УДК 004.056 DOI: https://doi.org/10.33099/2304-2745/2021-1-71/106-115

Беляченко В. В. (0000-0003-3938-5158)
Бобров С. В., канд. техн. наук, доцент (0000-0002-9647-9700)
Утюшев М. К. (0000-0002-7386-7831)
Мулявка А. С. (0000-0002-3113-0719)

Центр воєнно-стратегічних досліджень Національного університету оборони України імені Івана Черняховського

Основні відмінності та особливості структурного та об'єктно-орієнтованого підходів до проєктування інформаційних систем військового призначення

Резюме. У статті розглянуто переваги та недоліки двох підходів щодо проєктування інформаційних систем. Вибір структурного і об'єктно-орієнтованого підходу залежить від переваг обраної методології, стану цільового інформаційного середовища, обсягу та складності системи, яку необхідно проєктувати. Запропоновано алгоритм вибору методології проєктування на етапі концепції на основі семи ключових критеріїв.

Ключові слова: інформаційна система; проєктування інформаційних систем; структурний аналіз; об'єктно-орієнтований аналіз; критерії вибору моделі проєктування.

Постановка проблеми. Закріплення в Конституції України стратегічного курсу на набуття членства України в ЄС та НАТО зумовило зростання необхідності використання інформаційних технологій управлінської діяльності у військовій сфері. Управлінська діяльність у військовій сфері охоплює матеріальний, інформаційний та фінансовий потоки ресурсів і потребує інтелектоємних інструментів високого рівня. Досвід збройних конфліктів за участю збройних сил провідних країн світу (ПКС) свідчить, що без автоматизації процесів управління матеріальними та нематеріальними ресурсами неможливо ефективно забезпечити операції військ (сил). Створення автоматизованих систем управління логістичним забезпеченням як інформаційної системи військового призначення (ІС ВП) були першочерговими завданнями комплексної інформатизації процесів управління оборонними ресурсами та військами в США й інших країнах – членах НАТО. Ускладнення під час реалізації проєктів створення ІС ВП у ПКС загострили увагу до низки проблемних питань, зокрема, проблеми вибору методології проєктування, які зменшували б складність проєктних завдань і мінімізували проєктні ризики.

Аналіз останніх досліджень та публікацій. Вимоги до сучасних ІС ВП і їх складність постійно зростає, відповідно, загострюється проблема ефективного управління процесами створення, тестування та впровадження ІС ВП в інформаційне середовище військового відомства Комплексне вирішення цієї проблеми – складна і довготривала робота, яка потребує високої кваліфікації задіяних у ній спеціалістів. Попри технологічний прогрес, завдання у сфері проєктування ІС ВП виконуються на інтуїтивному рівні згідно з досвідом і прогностичними здібностями учасників проєкту і обґрунтовуються суб'єктивними оцінками з ознаками конфлікту інтересів. Експериментальна перевірка проєктних рішень на наступних стадіях розроблення може виявити помилку вибору методології і технології проєктування як один з наслідків недостатньо ретельного підходу до збору вимог до майбутньої ІС ВП і точності опису процесів. Досвід виконання ДКР "Ніка-1" та ін. ІС ВП в Збройних Силах України, а також досвід розроблення ІС ВП у Збройних силах США та інших ПКС, свідчить, що вартість виправлення помилок, які було зроблено на ранніх стадіях проєкту, з часом значно зростає [1, 3–5].

Зокрема, під час впровадження Системи Підтримки Бою в Експедиції (англ. – ECSS) у Військово-повітряних силах США спостерігались значні відхилення від попередніх оцінок обсягів проєктних ресурсів і часу реалізації [1], що разом з іншими причинами підтверджує необхідність критичного розгляду методологій проєктування ІС ВП для обґрунтованого вибору з урахуванням комплексної інформатизації та реінжинірингу процесів управління потоками ресурсів. У сфері економіки, згідно з щорічним звітом Інституту проєктного менеджменту (англ. – PMI) за 2016 р. про результати опитування учасників проєктів з усіх галузей економіки (сфера IT становила 17 % від загального числа опитаних учасників проєктів), неточність збору вимог і первинних даних, на підставі яких серед інших проєктних стратегій обирається методологія і технологія проєктування ІС, посідає третє місце серед причин невдалих проєктів (у

37 % проєктів респонденти називали ці причини серед інших трьох основних причин) [2, 8]. Виправлення помилок структурного та/або об'єктно-орієнтованого аналізу, виявлених на стадії робочого проєктування, збільшують вартість і тривалість проєкту. Виявлення та усунення помилок на етапі тестування ще дорожче і довше. Усунення помилок на етапі експлуатації може перевищувати вартість усього проєкту та в деяких випадках призводити до відмови від розроблення загалом і дискредитації як розробників, так і замовників.

