEF4 (від англ. Object-Oriented Design). Цей метод становить методологію ООП, що дозволяє відображати структуру об'єктів і принципи їхньої взаємодії у контексті різних нотацій і об'єктних моделей. Для опису об'єктно-орієнтованих моделей використовують спеціальні мови. Одними з найпопулярніших мов об'єктно-орієнтованого моделювання є UML (від англ. The Unified Modeling Language) та SysML. Уніфікована мова моделювання UML – це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи (зокрема UML-моделі). Він був створений для визначення, візуалізації, проектування та документування переважно програмних додатків. Методи описання результатів аналізу та проектування UML семантично близькі до методів програмування на сучасних об'єктно-орієнтованих мовах. На підставі UML-моделей можлива генерація програмного коду. Крім того, на UML можна розробити докладний план системи, що містить системні функції і бізнес-процеси, схеми баз даних, програмні компоненти багаторазового використання тощо. Але головними перевагами використання UML є те, що:

1) UML дає змогу розглянути систему з усіх поглядів, що стосуються її розроблення й подальшого розгортання.

2) UML забезпечує підтримку всіх етапів життєвого циклу ІС і пропонує для цього відповідний набір діаграм: структурні (Structure Diagrams); діаграми поведінки (Behavior Diagrams); діаграми взаємодії (Interaction Diagrams).

UML фокусується на трьох архітектурних інтерпретаціях ІС:

– функціональній – описує зовнішню поведінку системи з погляду користувача за допомогою use-case діаграм;

– статичній, – представляє систему в термінах атрибутів, методів, зв'язків, класів, повідомлень за допомогою CRC (cards, class diagrams, object diagrams);

– динамічній – орієнтована на опис сервісних архітектур (SOAs) у вигляді діаграм послідовностей (sequence diagrams), схем співпраці (collaboration diagrams), діаграм стану (statecharts) тощо.

Формальна специфікація останньої версії UML була опублікована в серпні 2005 року. Протягом останнього часу семантика мови істотно змінювалась, удосконалювалась та була розширена до рівня підтримки методології Model Driven Development або MDD. Остання версія UML 2.4.1 була прийнята як міжнародний стандарт ISO/IEC 19505-1, 19505-2.

Проблеми вибору

Доступність різних технологій робить актуальною проблему вибору певного рішення щодо методології проектування системи. Поява нових, більш розвинутих

об’єктно-орієнтованих мов мала б (на перший погляд) стимулювати потребу у використанні об’єктного підходу для розроблення бізнес-додатків. Однак на практиці того не сталося. У багатьох випадках використання об’єктно-орієнтованого програмування не є гарантованою перевагою перед традиційною структурною методологією будування системи. У будь-якому разі вибір рішення залежить від специфіки проекту й особливостей тої чи іншої технології. Нижче вказані головні переваги й недоліки структурної та об’єктно-орієнтованої методологій, які можуть вплинути на вибір під час проектування певної системи.

Стандарти проектування інформаційних систем. Методологія CDM

Стандарти та методики

Важливою умовою ефективного використання інформаційних технологій є впровадження корпоративних стандартів. Такі стандарти декларують угоду щодо єдиних правил організації технології та управління під час реалізації проекту. У цьому разі за основу корпоративних стандартів можуть прийматися галузеві, національні й навіть міжнародні стандарти. Останні можна умовно поділити на декілька груп [1, 2]. За предметом стандартизації. До цієї групи можна віднести функціональні стандарти (на мови програмування, інтерфейси, протоколи) та стандарти щодо організації ЖЦ створення й використання ІС і ПЗ. За організацією-розробником (затверджувачем). Серед них можна виділити офіційні міжнародні, офіційні національні або відомчі національні стандарти (наприклад ГОСТ, ANSI, IDEFO/1), стандарти міжнародних консорціумів та комітетів зі стандартизації тощо. За методичним джерелом. Сюди належать різного роду методичні матеріали провідних фірм-розробників програмного забезпечення, фірм-консультантів, наукових центрів, консорціумів зі стандартизації тощо. Представники з будь-якої групи можуть бути прийнятими за основу для розроблення внутрішніх стандартів. Тут принциповим є вимога до кінцевого результату, а саме: правильно побудовані корпоративні стандарти утворюють цілісну систему, яка включає три види стандартів:

– на продукти й послуги;
– на процеси та технології;
– на форми колективної діяльності, або управлінські стандарти .

