

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет імені Тараса Шевченка

АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лабораторний практикум
для студентів спеціальності
121 «Інженерія програмного забезпечення»

Київ 2019

УДК 004.412 (076.5)
ББК В 192.145 я 7
А 605

Укладачі: С. В. Поперешняк, *к.ф.-м. н., доцент*

І. А. Юрчук, *к.ф.-м. н., доцент*

Рецензент: д.т.н., професор, завідувач кафедри інтелектуальних технологій факультету інформаційних технологій Київського національного університету ім. Тараса Шевченка *Снитюк Віталій Євгенович*

Схвалено Вченою радою факультету інформаційних технологій
(протокол № 6 від 4 грудня 2019 року)

А Аналіз вимог до програмного забезпечення
 лабораторний практикум / уклад. С. В. Поперешняк, І.А. Юрчук – ,
 2021. – 22 с.

Містить завдання, методичні вказівки та варіанти для виконання лабораторних робіт. Складено відповідно до робочої програми дисципліни «Аналіз вимог до програмного забезпечення».

Для студентів спеціальності 121 «Інженерія програмного забезпечення».

ВСТУП

Метою лабораторного практикуму з дисципліни «Аналіз вимог до програмного забезпечення» є вивчення основ наукових концепцій, понять та технологій, які використовуються при аналізі вимог з метою більш ефективної розробки програмного забезпечення, управління його якістю та наукових досліджень.

Лабораторні роботи структуровані за модульним принципом. До складу модуля 1 «Аналіз та встановлення вимог» входить сім лабораторних робіт, які призначені для вивчення концепції, встановлення призначення системи та аналізу вимог за допомогою UML та IDEFX. Модуль 2 «Специфікація та відстеження вимог» складається з семи лабораторних робіт, призначених для вивчення специфікації різних типів вимог, метрик, орієнтованих на вимоги та деяким засобам керування та відстеження вимог.

Кожна лабораторна робота повинна виконуватися особисто студентом під керівництвом викладача згідно з його варіантом завдання. Номер варіанта для виконання лабораторних робіт надається викладачем.

Опис кожної лабораторної роботи містить завдання для виконання роботи, рекомендації щодо реалізації завдання, контрольні запитання та посилання на відповідну літературу.

Виконанню кожної лабораторної роботи повинна передувати самостійна підготовка студента за конспектом лекцій і джерелами, наведеними в списку літератури.

Після виконання роботи студент повинен продемонструвати і пояснити викладачеві, який проводить заняття, результати роботи, оформити електронний звіт та відповісти на контрольні запитання.

МОДУЛЬ 1

АНАЛІЗ ТА ВСТАНОВЛЕННЯ ВИМОГ

1. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.1

ПОСТАНОВКА ПРОБЛЕМИ. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Мета: створити попередній опис проблеми у деякій предметній галузі та проаналізувати існуючі програмні продукти, які в ній функціонують.

Завдання

1. Вибрати предметну область із запропонованого переліку або власну узгоджену з викладачем (Додаток 1).
2. Зробити опис обраної предметної галузі
3. Зробити опис проблеми, яка підлягає вирішенню.
4. Здійснити пошук в мережі Інтернет п'яти існуючих програм-аналогів (сайтів) для обраної предметної області (вітчизняні та закордонні ресурси).
5. Описати кожну із розглянутих програмних систем за наступним планом:
 - назва продукту, компанія-виробник;
 - основні функціональні можливості;
 - переваги програмної системи;
 - недоліки програмної системи.
6. Здійснити порівняння існуючих програм-аналогів і внести дані в таблицю 1.

Таблиця 1.

	Програмний засіб 1	Програмний засіб 2	Програмний засіб 3	Програмний засіб 4	Програмний засіб 5	Власний програмний засіб
Властивість 1	+/-	+/-	+/-	+/-	+/-	
Властивість 2	+/-	+/-	+/-	+/-	+/-	
Властивість 3						
...						
Властивість n	+/-	+/-	+/-	+/-	+/-	

7. На основі проведеного аналізу скласти перелік вимог до програмної системи для заданої предметної області

Література: [1-4]

2. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.2

СТВОРЕННЯ БІЗНЕС-ВИМОГ

Мета: створити бізнес-вимог до власного проекту, який функціонує в обраній з предметній області з лабораторної роботи №1.1.

Завдання

1. Ознайомитися із основними процесами створення бізнес-вимог.
2. Провести попередній аналіз предметної галузі та створити бізнес-вимоги до системи, що задана згідно з варіантом (Додаток 1).
3. Заповнити таблицю з бізнес-цілями (див. Таблицю 2).
4. Сформулювати образ проекту.

Основні теоретичні відомості

УВАГА! Бізнес-вимоги при створенні програмного забезпечення не відносяться до теми “Бізнес”, це спеціальний розділ технічного завдання, який створюють бізнес-аналітики, та який має бути обговорений з замовником.

Цю тему без теорії важко зрозуміти, а при невиконанні цього завдання – при здачі будь-якої лабораторної роботи або на іспиті подальші завдання навіть не перевіряються, тому що без розуміння **ДЛЯ ЧОГО** ми створюємо цей проект – нема відповіді на питання, чи правильно ми зробили подальші завдання (юз-кейси або прототипи).

Для відповіді, що є бізнес вимогами треба прояснити для себе відповіді на такі запитання:

1. Проблема та задача, яку має вирішити проект. В чому є головна “фішка” проекту.
2. Для яких потреб ми створюємо наш проект і яким чином зараз вирішується така проблема. Необхідно навести приклади схожих програмних систем або сайтів та пояснити, в чому відмінність нашого рішення.
3. Для комерційного проекту визначити ринок, для якого створюється проект. Ринок – це хто та як буде купувати наше рішення. Від цього залежить, як буде виглядати проект. Скільки плануємо користувачів, скільки користувачів зможуть одночасно працювати з сервісом.

Для інформаційної системи (яка не ставить за мету продаж послуг користувачам) – яку задачу планується вирішити. Як ці задачі вирішуються зараз, в чому переваги нашого рішення.

4. Які бізнес-цілі планується досягти після реалізації проекту. Як ми будемо вимірювати цілі? Які цілі, що неможливо виміряти? Приклад бізнес-цілей є в таблиці, Треба обрати цілі проекту та описати показники.