Вибір методології проєктування, засобів супроводження ІС, тестування її компонентів та інкрементів на всіх стадіях розроблення є складними та відповідальними *управлінськими рішеннями*, від яких залежить результативність та ефективність проєкту загалом. Порядок розв'язання проблеми вибору з погляду теорії прийняття рішення розглянуто в роботі [3]. Під час пошуку рішення визначаються критерії, які впливають на це рішення, їх вагомість і порядок розгляду, збираються та аналізуються дані щодо альтернатив, визначаються і оцінюються ризики та обмеження щодо об'єкта. Відомості про переваги та недоліки структурного та об'єктно-орієнтованого підходу, які придатні на роль чинників впливу на управлінські рішення були описані дослідниками раніше та зафіксовані в багатьох джерелах 90-х – початку 2000-х, зокрема [4–10]. Проте дані щодо новітнього проєктного досвіду [2] свідчать, що на етапах розробляння, тестування програмних рішень продовжують виявлятись ускладнення, зумовлені неточністю збору вимог до проєкту, поспішністю висновків щодо первинних даних, недбалістю під час моделювання тощо.

Структурний і об'єктно-орієнтований підходи до проєктування системи, результатом яких є сукупність моделей об'єкта інформатизації попри особливості, переваги і недоліки, описані в роботах [4–6, 12–14], мають спільне – об'єкт інформатизації, що дає змогу зробити припущення про можливість успішного комбінування обох підходів у межах ширшої концепції – архітектурного підходу [12]. Комбінація підходів реалізована в методології моделювання діяльності організації ARIS (англ. Architecture of Integrated Information Systems), яка об'єднує структурні та об'єктно-орієнтовані методи для формування цілісного погляду на процеси, об'єкти та потоки інформації в організації. Близькість методів моделювання предметної сфери в структурному та об'єктно-орієнтованому підходах проілюстрована в роботі [11], у якій обґрунтовується раціональність їх поєднання. Результатом проєктування на етапі концепції є "моделі, які дають змогу зрозуміти структуру майбутньої ІС, досягти узгодження суперечливих вимог та окреслити план реалізації" [9]. Висновки дослідників щодо вибору методології проєктування, або їх комбінації, сходяться на тому, що кінцевий вибір архітектора системи значною мірою зумовлений критеріями, які безпосередньо залежать від такого: що (процеси та об'єкти); у якому обсязі (вимоги) передбачається автоматизувати; що відомо про існуюче інформаційне середовище (початкові дані); чи існує типове проєктне рішення; яке передбачено ресурсне забезпечення (час, фінанси, кадровий потенціал); які є проєктні обмеження; у якому організаційному та інформаційному середовищі перебуватиме розроблення на кожному етапі ЖЦ системи та які проєктні ризики потребують обробляння.

Проте більшість проєктів усе ж виконуються з використанням одного з підходів, що дає змогу авторам статті зробити припущення про можливість швидко обрати методологію проєктування завдяки використанню декількох ключових критеріїв. Швидкий вибір дасть змогу зосередитись на зборі загальних вимог до системи та аналізі первинних даних, згідно з якими архітектор системи обирає остаточне рішення щодо організаційних і технологічних форм, у яких створюватиметься АС, або її компоненти, та який підхід застосувати для опису і проєктування системи – структурний чи об'єктно-орієнтований, або комбінацію підходів, з урахуванням складності поєднання, якщо є необхідність узгодження розроблюваної системи з ієрархією вже існуючих систем та актуальним інформаційним середовищем.

Метою статті є огляд підходів щодо проєктування ІС, систематизація особливостей та обґрунтування критеріїв, на які розробники ІС ВП можуть спиратись під час первинного вибору методології та технології проєктування.

Виклад основного матеріалу. *Проєктування* – це процес пошуку, досліджень та реалізації способів створення системи, що задовольняє вимогам до її функціональності з огляду на обмеження часу, коштів і доступність технології та охоплює три сфери:

проєктування об'єктів даних, які будуть реалізовані в базі даних;

проєктування програм, екранних форм, звітів, які забезпечуватимуть виконання запитів до даних;

проєктування певного середовища, топології мережі, конфігурації апаратних засобів, використовуваної архітектури (файл-серверної або клієнт-серверної), способів обробки даних (паралельна, розподілена).

Мета проєктування полягає у забезпеченні надійного та ефективного функціонування ІС ВП та конструктивної взаємодії замовника (користувачів) і розробників ІС ВП. У процесі проєктування ІС ВП удосконалюється як інструмент реінжинірингу управлінських процесів органів військового управління.

Основу проєкту будь-якої ІС складають такі компоненти [3–5, 12]:

методологія проєктування;

технологія проєктування;

стандарти і методики проєктування;

інструментальні CASE-засоби проєктування як окремих елементів життєвого циклу програмного засобу (надалі ЖЦ ПЗ), так і інтегровані набори інструментів, які підтримують ЖЦ ПЗ, у яких реалізована підтримка процесів створення ІС.

Між компонентами діє щільний зв'язок, а саме: реалізація методології здійснюється через конкретні технології, кожна технологія підтримується відповідними стандартами і методиками, а інструментальні засоби забезпечують реалізацію процесів проєктування, що описані у методиках і стандартах. *Методологія проєктування* – це знання про організацію процесу розроблення системи, методи управління цим процесом, які гарантують дотримання вимог як до самої системи, так і до характеристик процесу її розроблення і засоби проєктування. Найважливіші завдання, які мають виконуватись учасниками проєкту під час створення ІС:

відповідність розроблюваної ІС меті та завданням організації, а також вимогам щодо обсягів автоматизації її функціональних процесів;

гарантоване створення системи з обмеженням часу та бюджету;

підтримання порядку супроводження, модифікацій і масштабування системи для забезпечення її адаптації до зміни умов;

забезпечення відповідності ІС вимогам до відкритості та здатності до масштабування;

забезпечення можливості спадкоємності розробки через використання елементів існуючого інформаційного середовища організації.