Наявність та впровадження стандартів є необхідною умовою для забезпечення якісного проектування та реалізації системи. Однак цілком зрозуміло, що самих стандартів для вирішення цього питання недостатньо. Для їхньої належної підтримки на рівні розробки необхідна певна методика. Однією з таких методик є CDM (від англ. Custom Development Method), запропонована компанією Oracle. Безумовною перевагою цієї методики є той факт, що вона орієнтована на розробку прикладних ІС відповідно до Міжнародного стандарту ISO/IEC 12207: 1995-08-01 01 та підтримує всі фази життєвого циклу відповідно до концепції вказаного стандарту. Методика CDM фірми Oracle Компанія Oracle добре відома на ринку програмного забезпечення як лідер із розроблення потужних СУБД та відповідних

інструментів щодо проектування баз даних. Однак цим сфера інтересів Oracle не обмежується, оскільки одним із важливих напрямів діяльності фірми є розроблення методологічних основ та інструментальних засобів для автоматизації процесів проектування та реалізації складних інформаційних систем, що орієнтовані на активне використання баз даних. Методика CDM є розвитком давно розробленої методики CASE-Method фірми Oracle, яку реалізовано у CASE-засобі Oracle Designer/2000 [1, 2]. Розглянемо головні складові CASE-технології та інструментального середовища від фірми Oracle.

1) Проектування має структурне значення, весь процес розроблення системи має вигляд послідовності чітко визначених етапів.

2) Підтримка здійснюється на всіх етапах ЖЦ системи, починаючи від загальних пропозицій і закінчуючи супроводженням готового продукту.

3) Перевага віддається архітектурі «клієнт-сервер», зокрема складним структурам розподілених баз даних.

4) Під час розроблення всі специфікації проекту зберігаються в спеціальній базі даних (репозиторії). Цей засіб включено до складу ПО Дизайнер і працює під управлінням СУБД Oracle. Репозиторій дає змогу підключатися до нього великому числу користувачів із різними рівнями прав доступу шляхом стандартних засобів СУБД Oracle. Унаслідок цього всі дії розробників стають строго узгодженими та стають неможливими незалежні дії кожного з них.

5) Послідовний перехід від одного етапу до іншого автоматизований через використання спеціальних утиліт. За допомогою них за специфікаціями концептуальної стадії можна отримати початковий варіант специфікації рівня проектування. Надалі генерація доповнень значно спрощується.

6) Стандартні дії етапів проектування та розроблення автоматизовані. У будь-який момент може бути згенерований певний обсяг звітів за вмістом сховища, які забезпечують документування поточної версії системи на всіх етапах її розроблення. Існують спеціальні процедури, що дозволяють здійснити перевірку специфікацій на повноту й несуперечність.

Структура життєвого циклу згідно з методологією CDM

Методика CDM від Oracle пропонує свій варіант структури життєвого циклу інформаційної системи:

1) формування стратегії. Цей етап передбачає моделювання та аналіз процесів, які описують діяльність організації, особливості роботи. Кінцева мета – створення моделей процесів, виявлення можливостей їхнього вдосконалення. Етап не обов’язковий, якщо існуючі технології та організаційні структури чітко визначені, добре вивчені й загалом зрозумілі;

2) аналіз. На цьому етапі відбувається розроблення повних концептуальних моделей, які описують інформаційні потреби організації, особливості функціонування. Унаслідок цього формуються моделі двох типів: інформаційні (вони відображають структуру та загальні закономірності предметної області) і функціональні (описують особливості вирішуваних завдань);