Таблиця 2. Приклади бізнес цілей

Фінансові	Нефінансові
Досягти x% ринку за Y місяців	Розробити платформу
Досягти об'єму продаж X одиниць в місяць за Y місяців	Розробити систему обліку товарів на складі в режимі реального часу
Досягти доходу X гривень в місяць за Y місяців	Зменшити час обслуговування клієнтів
Зменшити вартість обслуговування системи до X гривень на рік	Збільшити впізнаваність бренду
Отримати X дзвінків на день від потенційних покупців	Отримати позитивні відгуки в професійному осередку
Повернути вкладені в проект гроші за X місяців	Покращити імідж компанії за рахунок створення модернового меню
Збільшити кількість осіб, які підписались на сервіс, до X за Y місяців	Досягти рівня задоволення клієнтів на 15%

Для інформаційної системи вказати, які показники ми покращимо. Скільки клієнтів будуть використовувати систему, скільки людей треба на підтримку системи

УВАГА! Бізнес-цілі з таблиці не можна просто переписати. По-перше, цілі в таблиці протирічать одна одній, треба обрати власне ВАШІ! По-друге, треба вказати КОНКРЕТНІ ЗНАЧЕННЯ (їх в таблиці нема!) для вашого проекту.

5. Висновки з пунктів 1-4 представити у вигляді **образу проекту** (це термін) за схемою:

Для кого

який (яка) мають потребу,

Мій продукт (система)

Має можливість задовольнити потребу

На відмінність від

Наш продукт має переваги

Наприклад, для молодих мам, у яких немає часу ходити в магазин, **наша система** дозволить швидко зробити замовлення дитячого харчування. **На відміну від** всіх інших інтернет-магазинів ми забезпечимо доставку

товару 24 години на добу без вихідних (24/7) протягом 30 хвилин в будь-яку точку Києва.

УВАГА! ОБРАЗ ПРОЕКТУ ВИДІЛИТИ ЖИРНИМ ШРИФТОМ (як цей текст)

Дати відповіді на наступні питання:

1. Які технології і ресурси потрібні для розв'язання цієї задачі?

Скільки є грошей на створення проекту?

Скільки треба фахівців, яких?

Скільки треба грошей та часу на створення проекту?

Де буде розміщений проект (домени, хостінги, хмарні сховища)?

Яка планується база даних?

Це один проект чи тільки перший етап (реліз).

2. Визначити всіх користувачів системи (Екторів).

Користувачі – це ВСІ, хто працює з системою. Якщо хтось отримує інформацію від системи, завантажує інформацію до системи, включає/виключає, передає інформацію – це користувач.

Часто користувачами студенти невірно вважають тільки клієнтів.

На цьому етапі користувачів треба тільки назвати, в подальших завданнях ми будемо з ними розбиратися більш детально.

Література: [2, 15-19]

3. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.3

ВИВЧЕННЯ КОНЦЕПЦІЇ

Мета: ознайомитися з основними принципами формулювання концепції системи та набутти практичних навичок зі створення документу-концепції системи.

Завдання

1. Ознайомитися із основними процесами розробки концепції.
2. Вивчити структуру документу-концепції системи (Додаток 2).
3. Провести попередній аналіз предметної галузі та створити документ-концепцію системи, що задана згідно з варіантом (Додаток 1).

Основні теоретичні відомості

Інженерія вимог – це розділ програмної інженерії, що займається проблемами отримання вимог до програмного забезпечення їх документування, а також проблемами управління вимогами. Тобто інженерія вимог полягає у виявленні вимог, їх описі (створенні специфікації) і валідації.

Вимоги – це сукупність тверджень відносно атрибутів властивостей або якостей програмної системи що підлягають реалізації.

Замовник і розробник спільно виявляють вимоги, аналізують, переглядають, визначають необхідні обмеження і умови, а також описують їх.

Фази розробки вимог:

1) Виявлення вимог – збір, розуміння, розгляд та з'ясування потреб зацікавлених сторін;

2) Аналіз вимог – уточнення, встановлення пріоритетів, перевірка закінченості та цілісності вимог;

3) Специфікація вимог – формалізований опис вимог;

4) Верифікація вимог – перевірка правильності специфікації вимог.

Концепція (бачення) – коротке і чітке формулювання мети створення нового програмного продукту або покращення існуючого, тобто опис суті майбутнього програмного продукту. Основне призначення концепції – дати загальне уявлення про продукт, зібрати бізнес-вимоги (бізнес-цілі, поставлені перед продуктом).

Результатом розробки концепції є документ, що містить детальну інформацію про цілі і завдання створення продукту, вимоги замовника, можливості продукту, які задовольняють ці вимоги, а також орієнтовані його терміни та бюджет. По закінченню розробки концепції продукту робиться висновок про доцільність його розробки.

Серед розробки концепції виділяють такі процеси:

1) Визначення ідеї та потреб;

2) Формування потенційних підходів;

3) Вивчення здійсненності;

4) Уточнення та оформлення ідей потреб.

Визначення ідей та потреб – процес визначення інформації відносно того, які проблеми та в яких умовах вирішуються за допомогою майбутнього продукту (визначення сценаріїв роботи користувачів із майбутнім продуктом). вхідними даними процесу є вимоги замовника, ідеї від розробника, маркетингова інформація, вимоги користувачів та змінені вимоги до програмного забезпечення. Результат – документ, в якому наводяться попередньо сформовані потреби.

Формування потенційних підходів – процес, під час виконання якого розробляються або виділяються прийоми та способи вирішення певних завдань. При цьому використовуються попередньо сформульовані потреби, відповідні інформаційні джерела, дані про бюджет, інформація про наявне повторно використовуване програмне забезпечення та інші дані, які є дочірніми при прийнятті рішення про розробку/придбання продукту. Вхідні

дані процесу – бюджет та ресурси розробки, дані, що стосуються ринку розповсюдження продукту. Результатом даного процесу є опис потенційних підходів, їх обмежень та переваг.

Проведення вивчення здійсненності – процес, що включає аналіз ідей та потреб, потенційних підходів, обмежень та переваг життєвого циклу розробки програмного забезпечення для визначення можливості розробки програмного продукту. Вхідні дані процесу – документ, у якому наведено попередньо сформульовані потреби. Вихідні дані – рекомендації, у яких зазначається, чи потрібно розробляти проект “з нуля”, чи можна купити якусь його частину або взагалі відмовитися від його розробки.

Уточнення та оформлення ідей та потреб – процес, під час якого визначаються цілі програмного продукту, потреби та побажання замовника, рекомендовані підходи до реалізації. Вхідні дані – документ, у якому представлені попередньо сформульовані потреби, обмеження та переваги, потенційні підходи та рекомендації. Вихідні дані – документ, у якому зазначаються погоджені потреби.