Головний зміст *технології проєктування* – це технологічні інструкції, які є сукупністю з трьох складових:

послідовності виконання технологічних операцій, передбачених процесом проєктування;

критеріїв щодо оцінювання результатів виконання технологічних операцій;

нотацій (тобто графічних і текстових засобів) для опису системи, що проєктується.

Кожна технологічна операція потребує матеріальних, інформаційних і людських ресурсів, зокрема:

даних, отриманих від попередньої операції;

методичних матеріалів, інструкцій, нормативів, стандартів;

програмних і технічних засобів;

виконавців.

Результати здійснення операції мають бути подані у стандартному вигляді, що гарантує його адекватне сприйняття під час виконання наступної технологічної операції для якої вони є вхідними даними.

Низка *загальних вимог до технології проєктування*, розроблення і супроводження ІС є такою:

підтримувати повний ЖЦ ІС;

гарантувати досягнення цілей розроблення із заданою якістю та у межах встановленого часу;

забезпечувати можливість розподілу (декомпозиції) великої ІС на кілька складових частин (підсистем), які розробляються окремими групами виконавців із подальшою інтеграцією цих частин;

надавати можливість проєктування підсистем малими групами, що гарантує керованість колективом та підвищує продуктивність унаслідок мінімізації кількості зв'язків із зовнішнім середовищем;

забезпечувати мінімальний час для отримання працездатної системи;

давати можливість керувати конфігурацією проєкту, вести облік і синхронізацію версій проєкту та його складових;

забезпечувати незалежність проєктних рішень від засобів реалізації, – систем управління базами даних, операційної системи, мов програмування тощо.

Існують дві основні парадигми проєктування, які реалізують два різних підходи до опису систем [3–5]:

структурна (процесно-орієнтована), яка спирається на каскадну модель ЖЦ ІС;

об'єктно-орієнтована, яка спирається на ітеративну модель ЖЦ ІС.

Основні відмінності та особливості використання цих підходів у такому [6]:

1. Структурний аналіз (англ. – Structured Analysis або SA) і структурне проєктування (англ. – Structured Design або SD) бере свій початок від класичного системного аналізу [7]. Сутність структурного підходу до початку розроблення ІС полягає в декомпозиції опису системи за допомогою ієрархічного набору діаграм та виокремлення функцій, які підлягають автоматизації. Система розкривається до функціональних підсистем, які зі свого боку розподіляються на окремі завдання. Процес деталізації триває до рівня окремих процедур, у яких ще є вхід і вихід. Спроба подальшої деталізації призводить до втрати цілісності та пов'язаності з іншими елементами системи. Під час виконання структурного аналізу система має зберігати цілісність, взаємопов'язаність складових компонентів і загальне підґрунтя. Якщо розробляти систему у напряму "знизу-догори", тобто від окремих завдань до всієї системи, то втрачається цілісність і виникають проблеми взаємодії окремих компонентів. Усі найпоширеніші методології структурного підходу [8] базуються на шести загальних принципах [9], два з яких є основними:

принцип "розділяй і володій" – декомпозиційний підхід до розв'язання складних проблем завдяки їх розкладанню на низку простіших і відносно незалежних один від одного завдань, придатних для аналізу та удосконалення під час практичної реалізації;

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

Додаткові принципи:

принцип абстрагування – виділення суттєвих аспектів системи серед несуттєвих;

принцип формалізації – необхідність чіткого методичного підходу до аналізу;

принцип несуперечності – обґрунтованість та узгодженість елементів;

принцип структурування даних – усі дані мають бути структуровані та ієрархічно впорядковані.

У структурному аналізі використовуються дві головні групи методологій, які застосовуються для опису системи:

1) функцій, які виконує система (SADT-моделювання);

2) відношень між даними, яким відповідають функціональні моделі та моделі потоків даних (SSADM-моделювання).

На етапі проєктування системи функціональні моделі та моделі потоків даних доповнюються діаграмами, що відображають архітектуру системи та ПЗ. Склад діаграм у моделях залежить від необхідного рівня повноти опису системи. Найпоширенішими засобами структурного аналізу є:

SADT (від англ. Structured Analysis and Design Technique) – функціональні моделі та відповідні функціональні діаграми;

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

ERD (англ. – Entity Relationship Diagrams) – діаграми "сутність-зв'язок";

EED (англ. – Entity Event Diagrams) – діаграми "сутність-подія".

SADT-моделі є типовими для структурного підходу під час проєктування систем.

Процес моделювання за методологією SADT складається з етапів:

збір і аналіз інформації щодо предметної області;

документування отриманої інформації;

коригування моделей за допомогою ітеративного рецензування.