3) проектування. Цей етап передбачає перетворення початкових вимог до системи в докладні специфікації. Спеціальні утиліти, що входять до складу ПО Oracle, значно спрощують цю процедуру;
4) реалізація. На цьому етапі розробляються і тестуються програми, що входять до складу майбутньої ІС. ПО Oracle містить спеціальні генератори додатків, які гранично автоматизують цей етап, зазвичай тривалий і найскладніший. Терміни розробки значно знижуються;
5) впровадження. На цій стадії система встановлюється, підготовляється до початку експлуатації. Відбувається підготовка персоналу;
6) експлуатація. Підтримка системи, планування майбутніх доповнень і змін.
Варто відзначити, що Oracle не розглядає таку стадію існування ІС, як зняття з експлуатації. Тобто розробник, який застосовує методику і ПО Oracle, автоматично «підсаджується» на нього. Інша справа, що зручність засобів розроблення багато в чому виправдовує таку залежність.
Методика CDM виділяє такі процеси, що мають перебіг протягом ЖЦ ІС:

- визначення виробничих вимог;
- дослідження існуючих систем;
- визначення технічної архітектури;
- проектування і побудова бази даних;
- проектування і реалізація модулів;
- конвертування даних;
- документування;
- тестування;
- навчання;
- перехід до нової системи;
- підтримка й супроводження.

Особливості методики CDM

На наш погляд, серед чисельних особливостей методики головними є такі:
1) за ступенем адаптивності методика CDM пропонує три моделі життєвого циклу: класична – передбачає всі етапи, швидке розроблення – з використанням інструментів моделювання і програмування Oracle, полегшений підхід – для малих проектів;
2) методикою не передбачено включення додаткових завдань, які не обумовлені в CDM. Зокрема не передбачена прив'язка їх до решти. Крім того, неможливо видалення завдання і відповідних до нього документів, якщо це не передбачено в жодній із трьох моделей життєвого циклу ІС, неможлива зміна послідовності виконання завдань й т. і.;
3) модель життєвого циклу ІС в Oracle CDM, по суті, є каскадною;
4) методика не є обов'язковою, хоча і є фірмовим стандартом, однак в разі використання ПЗ Oracle інший підхід малоймовірний;
5) методика спирається на використання ПЗ розробника від Oracle, пристосування її до інших засобів важко;

6) методика CDM становить певний матеріал, деталізований до рівня шаблонів проектних документів. Відповідно ці документи розраховані на пряме використання у проектах інформаційних систем, що спираються на інструментальні засоби та СУБД фірми Oracle.

Корпоративні стандарти

Реальне застосування будь-якої технології проектування, розробки й супроводу ІС у певній організації або проекті неможливе без застосування низки стандартів (правил або угод), яких мають дотримуватись усі учасники проекту. Для більшості складних проектів доводиться створювати власні комплекти нормативних і методичних документів, що регламентують процеси, етапи, роботи й документи для певних програмних продуктів. Такі стандарти називаються корпоративними й становлять угоду щодо єдиних правил організації технології або управління у межах організації.

До таких стандартів належать:

– стандарти проектування;
– стандарти на оформлення документації;
– стандарти на інтерфейси користувача.

Стандарт проектування має регламентувати:

– набір моделей (діаграм) на кожному етапі проектування системи та рівень їх деталізації;
– правила фіксації проектних рішень на діаграмах зокрема правила іменування (ідентифікації) об’єктів, набір атрибутів для всіх об’єктів, правила заповнення цих атрибутів на кожній стадії проектування, правила оформлення діаграм, зокрема вимоги до їхньої форми, розмірів, наповнення тощо;
– вимоги до конфігурації робочих місць, налаштування операційної системи, загальні налаштування проекту тощо;
– механізм підтримки спільної роботи над проектом, зокрема: правил інтеграції підсистем проекту, підтримки проекту в актуальному для всіх розробників стані (регламент обміну проектною інформацією, правила та методики синхронізації тощо), правил перевірки проектних рішень на їхню несуперечність тощо.