Контрольні питання

- 1) Що означає поняття «інженерія вимог»?
- 2) Що таке вимога? Які існують види вимог?
- 3) Яка відмінність між функціональними та не функціональними вимогами?
- 4) Які існують методи виявлення вимог? Розкрийте їх сутність.
- 5) Як визначаються джерела та користувачі вимог?
- 6) Які параметри якості вимог існують? Опишіть їх особливості.
- 7) У чому полягає особливість концепції системи?
- 8) Яка структура документу-концепції?

Література: [1, 2, 3, 8].

4. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.4

ВИМОГИ КОРИСТУВАЧІВ, ФУНКЦІОНАЛЬНІ ТА НЕФУНКЦІОНАЛЬНІ ВИМОГИ В ПРОЦЕСІ СТВОРЕННЯ ПЗ.

Мета: набути навичок у формулюванні вимог користувачів, а також функціональних та нефункціональних вимог у процесі створення ПЗ.

Завдання

1. Зробити три інтерв'ю з actors системи – користувачем сервісу та директором (СЕО, фінансистом, власником тощо).

2. Виконати роботу з визначення функціональних вимог власного проекту.

3. Визначити деякі системні вимоги.

Основні теоретичні відомості

На початковому етапі виконання роботи необхідно визначити всіх користувачів системи. Для кожного з них провести підготовку до проведення інтерв'ю, яку зазвичай оформляють у вигляді плану.

Приклад: план проведення інтерв'ю із відвідувачем кафе.

Проект – передзамовлення в кафе	Реліз 0.1
Назва документу – Карта тем – Інтерв'ю з відвідувачем	Автор Поперешняк С.В.
1520-00 Відвідувач – план типового інтерв'ю	

Вступ.

Даний документ призначений для розуміння вимог до сервісу “Передзамовлення” з боку клієнтів кафе.

Опис проекту.

Проект - Створення та впровадження системи яка дозволяє існуючим і новим клієнтам кафе здійснити онлайн передзамовлення на обслуговування. Включаючи формування замовлення на кухню, бронювання столика, оплату обслуговування та інші послуги які надаються в кафе

План проведення інтерв'ю

Цілі проведення інтерв'ю

1. Зібрати інформацію про те, наскільки відома послуга онлайн замовлення клієнтам кафе, і чи користуються подібними послугами в інших кафе міста.

2. Як, на думку клієнтів, повинен працювати подібний сервіс.

	Тема	Відкриті питання	На що звернути увагу
	Початок бесіди	1. Подання. Питання про можливість діалогу. 2. Чи цікавить Вас послуга попереднього онлайн бронювання замовлення?	
	Інформація про клієнта	1. Ви постійний клієнт кафе або сьогодні перший раз відвідали кафе? 2. Як часто Ви користуєтеся послугами кафе (для постійних клієнтів)? 3. Скільки часу Ви зазвичай чекаєте виконання замовлення? 4. Чи цікавить Вас послуга попереднього бронювання?	
	Знайомство з	1. Ви коли-небудь користувался	

онлайн сервісами	сервісами онлайн бронювання в кафе? 2. Які, на Ваш погляд, найбільш вдалі сервіси? 3. Чому? 4. Де Вам зручно завантажити додаток?	
Потреби в онлайн бронюванню	1. Чи хочете Ви зменшити час очікування замовлення? 2. Чи хочете ви забирати замовлення в ресторані в призначений час за попереднім бронюванням?	
Бажані особливості бронювання	1. Як Ви вважаєте, послуга бронювання повинна бути з дисконтом? 2. Чи готові Ви прив'язати свою банківську карту до сервісу? 3. Які ще послуги були б Вам цікаві?	
Підведення підсумків	Дякую Вам за витрачений час.	

Вимоги користувачів та функціональні вимоги оформляють у вигляді таблиці, яка заповнюється за наступною схемою:

1. Вносимо всіх користувачів, які будуть працювати з системою.
2. Кожний користувач виконує певну кількість ролей (дій, сценаріїв, use case). Навпроти кожного користувача записати всі сценарії, які буде виконувати користувач в системі, що є результатом виконання сценарію та який сценарій буде виконуватись після.

Приклад оформлення таблиці функціональних вимог :

Користувач	Сценарій	Результат	Що далі	Примітка
Покупець	Замовлення товару з каталогу	Замовлення в корзині	Оплата товару	
	Оплата замовлення	Підтвердження оплати	Логістик відвантажує товар	

Для того, щоб оформити системні вимоги до продукту, необхідно дати відповіді на наступні питання:

1. Яке планується навантаження на систему (згідно бізнес-вимог)? Яке максимальне навантаження?
2. Які хмарні ресурси використовуються (які вендори мають хмарні сховища)
3. Яка система поштових розсилок використовується? Чому?

4. Яка інформація буде зберігатися в базах даних? Де будуть розміщені бази даних?
5. Чи відома платформа або мови програмування?
6. Які API ми маємо підключити до системи?
7. Чи планується WEB інтерфейс та мобільні додатки?

Література: [2, 20-27]

5. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.5

РОЗРОБКА СЦЕНАРІЮ РОБОТИ СИСТЕМИ.

Мета: вивчення бізнес-правил, алгоритмів та сценаріїв роботи ми будемо виконувати, створюючи сценарій роботи відео-іграшки мовою програмування Scratch.

Завдання

1. Підключитися до платформи онлайн курсів Prometheus та зреєструватися на курс CS50.
2. Прослухати лекцію №0.
3. Виконати завдання до лекції.
4. Створена відео-гра має відповідати всім вимогам лекції.
5. Відео-гра повина містити відповідні інструкції для користувачів, мати підказки, а також інтуїтивно зрозумілий інтерфейс.
6. Гру треба надати лектору згідно зазначених вимог (Мудл чи ГуглДиск, або на поштову скриньку викладача за згодою викладача) до наступної практичної роботи.
7. Виконати моделювання за допомогою мови моделювання BPNM

Основні теоретичні відомості

Ми живемо за алгоритмами – кожен день встаємо та виконуємо однотипну послідовність дій, ідемо на роботу та в університет, маючи в голові головний маршрут, та додатковий, який може використовуватись, наприклад, при аварії на дорозі, готуємо їжу за рецептами (алгоритмами), тощо.