На сьогодні методологія SADT використовує формалізовані методи моделювання систем (найчастіше в нотаціях IDEF0 та IDEF3 [7]), які дають змогу отримати опис ІС в обсязі, необхідному для аналізу вимог до системи та ПЗ та переходу до концептуального проєктування.

Методологія SSADM (англ. Structured Systems Analysis and Design Method) також базується на структурному аналізі систем і графічному поданні інформації у вигляді комплекту структурних моделей: логічної структури даних (DFD високого рівня + ERD), деталізованої структури інформаційних потоків (DFD), змін структур у динаміці (EED) – змін життєвого циклу

сутностей. Процес моделювання за методологією SSADM аналогічний як SADT. Побудова цих моделей дає змогу перейти до процесу проєктування архітектури системи та програмного засобу під час створення ІС.

Методологія структурного підходу використовуються в каскадній моделі ЖЦ ІС, результативність якої була перевірена на практиці:

однорідних ІС, де кожен додаток становить єдине ціле;

ІС, для яких на початку розроблення точно сформульовані вимоги, що сприяє пошуку розробниками оптимальних програмно-технічних рішень для ІС зі складними обчисленнями та необхідністю роботи в режимі реального часу.

На базі каскадної моделі було створено популярний в кінці 90-х – початку 2000-х програмний комплекс розробника Oracle Designer [3].

Нині існує достатня кількість інструментальних засобів проєктування, що входять до складу програмних засобів, які підтримують повний життєвий цикл програмних засобів та ІС, створених з їх використанням. Наприклад, клієнти SAP безоплатно отримують комплексне програмне рішення SAP Solution Manager 7.2 з управління життєвим циклом усіх IT-компонентів організації в мультисистемному інформаційному середовищі і підтримкою моделей проєктування від каскадної, V-подібної, спіральної до інкрементно-ітеративної та інших різновидів "гнучких" (agile).

Основні переваги структурного підходу до проєктування [3, 4, 7, 9]:

засновані на класичному системному аналізі, є процесно-орієнтованими, підходять для описання будь-яких систем, зручні для дослідження предметної області, реінжинірингу бізнес-процесів тощо;

структурні методології дають зрозумілу і просту візуалізацію системи, їх легко сприймають і користувачі, і розробники проєкту;

у структурних методологіях акцент зроблено на командній роботі;

чітко визначені етапи, які полегшують управління проєктом і дають змогу розробляти системи високої якості;

структурні методології допускають використання засобів перевірки вимог;

SADT і SSADM – класичні, поширені методології в області проєктування ІС, існують упродовж тривалого часу і добре перевірені на практиці.

Основні недоліки структурного підходу [3–5]:

оскільки структурна методологія є процесно-орієнтованою, то вона ігнорує нефункціональні вимоги до системи, тобто результатом її використання можуть бути не ідеальні рішення з точки зору використання об'єктно-орієнтованих мов програмування на етапі реалізації проєкту (методологія краще відповідає структурним мовам);

оскільки SSAD і SSADM не є ітеративними, то зміна вимог може призвести до перезавантаження практично всього процесу розроблення;

під час використання структурного підходу можуть виникати труднощі у визначенні глибини декомпозиції, тобто моменту, коли доцільно зупинитися та перейти до практичної реалізації моделей.

Об'єктно-орієнтований підхід до проєктування (від англ. Object-Oriented Analysis and Design, OOAD) – це підхід до інженерії ПЗ, що розглядає систему як групу взаємодіючих об'єктів. Об'єктно-орієнтований аналіз ООА (від англ. Object-oriented Analysis, OOA) використовує методи об'єктного моделювання для аналізу функціональних вимог до системи. Об'єктно-орієнтоване проєктування (ООП) (від англ. Object-oriented Design, OOD) має на меті розробити моделі процесів, придатні для аналізу та подальшого створення специфікацій для їхньої реалізації.

Одним з найважливіших результатів специфікацій є технічне завдання. Концептуальною основою ООП є об'єктна модель, яка будується з урахуванням принципів абстрагування, інкапсуляції, модульності, ієрархії, типізації, паралелізму, стійкості. Основними поняттями ООП є клас і об'єкт. Для взаємодіючих між собою об'єктів можна створити моделі, які характеризують їхню статичну структуру або поведінку в динаміці під час розгортання системи (run-time deployment).

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 і методів).

У сучасній практиці використовують велику кількість методів моделювання під час ООП, зокрема **IDEF4**. Цей метод дає змогу 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.

Найпопулярнішою мовою ООП є UML (англ. - the Unified Modeling Language). UML – це відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи та був створений для визначення вимог до системи, візуалізації, проєктування та документування програмних рішень. Методи описання результатів аналізу та проєктування UML семантично близькі до методів програмування на сучасних об'єктно-орієнтованих мовах. За допомогою CASE-засобів на підставі UML-моделей реалізована генерація програмного коду. На UML можна розробити докладний план системи, що містить системні функції і бізнес-процеси, схеми баз даних, програмні компоненти багаторазового використання тощо [3, 8].