Стандарт оформлення проектної документації має встановлювати:

– комплектність, склад та структуру документації на кожній стадії проектування;
– вимоги до її оформлення у т. ч зміст розділів, підрозділів, пунктів, таблиць тощо;
– правила підготовки, розгляду, погодження та затвердження документації із вказанням граничних строків виконання кожної стадії;
– вимоги до налаштування технічних засобів підготовки документації;
– вимоги до налаштування CASE-засобів для забезпечення підготовки документації відповідно до встановлених вимог.

Стандарт інтерфейсу користувача має встановлювати:

– правила оформлення екранних форм (шрифти, колір символів, тла), склад і розташування вікон та елементів управління;
– правила використання засобів вводу (клавіатури, миші тощо);
– правила оформлення текстів довідок;
– перелік повідомлень;
– правила обробки (реакцію) на дії користувача.

За основу корпоративних стандартів можуть прийматися галузеві, національні або міжнародні стандарти. До них можуть належати різного роду методичні матеріали провідних фірм-розробників ПЗ, фірм-консультантів, наукових центрів, консорціумів зі стандартизації тощо.

Канонічне проектування інформаційних систем та його нормативне забезпечення

За своєю сутністю канонічне проектування відображає технологію індивідуального проектування ІС. Серед особливостей канонічного підходу можна виділити такі:
– відображення особливостей «ручної» технології проектування;
– орієнтація на індивідуальне проектування;
– акцент на планування та розподілі робіт на рівні виконавців;
– можливість інтеграції виконання елементарних операцій;
– орієнтація на порівняно невеликі інформаційні проекти;
– використання універсальних інструментальних засобів комп'ютерної підтримки.

Канонічне проектування спрямоване на мінімальне використання типових проектних рішень. Адаптація проектних рішень у канонічному проектуванні здійснюється виключно шляхом перепрограмування відповідних програмних модулів.

Організація канонічного проектування ІС заснована на використанні каскадної моделі життєвого циклу й передбачає низку стадій та етапів. Принцип розподілу процесу проектування на стадії та етапи спрямований на проектування системи «зверху-вниз», тобто на послідовну розробку спочатку узагальнених, а потім деталізованих проектних рішень.

Оскільки об'єкти автоматизації за складністю різні, то набір завдань для локальних рішень у межах певної системи, стадії та етапи робіт для її створення можуть суттєво відрізнятись за трудомісткістю. Тому допустимо об'єднувати послідовні етапи, вилучати певні етапи на довільній стадії проекту, а також починати виконання наступної стадії до закінчення попередньої. Усі стадії та етапи створення ІС фіксуються в договорах й технічних завданнях на виконання робіт.

Канонічне проектування регламентує низка стандартів, а саме:
ГОСТ 34.003 – Терміни та визначення головних понять у сфері автоматизованих систем;

ГОСТ 34.201 – Види, комплектність і позначення документів при створенні автоматизованих систем;
ГОСТ 34.601 – Стадії створення автоматизованих систем;
ГОСТ 34.602 – Технічне завдання на створення ІС;
ГОСТ 34.603 – Види випробувань автоматизованих систем;
ГОСТ 2.105 – Загальні вимоги до текстових документів.
Стосовно проекту розробки ІС можна виділити три узагальнені стадії проектування:
– передпроектну;
– проектну;
– післяпроектну.
Розглянемо вказані стадії більш детально.