Процес може бути описаний просто словами, або більш складними мовами моделювання, наприклад, BPNM.

Для ознайомлення з процесом моделювання ми використаємо мову Scratch. Ця мова розроблена Google для початківців, і тому написати на ній програму може будь-хто.

Тім не менш, процес розробки програми такий самий, як і при програмуванні на складних мовах. Хоча, чомусь, на Scratch всі пишуть програми! Чого не скажеш про інші мови програмування.

Зауважимо, що курс CS50 – один з найкращих світових курсів по комп'ютерних науках. Рекомендуємо, пройти його до кінця.

Література: [1, 2, 3, 8]

6. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.6

ВСТАНОВЛЕННЯ ПРИЗНАЧЕННЯ СИСТЕМИ

Мета: ознайомитися з особливостями процесів встановлення призначення системи та набути практичних навичок з виділення та формулювання функціональних вимог.

Завдання

1. Ознайомитися із процесами встановлення призначення системи.
2. Провести аналіз предметної галузі та визначити функції системи, що задана згідно з варіантом (додаток 1).
3. Побудувати системну архітектуру, використовуючи інструментальний засіб MS Visio та навести її текстовий опис.
4. Сформулювати функціональні вимоги.
5. Провести декомпозицію вимог, а також графічно представити за допомогою відповідних діаграм.

Основні теоретичні відомості

Встановлення призначення системи – процес визначення того, навіщо система використовується. При встановленні призначення системи визначають вид діяльності, що автоматизуватиметься, та перелік об'єктів автоматизації, на яких передбачається її використовувати.

Процесами встановлення призначення системи є:

- 1) Аналіз функцій;
- 2) Розробка системної архітектури;
- 3) Декомпозиція системних вимог;

Аналіз функцій. Предметна галузь, сформульовані потреби та рекомендації аналізуються для визначення функцій усієї системи. Визначені функції становлять собою функціональний опис системи і використовуються при розробці системної архітектури та декомпозиції системних вимог. Вхідні

дані процесу – інформація, отримана на етапі вивчення концепції; вихідні дані – функціональний опис системи.

Розробка системної архітектури. Системна архітектура – “фундаментальна” організація системи, реалізована в її компонентах, зв’язках цих компонентів один із одним і зовнішнім середовищем, і принципах, що визначають структуру і розвиток системи. Вхідними даними процесу є сформульовані потреби й функціональний опис системи; вихідні дані – системна архітектура.

Декомпозиція системних вимог. Декомпозиція – подання складного проекту у вигляді сукупності простих. Декомпозиція системних вимог полягає в розділенні функцій системи, що становлять собою функціональний опис системи відповідно до архітектури системи для формування функціональних вимог до програмного продукту, вимог до інтерфейсу системи. Вхідні дані процесу – функціональний опис системи та системна архітектура; вихідні дані – функціональні вимоги до розроблюваного продукту.

Контрольні питання

- 1) Яким чином здійснюється діяльність по встановленню призначення системи?
- 2) В чому полягає особливість аналізу функцій як процесу встановлення призначення системи?
- 3) Дайте визначення поняття «системна архітектура» та опишіть особливості її побудови.
- 4) Наведіть приклад декомпозиції системних вимог.
- 5) Яким чином здійснюється процес визначення вимог до програмного забезпечення, що імпортується?

Література: [3, 5, 8].

7. ЛАБОРАТОРНЕ ЗАНЯТТЯ 1.7

ФОРМУВАННЯ БАЧЕННЯ ПРОГРАМНОГО ПРОДУКТУ

Мета роботи: дослідити процес створення документу опису первинних вимог до програмного забезпечення та набути практичних навичок у виділенні та документуванні вимог.

Завдання

1. Ознайомитися з основними етапами розробки програмного забезпечення при структурному підході до програмування.

2. Ознайомитися з особливостями створення специфікації вимог до програмного забезпечення.
3. Дослідити предметну галузь, задану згідно з варіантом (додаток 1) та виділити функціональні та нефункціональні вимоги до програмного забезпечення.
4. Обрати підхід (методологію) до розробки ПЗ
5. Створити специфікацію функціональних вимог до програмного забезпечення на основі шаблону в залежності від обраної методології розробки ПЗ та відповідного формату представлення специфікації вимог
 - a. «Технічне завдання», ГОСТ 19.XXX.
 - b. «Технічне завдання», ГОСТ 34.XXX.
 - c. Software Requirements Specification, стандарт IEEE 830.
 - d. Документ Бачення в RUP
 - e. Документ Бачення в MSF

Основні теоретичні відомості

Роботи по формуванню бачення продукту та меж проекту зазвичай починаються на самій ранній фазі проекту, до початку широкомасштабних консультацій щодо виявлення детальних вимог, хоча в цілому наявність і послідовність даних кроків залежить від обраної методології.

Аналізуючи літературу з даної тематики, можна виділити наступні широко вживані ключові слова: з одного боку - концепція, бачення, образ, специфікація, технічне завдання, з іншого - рамки, межі, контекст.

У першому випадку мова йде про бачення того, якою має бути система. Обговорюються високорівневі вимоги (можливості, властивості) продукту і найбільш істотні обмеження. Ряд авторів, навпаки, наполягає на тому, що бачення повинно бути «нічим не обмеженим».

Поняття бачення широко вживано в бізнес-аналізі. Якщо у топ-менежменту компанії є уявлення про те, які ключові цілі, сегменти ринку, товарні позиції, прибуток повинні бути досягнуті, припустимо, через 5 років - значить, компанія має довгострокове бачення себе на ринку. Спосіб зняття обмежень при виробленні бачення дозволяє виробити новий погляд на речі, «піднятися над ситуацією», планувати майбутнє, відштовхуючись немає від поточних ресурсів і обмежень, а від стратегічних цілей, застосовуючи інновації, ноу-хау тощо.

Даний досвід формування бачення багато в чому переносимо і на процес розробки інформаційних систем: потрібно «побачити» в горизонті середньо- і (або) довгострокового планування, як АІС впишеться в організаційні процеси підприємства, які ключові вигоди вона дасть, які проблеми дозволить

вирішити. При пошуку нових методів і засобів управління підприємством на основі інформаційних технологій часто доводиться «перекроювати» існуючі бізнес-процеси; по суті впровадження АІС, що зачіпає істотний відсоток процесів підприємства, неминуче призводить до перебудови цих процесів з метою оптимізації діяльності підприємства, досягнення ключових факторів ефективності та ін.