Основні переваги об'єктно-орієнтованих методологій. Більшість цих переваг є наслідком об'єктно-орієнтованого програмування, а саме [3–5, 13, 14]:

мультимодельне представлення складних інформаційних систем дає змогу уніфікувати та інтегрувати знання з різних дисциплін усіх учасників проєкту для адекватного бачення майбутньої системи з максимально можливих точок зору;

під час розроблення об'єктів зі складною взаємодією аналітик акцентує увагу на іншому рівні деталізації, ніж це можливо в структурному підході, а стандартизація об'єктів збільшує рівень розуміння проєкту;

спрощення та прискорення програмної реалізації системи на відміну від структурних методологій завдяки автоматизованим засобам візуального та імітаційного програмування з можливістю прямого і зворотного генерування коду найпоширенішими мовами програмування;

модульність, розширення атрибутів і поведінки об'єктів, можливість повторного використання коду в інших проєктах (як наслідок незалежності об'єктів та їх інкапсуляції), що підвищує продуктивність розроблення коду порівняно з традиційними методами програмування, скорочує вартість і тривалість проєктування, програмування і тестування ІС та сприяє поліпшенню якості подальших проєктів;

відсутність чіткого розподілу між фазами аналізу і розроблення, що забезпечує взаємодію з користувачами до завершення проєкту;

швидке генерування коду вивільняє час і ресурси проєкту на ретельне опрацювання вимог замовника до системи, об'єктно-орієнтований аналіз і проєктування, що зменшує проєктні ризики розробки системи і збільшує цінність впровадження для замовника;

якісніше опрацювання вимог до ІС, моделювання та швидке розроблення загалом вивільняє час і ресурси проєкту на ретельніше тестування розроблюваних ІС, що дуже важливо в гетерогенному середовищі вже існуючого інформаційного середовища;

відсутність жорстких обмежень щодо впровадження системи, що дає змогу реалізувати проєкти створення ІС для функціонування у різних середовищах;

велика кількість програмістів, які використовують об'єктно-орієнтовані мови програмування і, відповідно, суттєвіша підтримка виробників ПЗ і розробників мов програмування, які надають великі бібліотеки перевірених практикою об'єктів з коректними кодами, що зменшує ризики під час розроблення складних ІС і їх подальшого супроводження завдяки модульному дизайну;

модульний дизайн дає змогу працювати над одним проєктом різним командам розробників та організовувати процеси розроблення з використанням аутсорсінгової моделі з використанням принципів спільного розроблення застосунків JAD (Joint Application Development);

модульний дизайн дає змогу оновлювати окремі частини ІС у разі виникнення проблем без необхідності внесення значних змін у дизайн інших частин ІС.

Недоліки об'єктно-орієнтованих методологій [3–5, 10, 13, 14]:

об'єктно-орієнтований підхід надмірно фокусується на коді, внаслідок чого поєднання об'єктного програмування з аналізом функцій системи є проблематичним;

читання об'єктно-орієнтованого коду не дає бачення загальної картини і часто неможливо переглянути всі функції, які викликаються функцією, яка була змінена;

ускладнений пошук причин виникнення помилок у коді через необхідність відстеження потоку виконання по всіх об'єктах і класах через які він пройшов та який до того ж може багато разів переміщуватись між декількома функціями-членами, перш ніж потік перейде до іншого об'єкта. Унаслідок цього іноді неможливо швидко виявити помилки потоку виконання за

допомогою простого перегляду коду, створеного під час об'єктно-орієнтованого проєктування на відміну від коду, створеного мовою низького рівня (Assembly code);

проблематичність відстеження потоків виконання і, відповідно, визначення більш ефективного з них під час виконанні функцій;

визначення всіх класів і об'єктів, необхідних для розроблення складних інтегрованих інформаційних систем, є непростим завданням і потребує специфічної і довшої підготовки розробників ІС та залучення бізнес-аналітиків;

програми, які створюються на основі взаємодії об'єктів є складнішими для розуміння, а деякі методи програмування, зокрема успадкування та поліморфізм, викликають когнітивні ускладнення;

об'єктному підходу до моделювання даних складніше співіснувати з реляційними моделями даних, які присутні в переважній більшості вже існуючих ІС, тому нові ІС часто розробляються, або модифікуються з використанням комбінації об'єктно-орієнтованих мов програмування і реляційних баз даних;

наслідком об'єктного підходу та програмування різних модулів окремими командами програмістів під час створення великих ІС є також випадки нерозуміння яким чином обробляти помилки потоку виконання, якщо програміст або команда програмістів не були знайомі з об'єктами, з якими працювала інша команда програмістів;

програми, створені об'єктно-орієнтованими мовами загалом працюють повільніше, ніж програми створені процесно-орієнтованими мовами, оскільки потребують виконання більшої кількості інструкцій і є більш вибагливими до продуктивності апаратної платформи;

не підходить для всіх типів завдань: є завдання, які краще піддаються стилю функціонального програмування, стилю логіки програмування або стилю програмування на основі процедур, а застосування об'єктно-орієнтованого програмування в таких ситуаціях не призводить до ефективних програм.