Стадії та етапи канонічного проектування інформаційних систем

Головні етапи робіт (стадії) відповідно до ГОСТ 34.601 і додаткових пояснень, виглядають так:
1. Формування вимог до інформаційної системи має за мету виконання таких дій:
– обстеження об’єкту та обґрунтування необхідності створення ІС;
– формування вимог користувача до ІС;
– оформлення звіту щодо виконаної роботи й заявки на розроблення ІС.
2. Розроблення концепції ІС.
Ця стадія складається з таких етапів:
– вивчення об’єкта;
– проведення науково-дослідних робіт (за необхідністю);
– розроблення користувачів; варіантів концепції ІС відповідно до вимог
– оформлення звіту щодо виконаної роботи.
3. Технічне завдання
Це дуже відповідальний етап, що має за мету розроблення й затвердження ТЗ на створення інформаційної системи. Підготовчим етапом для формування технічного завдання є техніко-економічне обґрунтування проекту. Це спеціальний документ, що фіксує результати визначення стратегії впровадження ІС. У цьому документі має бути чітко визначено результати виконання проекту для замовника, а також вказані графіки проведення робіт і графік фінансування на різних стадіях виконання проекту. Додатково у такому документі вказують терміни, період окупності, очікувані зиски та економічний ефект від реалізації проекту. Зазвичай, у техніко-економічному обґрунтуванні вказують:
– усі ризики та обмеження, що впливають на успішність проекту;
– умови експлуатації майбутньої системи: архітектурні, програмні, вимоги до апаратних засобів та до компонентів ПО і СУБД;
– користувачів системи (можливо, за категоріями);
– функції, що виконуються системою;
– інтерфейси та розподіл функцій між людиною та системою;
– терміни завершення етапів, форма прийому/передачі робіт;
– горизонт проекту;
– можливості подальшого розвитку системи.

Відповідно до ГОСТ 34.602-89, технічне завдання – це головний документ, що визначає вимоги та порядок створення (розвитку або модернізації) системи, згідно до яких проводиться її розроблення та приймання під час введення до експлуатації.
Розроблення технічного завдання складається із таких розділів:

– загальні відомості;
– призначення та мета створення (розвитку) ІС;
– характеристика об'єктів автоматизації;
– вимоги до ІС;
– склад і зміст робіт щодо створення ІС;
– порядок контролю та приймання ІС;
– вимоги до складу та змісту робіт щодо підготовки та введення ІС до експлуатації;
– вимоги щодо документування;
– джерела розроблення.

Проектна стадія орієнтована переважно на розробку технічного та робочого проектів. Процес розробки технічного завдання передбачає обстеження об'єкта автоматизації (організації або підрозділу) та його діючої системи управління. Для вирішення завдань інформаційного забезпечення необхідно також проаналізувати інформаційні потоки, форми документації, системи кодування, а також усе, що пов'язано зі структурою баз даних та СУБД, що визначає склад вихідних технологічних вимог.

4. Ескізний проект.

Включає такі етапи робіт:

– розробка попередніх проектних рішень щодо ІС та її складових;
– розробка ескізної документації на систему та на її складові.

Якщо для ІС певного об'єкта автоматизації раніше обрані проектні рішення є очевидними, стадія ескізного проекту може бути виключена з послідовності робіт. Тобто ця стадія не є обов'язковою. Крім того, на етапі ескізного проекту мають бути визначені:

– цілі та функції ІС та її підсистем;
– склад комплексів задач і окремих завдань;
– концепція та структура інформаційної бази;
– функції СУБД;
– функції та параметри програмних засобів;
– очікуваний ефект від впровадження системи.

Документація з результатами робіт за сукупністю прийнятих проектних рішень погоджується, затверджується і використовується у подальшому проектування для виконання робіт щодо створення ІС.

На підставі технічного завдання (зокрема за наявністю ескізного проекту) розробляється технічний проект.

5. Технічний проект

Включає такі етапи:

– розробка проектних рішень щодо системи та її складових;

– підготовка документації на систему в цілому та на її окремі складові;
– розробка й оформлення документації на постачання комплектуючих до системи і (або) технічних вимог (завдань) на їх розробку;
– розробка завдань на проектування в суміжних частинах проекту об’єкта автоматизації.

На етапі технічного проекту проводяться роботи науково-дослідного та експериментального характеру для вибору головних проектних рішень, а також розраховується економічна ефективність системи. Важливим аспектом розробки технічного проекту є аналіз всієї інформації, що використовується на предмет таких характеристик, як повнота, відсутність дублювання і надмірності, несуперечливість тощо. Також на цьому етапі визначають форми вихідних документів. Документацію за результатами робіт оформляють відповідно до вимог ГОСТ 34-201.

6. Робоча документація (РД)

Передбачає такі дії:
– розроблення РД на систему й окремі її складові;
– розроблення або адаптацію програм.