У другому випадку (рамки, кордону, контекст) обговорюються такі питання, як межа системи і середовища, необхідні ресурси на створення системи, терміни. Побудувавши «нічим не обмежене бачення», рано чи пізно доводиться повернутися до таких прозаїчним речам, як бюджет, календарне планування, підбір персоналу, віхи проекту.

Чи завжди потрібно створювати документ «Концепція»? Чи слід розділяти бачення і межі?

Найчастіше Замовник усвідомлює необхідність автоматизації, як спосіб вирішення накопичених проблем. Сформулювавши для себе проблему, Замовник часто бачить і варіант її рішення, з яким приходить до Виконавця («мені потрібен сайт», «потрібно CRM-система» тощо). Кваліфікований Виконавець не повинен, стрімголов, поспішати вирішувати завдання в формулюванні Замовника. За образним висловом Г.Калянова автоматизувати процеси «як є» - все одно, що асфальтувати доріжки, по яких ходять корови.

В нотації RUP присутній важлива метафора: «Побачити проблему за проблемою». Концепція якраз і служить для того, щоб допомогти Замовнику виявити саме ті вимоги до системи, які допоможуть йому оптимізувати роботу свого підприємства в довгостроковій перспективі.

Тому етап формування концепції важливий, але він пред'являє і до Замовника і до Виконавця досить високі вимоги: Замовник повинен виділити ресурси і бути готовим до трудовитрат на спільний пошук рішень; Виконавець повинен мати достатню кваліфікацію як в сфері ІТ-, так і в сфері управління підприємствами, щоб розробляється засіб автоматизації дійсно принесло користь. Все вищесказане нітрохи не виключає можливість роботи без концепції: або мова йде про невеликий проект, закладати в бюджет якого етап вироблення концепції просто нерентабельно, або Замовник сам володіє достатньою кваліфікацією, щоб сформулювати вимоги до АІС, маючи «концепцію в голові» і час для консультування розробника.

Деякі аргументи за розділення бачення і меж були приведені вище. Провести чітку межу між цими поняттями пропонує, зокрема, процес MSF. В кінцевому підсумку, питання «розділяти або не розділяти» визначається обраною методологією.

Розглянемо основні вимоги до вироблення концепції, закладені в вітчизняних стандартах **ГОСТ**, міжнародних стандартах **IEEE**, методології **RUP** і **MSF**.

Концепція в ГОСТ

Відповідно до ГОСТ 34.601-90 «Автоматизовані системи. Стадії створення» [28], після етапу формування (виявлення) вимог до системи виконується етап розробки концепції системи.

Основні роботи етапи:

1. Вивчення об'єкта;
2. Проведення науково-дослідних робіт (НДР);
3. Розробка варіантів концепції АС;
4. Оформлення звіту про виконану роботу.

Так як даний етап хронологічно стоїть на другому місці, до його початку у Розробника на руках вже є документ, в якому зібрані основні вимоги користувачів.

Роботи над концепцією починаються з обстеження об'єкта автоматизації. Виконуються НДР, спрямовані на дослідження принципової можливості бути реалізованим вимог і можливих варіантів реалізації.

ГОСТ, на відміну від більшості сучасних методологій, в загальному випадку закладає багатоальтернативність варіантів концепції системи і планів їх реалізації. Кожен з опрацьованих варіантів оцінюється з позицій необхідних ресурсів і функціональності. Для варіантів мають бути подані оцінки переваг і недоліків. Корисність опрацювання кількох варіантів концепції полягає в тому, що Замовнику важко сформулювати самостійно бачення системи, в той час, як вибір з набору варіантів, представлених Розробником - цілком посильне завдання.

Крім того, концепція повинна відображати оцінки якості, умови приймання системи, оцінку ефекту, очікуваного від реалізації. При оформленні звіту необхідно привести обґрунтування пропонованого варіанту.

Розробка технічного завдання

Технічне завдання є документ, в якому сформульовані основні цілі розробки, вимоги до програмному продукту, визначені терміни і етапи розробки і регламентований процес приймально-здавальних випробувань. В розробці технічного завдання беруть участь як представники замовника, так і представники виконавця. В основі цього документа лежать вихідні вимоги замовника, аналіз передових досягнень техніки, результати виконання науководослідних робіт, передпроектних досліджень, наукового прогнозування тощо.

Порядок розробки технічного завдання

Розробка технічного завдання виконується в наступній послідовності. Перш за все, встановлюють набір виконуваних функцій, а також перелік і характеристики вихідних даних. Потім визначають перелік результатів, їх характеристики і способи подання.

Далі уточнюють середу функціонування програмного забезпечення: конкретну комплектацію і параметри технічних засобів, версію операційної системи і, можливо, версії і параметри іншого встановленого програмного забезпечення, з яким належить взаємодіяти майбутнього програмного продукту.

У випадках, коли розробляється програмне забезпечення збирає і зберігає деяку інформацію або включається в управління будь-яким технічним процесом, необхідно також чітко регламентувати дії програми в разі збоїв обладнання і енергопостачання.

1. Загальні положення

1.1. Технічне завдання оформляють відповідно до ГОСТ 19.106-78 на аркушах формату А4 і А3 по ГОСТ 2.301-68, як правило, без заповнення полів листа. Номери аркушів (сторінок) проставляють у верхній частині аркуша над текстом.

1.2. Лист затвердження і титульний лист оформляють відповідно до ГОСТ 19.104-78. інформаційну частину (анотацію і зміст), лист реєстрації змін допускається в документ не включати.

1.3. Для внесення змін і доповнень в технічне завдання на наступних стадіях розробки програми або програмного виробу випускають доповнення до нього.

Узгодження і затвердження доповнень до технічного завдання проводять в тому ж порядку, який встановлений для технічного завдання.

1.4. Технічне завдання повинно містити наступні розділи:

- вступ;
- найменування та область застосування;
- підстава для розробки;
- призначення розробки;
- технічні вимоги до програми або програмного виробу;
- техніко-економічні показники;
- стадії і етапи розробки; • порядок контролю і приймання;
- додатки.

Залежно від особливостей програми або програмного виробу допускається уточнювати зміст розділів, вводити нові розділи або об'єднувати окремі з них. При необхідності допускається в технічне завдання включати додатки.

2. Зміст розділів

2.1. Вступ повинен включати коротку характеристику області застосування програми або програмного продукту, а також об'єкта (наприклад, системи), в якому передбачається їх використовувати. Основне призначення введення - продемонструвати актуальність даної розробки і показати, яке місце ця розробка займає в ряду подібних.