З огляду на велику кількість чинників, які можуть впливати на рішення з вибору підходу до моделі проєктування та методів програмування ІС ВП, пропонується під час передпроєктної підготовки скоротити кількість можливих варіантів проєктних моделей, технологій, методик та інструментів проєктування через застосування алгоритму вибору моделі проєктування та методу програмування за сімома критеріями (рис. 1).

Критерій 1. "*Оцінка ймовірності та складності проєктних ризиків*". Якщо передбачається велика кількість інноваційних рішень і не існує близьких аналогів розроблюваної АС ВП, то високий рівень ризикованості обумовлює вибір моделі спірального проєктування, під час якого розробник реалізує частину загального набору вимог на прототипі "X", оцінює ризики та відповідність прототипу вимогам і переходить до наступного кола проєктування з розроблення прототипу "X+1" з додаванням наступного набору функцій, або поліпшення продуктивності вже розроблених. Для такого проєкту підходить ***об'єктно-орієнтоване програмування***.

Критерій 2. "*Сталість вимог до АС ВП*" застосовується у разі, якщо кількість ризиків незначна, алгоритми рішення добре відомі, є ринкові аналоги для порівняння та оцінювання, то чи передбачає замовник часту зміну вимог. Якщо не передбачає, що є традиційною практикою у сфері безпеки і оборони України, хоча і застарілою через безпрецедентне зростання можливостей сучасних програмно-технічних рішень, розроблюваних програмною індустрією і новітнім підходом до оборонного планування, з урахуванням загроз майбутнього, зокрема у кіберпросторі, то для уточнення додається третій критерій – "складність вимог". Якщо складність незначна, то оптимальним вибором є класична "каскадна модель" і процесний підхід з використанням методів структурного аналізу.

Критерій 3. "*Складність вимог*". Якщо вимоги складні, що актуально в умовах гетерогенного інформаційного середовища, типового для бюджетних організацій сектору безпеки і оборони із значною кількістю історичних систем, з якими необхідно інтегрувати новий компонент, або здійснити міграцію даних з історичних систем у нову АС ВП і до того ж передбачати дотримання жорстких вимог щодо кібербезпеки вже на етапі концептуального проєкту, особливо в контурах інформаційного середовища, де циркулюють дані та інформація з обмеженим доступом, то доцільним є використання "V-моделі" (різновид "каскадної" моделі). При цій моделі розробник на кожному етапі здійснює детальне планування сценаріїв тестування функцій, інтеграції в інфосередовище та окремих модулів, що можливе за умови застосування ***процесного підходу*** та результатів структурного аналізу.