Один із головних етапів стадії робочого проектування – розроблення РД на інформаційне забезпечення системи. До складу такої документації входять:
– технічний проект ІС;
– описання баз даних;
– перелік вхідних та вихідних даних і документів.

Стадія технічного проектування завершується підготовкою та оформленням документації на постачання та комплектування ІС, визначенням технічних вимог і складанням ТЗ на розроблення системи.

Стадія «Робоча документація» передбачає створення як програмного продукту, так і повного комплекту супроводжувальної документації. Така документація має надавати всі відомості, що забезпечують виконання робіт на стадіях введення ІС до експлуатації, самої експлуатації ІС, а також відомості щодо підтримки рівня якості системи (дотримання експлуатаційних характеристик). Після проектна стадія має за мету реалізацію заходів щодо впровадження системи, підготовку приміщень і технічних засобів, навчання персоналу. Також проводиться експлуатація системи з вирішенням низки завдань, аналізуються результати випробувань, реалізуються заходи щодо супроводу й доопрацювання тощо.

Стадія 7. Введення до експлуатації складається із таких пунктів:
– підготовка об’єкта автоматизації до введення ІС у дію;
– підготовка персоналу, за необхідністю – його перекваліфікація;
– комплектація ІС необхідними програмними й технічними засобами, інформаційним забезпеченням тощо;
– проведення будівельно-монтажних робіт;
– проведення пусконалагоджувальних робіт;
– проведення попередніх випробувань;
– проведення дослідної експлуатації;
– здійснення приймальних випробувань.

Головними різновидами випробувань для ІС є такі:

– попередні випробування;

– дослідна експлуатація;

– приймальні випробування, вони можуть бути розширені додатковими випробуваннями ІС і її складових частин.

Впродовж попередніх випробувань (за відповідною програмою та методикою), проводять випробування системи на її працездатність і відповідність ТЗ, усувають недоліки та вносять зміни до проектної та супровідної документації. Далі проводять дослідну експлуатацію системи, аналізують її результати, здійснюють доробку програмного забезпечення і додаткове налаштування технічних засобів. На етапі приймальних випробувань увагу концентрують на відповідності ТЗ, аналізують результати комплексних випробувань системи, усувають недоліки, які були виявлені під час цієї роботи. За результатами всіх випробувань оформляють відповідні акти щодо: приймання ІС до дослідної експлуатації, її завершення та приймання системи до постійної експлуатації.

8. Супроводження ІС

Ця стадія передбачає проведення таких дій:

– виконання робіт згідно до гарантійних зобов’язань;

– після гарантійне обслуговування системи.

Головними процесами цієї стадії є здійснення робіт щодо усунення недоліків, виявлених підчас експлуатації системи протягом гарантійного терміну, аналіз роботи системи за реальних умов, виявлення відхилень з’ясування причини їхнього виникнення, усунення причин відхилень і недоліків, забезпечення стабільної експлуатації та характеристик системи.

Додаткова література:

1. Бакка М. Т. Метрологія, стандартизація, сертифікація і акредитація : навч. посібник: в 2-х ч., Ч.2 : Стандартизація, сертифікація і акредитація / – Житомир: ЖІТІ, 2002. – 384с.
2. Тарасова В. В. Метрологія, стандартизація і сертифікація : Підручник / – К. : ЦУЛ, 2006. – 264с.
3. Цюцюра С. В. Метрологія, основи вимірювань, стандартизація та сертифікація : Навч. посібник. – 3-тє вид. / – К. : Знання, 2006. – 242с.
4. Боженко Л. І. Стандартизація, метрологія та кваліметрія у машинобудуванні : Навч. Посібник. / – Л. : Світ, 2003. – 328с.
5. Вендров А.М. Проектирование программного обеспечения экономических информационных систем : учебник /А.М. Вендров. – 2-е изд., перераб. и доп. – М. : Финансы и статистика, 2006. – 544с.
6. Гвоздева В.А.Основы построения автоматизированныхин-формационных систем : учебник / В. А. Гвоздева, Ю.И.Лаврентьева. – М.:ИД"Форум": ИНФРА-М, 2007. – 320с.