2.2. У розділі «Найменування і область застосування» вказують найменування, коротку характеристику області застосування програми або програмного виробу та об'єкта, в якому використовують програму або програмне виріб.

2.3. У розділі «Підстава для розробки» повинні бути вказані:

- документ (документи), на підставі яких ведеться розробка. Таким документом може служити план, наказ, договір і т. п.

- організація, що затвердила цей документ, і дата його затвердження;
- найменування та (або) умовне позначення теми розробки.

2.4. У розділі «Призначення розробки» має бути вказано функціональне та експлуатаційне призначення програми або програмного виробу.

2.5. Розділ «Технічні вимоги до програми або програмного виробу » повинен містити такі підрозділи:

- вимоги до функціональних характеристик;
- вимоги до надійності;
- умови експлуатації;
- вимоги до складу і параметрів технічних засобів;
- вимоги до інформаційної та програмної сумісності;
- вимоги до маркування та упаковки; • вимоги до транспортування і зберігання;
- спеціальні вимоги.

2.5.1. У підрозділі «Вимоги до функціональних характеристик » повинні бути вказані вимоги до складу виконуваних функцій, організації вхідних та вихідних даних, тимчасовим характеристикам і т. п.

2.5.2. У підрозділі «Вимоги до надійності» повинні бути вказані вимоги до забезпечення надійного функціонування (забезпечення сталого функціонування, контроль вхідний і вихідний інформації, час відновлення після відмови і т. п.).

2.5.3. У підрозділі «Умови експлуатації» повинні бути вказані умови експлуатації (температура навколишнього повітря, відносна вологість і т. п. для обраних типів носіїв даних), при яких повинні забезпечуватися задані характеристики, а також вид обслуговування, необхідне кількість і кваліфікація персоналу.

2.5.4. У підрозділі «Вимоги до складу і параметрів технічних засобів» вказують необхідний склад технічних коштів із зазначенням їх технічних характеристик.

2.5.5. У підрозділі «Вимоги до інформаційної та програмної сумісності» повинні бути вказані вимоги до інформаційним структурам на вході і виході і методам рішення, вихідного коду, мов програмування. При необхідності повинна забезпечуватися захист інформації і програм.

2.5.6. У підрозділі «Вимоги до маркування та упаковки» в загальному випадку вказують вимоги до маркування програмного виробу, варіанти і способи упаковки.

2.5.7. У підрозділі «Вимоги до транспортування і зберігання» повинні бути вказані для програмного виробу умови транспортування, місця зберігання, умови зберігання, умови складування, терміни зберігання в різних умовах.

2.5.8. У розділі «Техніко-економічні показники» повинні бути вказані: орієнтовна економічна ефективність, передбачувана річна потреба, економічні переваги розробки в порівнянні з кращими вітчизняними і зарубіжними зразками або аналогами.

2.6. У розділі «Стадії і етапи розробки» встановлюють необхідні стадії розробки, етапи та зміст робіт (Перелік програмних документів, які повинні бути розроблені, узгоджені та затверджені), а також, як правило, терміни розробки і визначають виконавців.

2.7. У розділі «Порядок контролю і приймання» повинні бути вказані види випробувань і загальні вимоги до приймання роботи.

2.8. У додатках до технічного завдання при необхідності приводять:

- перелік науково-дослідних та інших робіт, обґрунтовують розробку;
- схеми алгоритмів, таблиці, описи, обґрунтування, розрахунки та інші документи, які можуть бути використані при розробці;
- інші джерела розробки.

У випадках, якщо будь-які вимоги, передбачені технічним завданням, замовник не пред'являє, слід в відповідному місці вказати «Вимоги не пред'являються».

Приклад 1.1. Розробити технічне завдання на програмний продукт, призначений для наочної демонстрації школярам програми сортування одновимірного масиву.

Документ Опис вимог (технічне завдання згідно ГОСТ 34.XXX).

Документ опису вимог є основним документом, який визначає вимоги та порядок створення програмного забезпечення та її прийняття при вводі в

експлуатацію. В Україні в якості такого документа використовується технічне завдання, зміст якого визначається діючим стандартом ГОСТ 34.602-89 на створення автоматизованих систем. Нижче наведено спрощений зміст технічного завдання:

1. Загальні відомості (повне найменування роботи, умовне позначення, шифр (номер) договору, найменування та реквізити замовника та розробника системи, планові терміни початку та закінчення робіт, відомості про джерела та порядок фінансування робіт).

2. Призначення та цілі створення ПЗ (від діяльності, що автоматизується, перелік об'єктів автоматизації, основні показники, що повинні бути досягнуті в результаті впровадження ПЗ).

3. Характеристика об'єктів автоматизації (короткі відомості щодо об'єкта автоматизації або посилання на документи, що містять такі відомості; відомості щодо експлуатації об'єкта автоматизації та характеристики оточуючого середовища).

4. Вимоги до ПЗ:

– вимоги до ПЗ в цілому (структура, основні компоненти або підсистеми, чисельність та кваліфікація персоналу, режим роботи; вимоги до: надійності, продуктивності, ергономіці, технічної естетиці, експлуатації, технічному забезпеченні та зберіганні, захисту інформації від несанкціонованого доступу, зберіганню інформації при аваріях, захисту від зовнішніх чинників, патентної чистоти, стандартизації та уніфікації);

– функціональні вимоги, які виконуються ПЗ (перелік підсистем, їх призначення, взаємодія між собою; вимоги для роботи з іншими системами, режиму функціонування, діагностуванню системи, перспективи розвитку та модернізації);

– вимоги до видів забезпечення (технічного, системного програмного, інформаційного).

5. Склад та зміст робіт зі створення системи (перелік етапів, терміни, результати, фінансування).

6. Вимоги до документування (перелік документів, які підлягають розробці).

7. Джерела розробки (перелік літератури та посилань).

Бачення в RUP

Кроки, які необхідно пройти для формування документа «Бачення»:

- ☐ Формулювання проблем.
- ☐ Ідентифікація співвласників
- ☐ Визначення меж системи
- ☐ Ідентифікація обмежень
- ☐ Формулювання постановки завдань

- ☐ Визначення можливостей системи
- ☐ Оцінка результатів

Для опису проблем пропонується шаблон, показаний в табл. 3.

Таблиця 3.