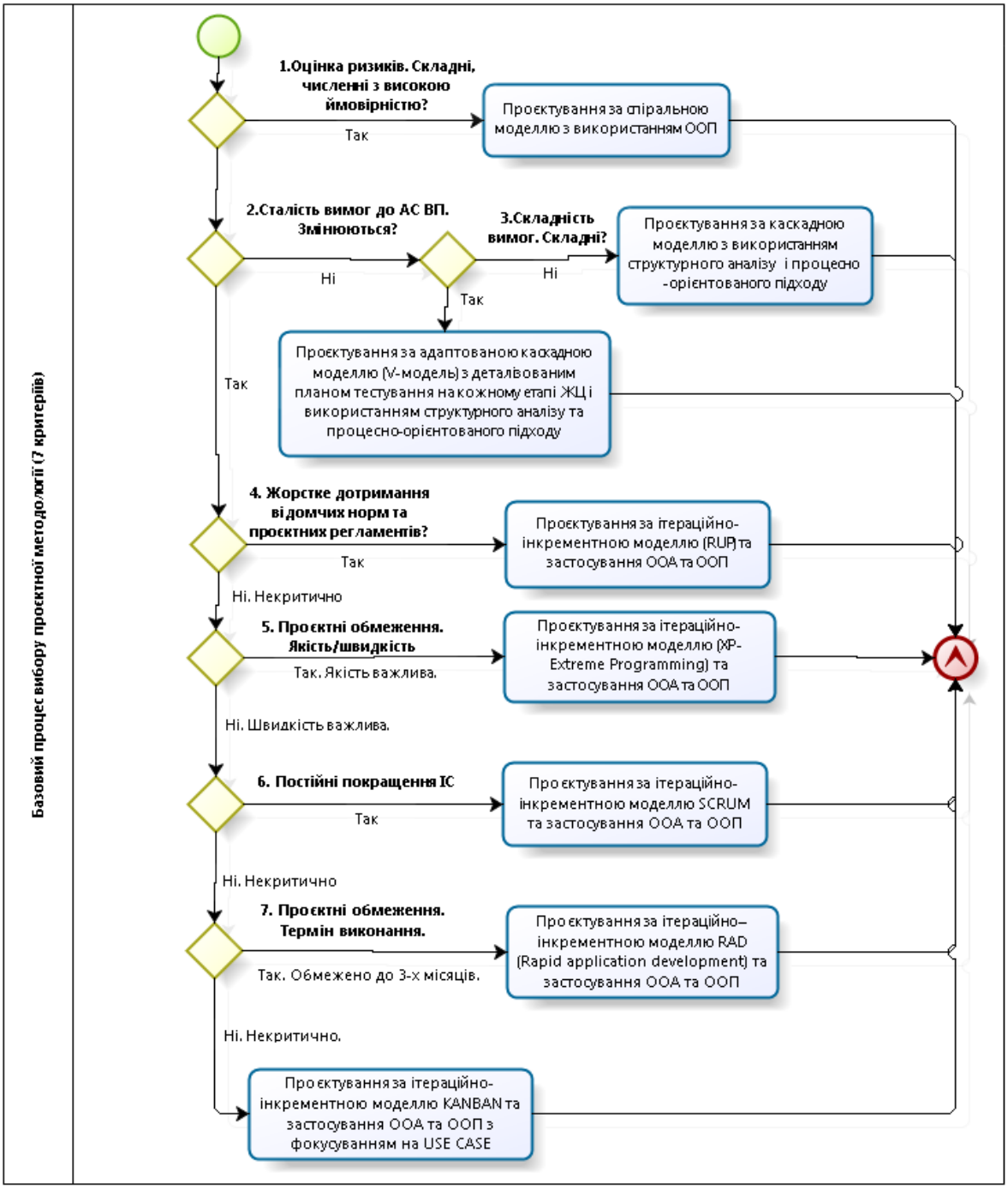

Рис. 1. Алгоритм вибору моделі проєктування та методу програмування за сімома критеріями

Критерій 4. "*Дотримання галузевих вимог і відомчого регламенту*" - передбачається необхідність гнучкого реагування на можливу зміну вимог до ІС ВП унаслідок швидких змін інформаційного середовища ЗС України та/або поява проривних технологічних рішень в управлінні військами та/або зброєю. Тоді доцільним є використання ітераційно-інкрементної моделі проєктування (наприклад UP, RUP від IBM, або модифікація OpenUP для невеликих проєктів з фокусуванням на управління ризиками) та застосування ***об'єктно-орієнтованого аналізу та об'єктно-орієнтованого програмування***.

Критерій 5. Проєктні обмеженням за швидкістю релізів і якістю. У разі, якщо *якість важливіша*, то доцільним є використання ітераційно-інкрементної моделі проєктування (наприклад, XP – Extreme Programming), у якій новий функціонал додається в ІС мінімально можливими обсягами одночасно з тестуванням, практикується спільне розроблення застосунків (JAD) та інтеграція з уже розробленою архітектурою системи під час кожного додаванні. Модель придатна для малих і середніх проєктів. У цьому разі найдоцільніше використовувати ООА та

ООП, а проєктній команді потрібно мати архітекторів, аналітиків та розробників, які вже мають досвід використання цієї моделі.

Критерій 6. У разі, коли вимога швидкості додавання нових функціональних можливостей в ІС критично важлива, тоді необхідно застосувати шостий критерій – "*необхідність постійних покращень додатків*", при якому доцільно використовувати модифікацію ітераційно-інкрементної моделі проєктування SCRUM, у якій практикуються періодичні уточнення щодо змісту майбутніх покращень та критичний розгляд щодо результатів додавання попереднього покращення, а також метод спільного розроблення застосунків (JAD) за допомогою ООП**.** Цей підхід спрацьовує у разі розроблення інноваційних інформаційних технологій, нестандартних програмних рішень, які не мають аналогів і коли команда розробників потребує злагодження.

Критерій 7. Проєктне обмеження за терміном виконання. Якщо нагальної потреби в постійному покращенні немає, але є проєктне обмеження за часом, тоді використовується сьомий критерій – "***обмеження часу на цикл розроблення від задуму до релізу від двох до трьох місяців***". У цьому разі доцільно використовувати ітераційно-інкрементну модель розроблення згідно з методологією RAD (Rapid Application Development), створеної як розвиток спіральної моделі, та методи об'єктно-орієнтованого проєктування, а якщо обмеження на час відсутні або незначні та потрібно додавати нові можливості до ІС згідно з дорожньою картою рівномірного удосконалення, тоді доцільно застосувати різновид методології Agile ітераційно-інкрементну модель проєктування – KANBAN, у якій доцільне використання ООП на основі клієнтських сценаріїв користування системою (USE CASE) для наближення готового продукту до користувацьких потреб.

Висновок. Вибір рішення щодо застосування методології та технології проєктування залежить від особливостей проєкту в конкретній організації, вимог до майбутньої ІС, особливостей тієї чи іншої технології, ресурсного забезпечення. Доцільно обирати методологію та методи проєктування, які відповідають рівню складності об'єкта автоматизації, інформаційному середовищу, у якому функціонуватиме ІС ВП, організаційній зрілості суб'єкта-замовника і проєктним обмеженням (час, бюджет, доступність технологій і кадрів). Метою використання методології та технології проєктування є зменшення складності процесу створення ІС та зменшення ймовірності виникнення типових проєктних проблем завдяки адекватному за витратами набору моделей ІС ВП. З метою зменшення складності прийняття рішення пропонується використати на передпроєктному етапі алгоритм відбору моделі проєктування та методу програмування за сімома критеріями.

Зважаючи на підвищений рівень загрози національній безпеці під час використання ІС ВП створених неналежним чином, критерій наявності досвіду використання проєктних методологій і технологій програмування має входити до загального переліку вимог до виконавців, тому доцільно на стадії попередньої кваліфікації та оцінювання пропозицій претендентів на роль виконавців створення ІС ВП застосовувати критерій "досвід використання проєктних методологій і технологій програмування окремо чи в комбінації" у розрізі двох показників: 1) "доступність персоналу, який виконував роботи з використанням проєктної методології та технології програмування", 2) "кількість проєктів, здійснених з використанням проєктної методології та технології програмування" за участі заявленого персоналу претендента за 5 років до подачі пропозиції.

Уточнення критеріїв і методів оцінювання проєктного досвіду претендентів на роль виконавців створення ІС ВП є **напрямом подальших досліджень.**

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. John McCain, Carl Levin. Report on the Air Force's Expeditionary Combat Support System (ECSS). U.S. Senate Committee on Homeland Security & Governmental Affairs. Permanent Subcommittee on Investigations. URL: https://www.hsgac.senate.gov/subcommittees/investigations /media/permanent-subcommittee-on-investigations-releases-report-on-ecss-program (дата звернення: 05.06.2020).
2. 8th Global Project Management Survey of Project Management Institute "Pulse of profession 2016". URL: https://www.pmi.org/-/media/pmi/documents/ public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2016.pdf (дата звернення: 05.06.2020).
3. Инюшкина О. Г., Кормышев В. М. Исследование систем управления при проектировании информационных систем : учеб. пособ. Екатеринбург : Форт-Диалог Исеть, 2013. 370 с.
4. Инюшкина О. Г. Проектирование информационных систем (на примере методов структурного системного анализа) : учеб. пособ. Екатеринбург : Форт-Диалог Исеть, 2014. 240 с.

5. Рудаков А. В. Технология разработки программных продуктов : учебник для студ. сред. проф. образования / 7-е изд., стер. Москва : Академия, 2012. 208 с.
6. Карпенко М. Ю. , Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем : навч. посіб. Харків : ХНУМГ імені О. М. Бекетова, 2017. 94 с.
7. Черемных С. В., Семенов И. О., Ручки В. С. Структурный анализ систем: IDEF-технологии. Москва : Финансы и статистика, 2003. 208 с.
8. Арлоу Д., Нейштадт А. UML2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. Санкт-Петербург : Символ-Плюс, 2007. 624 с.
9. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. Москва : И. Д. Вильямс, 2008. 720 с.
10. Asaf Shelly. Flaws of Object Oriented Modeling. Intel Software, Developer Zone, Blogs. URL: https://software.intel.com/en-us/blogs/2008/08/22/ flaws-of-object-oriented-modeling/ (дата звернення: 08.06.2020).
11. Вичугова А. А. Методы и средства концептуального проектирования информационных систем: сравнительный анализ структурного и объектно-ориентированного подходов. *Прикладная информатика*. Москва, 2014. № 1(49). С. 56–65.
12. Коцюба И. Ю., Чунаев А. В., Шиков А. Н. Основы проектирования информационных систем : учеб. пособ. Санкт-Петербург : Университет ИТМО, 2015. 206 с.
13. Alshamrani1 A., Bahattab A. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*. Vol. 12, Iss. 1, No 1. January 2015. P. 106–111. URL: https://www.ijcsi.org/papers/IJCSI-12-1-1-106-111.pdf (дата звернення: 08.06.2020).
14. Nabil Mohammed Ali Munassar, Govardhan A. A Comparison Between Five Models Of Software Engineering. *International Journal of Computer Science Issues*. Vol. 7, Iss. 5. September 2010. P. 94–100. URL: https://www.researchgate.net/publication/ 258959806_A_Comparison _Between_Five_Models_Of_Software_Engineering (дата звернення: 09.06.2020).

Стаття надійшла до редакційної колегії 01.07.2020

The main differences and features of structural and object-oriented approaches into designing of military information systems

Annotation

The experience of world's leading countries conflicts shows that it is impossible to ensure the operation of troops (forces) effectively, without the automation of the management of tangible and intangible resources.

The article considers the differences and features, advantages and disadvantages of two methodologies for designing information systems. The paper proves that the application of methodologies and the final choice of methods and means of designing military information systems depends on the following important components:

of the general requirements to the system;
of the level of technology innovation and complexity of the system;
of the volume and details of the description of the object of automation;
regulatory and design restrictions to the system;
needs for iterative improvement of the system;
the needs for integration with previous created systems;
the level of organizational and technological maturity of the customer organization;
of the availability and experience of performers of the system.

The value of the article in providing next recommendations:

for architects of the system - criteria to facilitate the solution of the problem of choosing the design methodology and tools for implementing the information system;

for project managers - information on the risks associated with methodologies;

for heads of functional groups of information developers - for the selection of qualified specialists who are able to effectively use the methods and tools of design that was selected by the architects of the system;

customers may have a better understanding of the importance of clearly formulating system requirements and providing primary data for further modeling.

In total the use materials of the article provides an opportunity for all project participants to optimize resources and time for pre-project preparation and improve the likelihood of successful implementation of project objectives.

Keywords: information system; design of Information Systems; structural analysis; object-oriented analysis; criteria for selecting a design model.