Проблема	(опис проблеми)
Кого стосується	(особи, яких стосується дана проблема).
Її наслідком є	(який вплив проблеми).
Успішне вирішення	(список деяких ключових переваг від успішного вирішення).

Ідентифікація співвласників передбачає пошук і фіксацію осіб зацікавлених в проекті - представників Замовника та Виконавця, інвесторів, зовнішніх експертів тощо.

Визначення меж системи являє собою нетривіальний процес. Для цього використовують контекстні діаграми [27]. RUP в пошуку кордонів пропонує відштовхуватися від акторів і варіантів використання.

Серед джерел обмежень зазвичай виділяють:

- ☐ Політичні,
- ☐ Економічні,
- ☐ Середовища,
- ☐ Технічні,
- ☐ Виконання,
- ☐ Системні.

Опис можливостей системи є формулювання високорівневих вимог.

Шаблон документа «Vision» RUP містить наступні основні розділи:

1. Введення
2. Позиціонування
3. Описи співвласників і користувачів
4. Короткий огляд виробу
5. Можливості продукту
6. Обмеження
7. Показники якості
8. Вартість і пріоритети
9. Інші вимоги до виробу
10. Вимоги до документації
11. Додаток.

У вступі описуються мета документа, його контекст (зв'язок і взаємовплив з різними проектами), визначення, акроніми та скорочення, посилання на інші документи, короткий зміст.

У розділі «позиціонування» поміщається визначення вирішуваної проблеми (проблем), вказується цільової замовник і досліджуються ділові переваги виробу перед аналогічними на ринку.

В описі співвласників і користувачів, крім власне опису цих двох груп, досліджується демографія ринку: цільові ринкові сегменти; розмір і темпи зростання ринку; існуючі конкурентні пропозиції на ринку; репутація Розробника на ринку;

Короткий огляд виробів містить резюме виробу, опис його перспектив і ключових можливостей, припущення і залежності, вказується вартість і її калькуляція, розглядаються питання ліцензування та інсталяції.

У розділі, присвяченому можливостям продукту, вони описуються більш детально, кожна - в окремому параграфі.

В розділ «Обмеження» слід виносити існуючі технічні, технологічні та ін. Обставини, які необхідно враховувати на даній стадії.

Розділ «Показники якості» містить опис найбільш істотних функціональних вимог до системи (ефективності, надійності, відмовостійкості тощо.).

Розділ «Старшинство і пріоритети» ранжує сформульовані раніше вимоги і можливості системи за ступенем важливості, черговості реалізації тощо.

Розділ «Інші вимоги до виробу» описує застосовувані стандарти, системні вимоги, експлуатаційні вимоги, вимоги до навколишнього середовища.

У вимогах до документації наводяться ключові характеристики керівництва користувача, довідки, керівництва по установці і конфігурації, файлу Read Me.

В додаток виносяться атрибути можливостей. RUP рекомендує наступний набір атрибутів: статус, вигода, обсяг робіт, ризик, стабільність, цільової випуск, призначення, причина.

Бачення / рамки в MSF

Згідно білій книзі MSF [29], на фазі вироблення концепції (envisioning phase) закладається одна з фундаментальних основ успіху проекту - створення і згуртування проектної групи на основі вироблення єдиного бачення. Проектна група повинна чітко уявити собі, що вона хоче зробити для замовника і сформулювати свою мету таким чином, щоб максимально мотивувати як замовника, так і саму проектну команду. Вироблення високорівневого погляду на цілі і умови проекту може розглядатися як рання форма планування; вона готує ґрунт для процесів створення детальних планів, які будуть здійснені безпосередньо під час фази планування.

Основними завданнями фази вироблення концепції є створення ядра проектної групи і підготовка документа загального опису і рамок проекту (vision / scope document). Формування бачення проекту і специфікування його рамок - не одне і теж, хоча для успіху проекту необхідно і те, і інше. Бачення (vision) - це нічим не обмежується уявлення про те, яким має бути рішення. Рамки (scope) ж дають чіткі межі того, що із запропонованого цим баченням буде реалізовано в умовах існуючих проектних обмежень.

Управління ризиками являє собою ітеративний процес, який здійснюється протягом всього життєвого циклу проекту. Під час фази вироблення концепції проектна група готує документ оцінки ризиків та представляє головні ризики проекту разом із загальним описом і рамками проекту. Для отримання подальшої інформації про управління ризиками, див. "Білу книгу" дисципліни управління ризиками MSF.

Також під час фази вироблення концепції проводиться виявлення і аналіз бізнес-вимог. Більш детально ці вимоги розглядаються під час фази планування.

Провідним рольовим кластером на фазі вироблення концепції є "Управління продуктом".

Шаблон MSF складається з таких розділів:

- ☐ Бізнес-переваги,
 - о Опис переваг
 - о Формулювання бачення
 - о Аналіз вигод
- ☐ Концепція рішення
 - о Цілі, завдання, припущення і обмеження
 - о Аналіз застосовності
 - о Вимоги
- ☐ Рамки
 - о Список характеристик / функцій
 - о Поза рамками
 - о Стратегія підготовки релізів
 - о Критерії застосовності
 - о Експлуатаційні критерії
- ☐ Стратегії проектування рішення
 - о Стратегія проектування архітектури
 - о Стратегія технічного проектування

Контрольні питання

1. Наведіть етапи розробки програмного забезпечення.
2. Що включає в себе постановка задачі і передпроектні дослідження?
3. Перерахуйте правила розробки технічного завдання.
4. Які стандарти використовуються для складання технічного завдання?
5. Назвіть основні розділи технічного завдання.
6. Назвіть основні розділи опису бачення згідно RUP.
7. Назвіть основні розділи Документу Бачення.

Література [28-30]

СПИСОК ЛИТЕРАТУРЫ

1. *Леффингуэлл Д.* Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл, Д. Уиндрик. – М.: «Вильямс», 2002. – 448с.
2. *Вигерс К.* Разработка требований к программному обеспечению / К. Вигерс. – М.: «Русская редакция», 2004. – 576 с.
3. *Халл Э.* Разработка и управление требованиями. Практическое руководство пользователя / Э. Халл, К. Джексон, Д. Дик. Лондон.: Спрингер Сайнс, 2005. – 229 с.
4. *Кармайл Э.* Быстрая и качественная разработка программного обеспечения / Э. Кармайл, Д. Хейвуд. – М.: «Вильямс», 2009. – 263с.
5. IEEE Standard for Developing Software Life Cycle Processes: IEEE Std. 1074-1997. – N.Y.: The Institute of Electrical and Electronics Engineers, 1997. – 88 p.
6. IEEE Recommended Practice for Software Requirements Specifications: IEEE Std. 830-1998. – N.Y.: The Institute of Electrical and Electronics Engineers, 1998. – 31 p.
7. *Кватрани Т.* Визуальное моделирование с помощью Rational Rose 2002 и UML – Пер. с англ. – М.: «Вильямс», 2003. – 192 с.
8. *Кендал С.* UML, основные концепции / С. Кендал. – М.: «Вильямс», 2009. – 275 с.
9. *Брауде Э.* Технология разработки программного обеспечения / Э. Брауде. – СПб.: Питер, 2004. – 655 с.
10. Data-Flow-Diagram-Notations [Электронный ресурс]. – Режим доступа: <http://www.smartdraw.com/resources/tutorials>.
11. IDEF0 Function Modeling Method. [Электронный ресурс]. – Режим доступа: <http://www.idef.com/IDEF0.htm>
12. IDEF3 Process Description Capture Methodod. [Электронный ресурс]. – Режим доступа: <http://www.idef.com/IDEF3.htm>
13. Using Rational RequisitePro. [Электронный ресурс]. – режим доступа <http://pic.dhe.ibm.com/infocenter/rqmhelp/v2r0>.
14. *Ноэл М.* Microsoft SharePoint 2010. Полное руководство / М. Ноэл, К. Спенс. – М.: «Вильямс», 2011. – 800 с.
15. <https://habr.com/ru/post/245625/>
16. <https://evergreens.com.ua/ru/articles/checklist-tz.html>
17. http://foranalysts.blogspot.com/2011/08/blog-post_17.html
18. <https://www.youtube.com/watch?v=CiIUNg49fgw>
19. <https://www.youtube.com/watch?v=c8bmvwWx80Q>

20. <https://habr.com/ru/post/215837/>
21. <https://dou.ua/lenta/articles/use-cases/>
22. <https://systems.education/use-case/>
23. <https://blog.business-analyst.info/2009/12/25/use-cases-for-use-cases/>
24. http://www.dsc.ufcg.edu.br/~garcia/cursos/ger_processos/seminarios/Crystal/Use%20Cases,%20Ten%20Years%20Later.htm
25. <https://habr.com/ru/post/231961/>
26. <https://www.youtube.com/watch?v=XR33OQUv17U>
27. <https://www.youtube.com/watch?v=ZQI0vrB7Fok>
28. Марка Д., МакГоуен К. Методологія структурного аналізу і проектування. - М.: МетаТехнологія, 1993.
29. ГОСТ 34.601-90. Інформаційна технологія. Автоматизовані системи. Стадії створення.
30. Білі сторінки MSF. <http://www.microsoft.com/rus/msdn/msf>

ДОДАТКИ

ДОДАТОК 1

ВАРІАНТИ ЗАВДАНЬ

Номер варіанта	Назва системи
1.	Автоматизована система кадрового обліку працівників аеропорту.
2.	Автоматизована система обліку аеропортів України.
3.	Автоматизована система продажу авіаквитків.
4.	Автоматизована система керування повітряним рухом.
5.	Автоматизована система державного реєстру повітряних суден.
6.	Автоматизована система довідкової служби залізничного вокзалу.
7.	Автоматизована система обліку робочого часу працівників підприємства
8.	Автоматизована система керування авіаційним тренажером.
9.	Автоматизована система складського обліку
10.	Автоматизована система управління світлосигнальними системами посадки. (наприклад, <i>OBI-1</i> ”)
11.	Система обліку роботи митниці аеропорту (вантажні перевезення).
12.	Автоматизована система бюро працевлаштування.
13.	Автоматизована система документообігу організації.
14.	Автоматизована контрольно-пропускна система реєстрації працівників організації.
15.	Автоматизована система служби охорони організації.
16.	Автоматизована система дистанційного навчання.
17.	Автоматизована система збору та зберігання інформації про ландшафти.
18.	Автоматизована система складання розкладу занять ВНЗ.
19.	Автоматизована система житлоуправління з обліку мешканців та їх розрахунків.
20.	Автоматизована система формування платежів телефонної компанії.
21.	Автоматизована банківська система безготівкових електронних платежів.
22.	Автоматизована система агентства з нерухомості.
23.	Автоматизована система аптечної мережі міста.
24.	Автоматизована система продаж торговельного центру (супермаркету).
25.	Автоматизована система Інтернет –магазину.
26.	Програмне забезпечення серверу електронної пошти.
27.	Автоматизована система кур'єрської поштової служби доставки рекламної інформації.
28.	Автоматизована система обліку пацієнтів лікарні.
29.	Програмне забезпечення системи керування мобільним телефоном.
30.	Автоматизована розрахункова система оператора мобільного зв'язку.
31.	Автоматизована система перевірки руху маршрутів міського транспорту.
32.	Автоматизована система продажу залізничних квитків.
33.	Автоматизована система оновлення програмного забезпечення організації (через Web-сайти фірм розробників програмного забезпечення)

34.	Автоматизована система обліку транспортних засобів у ДАІ.
35.	Автоматизована система приймальної комісії ВНЗ.
36.	Автоматизована інформаційна система деканату ВНЗ.
37.	Автоматизована система станції технічного обслуговування автомобілів.
38.	Автоматизована система бібліотеки.
39.	Автоматизована система «Розумний дім»
40.	Автоматизована система бюро знахідок
41.	Автоматизована система шлюбної сумісності
42.	Автоматизована система утилізації відходів
43.	Автоматизована система розрахунку оптимальної доставки вантажу різними видами транспорту
44.	Автоматизована система кошторисно-облікової вартості будівництва
45.	Автоматизована система вибору сільськогосподарської культури

ДОКУМЕНТ-КОНЦЕПЦІЯ

1. Загальна інформація

1.1. Назва системи

1.2. Реквізити замовника

1.3. Посилання на попередню документацію

1.4. Планові терміни початку і закінчення робіт

1.5. Фінансові обмеження

2. Цілі системи

2.1. Цілі системи (опис властивостей об'єкту автоматизації, які повинні бути досягнуті в результаті)

2.2. Задачі системи (опис діяльності, націленої на отримання бажаного результату)

3. Характеристика об'єктів автоматизації

3.1. Коротка інформація про об'єкт автоматизації

3.2. Інформація про умови використання та середовище об'єкту

4. Первинний опис функціональності системи

5. Опис потенційного підходу