

**Міністерство освіти і науки України  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра спеціалізованих комп'ютерних систем**

## **ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ**

з дисципліни

### **“Основи системної інженерії”**

для студентів освітньо-кваліфікаційного рівня «бакалавр»  
напряму підготовки 6050201 – «Системна інженерія»

**ТЕРНОПІЛЬ**

Опорний конспект лекцій з дисципліни **“Основи системної інженерії”** для студентів ОКР «бакалавр», 6050201 – «Системна інженерія» / Укл.: Возна Н.Я., Гуменний П.В.– Тернопіль: Гал-друк, 2015. – 118 с.

Методичні рекомендації до виконання лабораторних робіт складаються з частин, що рекомендовані програмою на основі галузевого стандарту вищої освіти України з напряму підготовки 6050201 – «Системна інженерія»

Укладачі:                   Возна Наталія Ярославівна к.т.н., доцент  
                                  Гуменний Петро Володимирович, к.т.н, викладач

Рецензенти:               Франко Ю. П. к.т.н., доцент кафедри комп’ютерних технологій Тернопільського національного педагогічного університету

                                  Якименко І.З., к.т.н., доцент кафедри комп’ютерної інженерії Тернопільського національного економічного університету

Затверджено на засіданні кафедри спеціалізованих комп’ютерних систем протокол №2 від 17.09.2015.

Розглянуто та схвалено науково-методичною комісією з системної інженерії протокол №2 від 17.09.2015.

Розглянуто та схвалено науково-методичною радою факультету комп’ютерних інформаційних технологій, протокол №1 від 29.09.2015 р.

**Міністерство освіти і науки України  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра спеціалізованих комп'ютерних систем**

## **ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ**

з дисципліни

### **“Основи системної інженерії”**

для студентів освітньо-кваліфікаційного рівня «бакалавр»  
напряму підготовки 6050201 – «Системна інженерія»

**ТЕРНОПІЛЬ  
2015**

## ЗМІСТ

ВСТУП.....	5
Лекцій 1. Вступ в системну інженерію.....	7
Лекція 2. Ландшафт системної інженерії.....	16
Лекція 3. Структура складних систем.....	25
Лекція 4. Процес розробки систем.....	37
Лекція 5. Управління системною інженерією.....	48
Лекція 6. Аналіз потреб.....	59
Лекція 7. Дослідження концепції. Концептуалізація.....	65
Лекція 8. Визначення концепції системи.....	74
Лекція 9. Підтримка та аналіз рішення.....	89
Лекція 10. Прогресивна розробка.....	103
Лекція 11. Програмна системна інженерія. Технічне проектування.....	109
Список використаних джерел.....	117

## ВСТУП

Інтелектуальні пристрої - невід'ємна частина нашого життя. З їх допомогою ми відстежуємо поштові відправлення, управляємо світлофорами, літаками і знаходимо потрібний маршрут. Це ключовий елемент систем і послуг, якими ми користуємося щодня, - від "розумних" телефонів і автомобілів, до технологій, застосовуваних у медицині, аерокосмічній та оборонній промисловості.

Інтелектуальні, технічно оснащені і взаємопов'язані пристрої виводять нас на принципово новий рівень взаємодії один з одним і навколишнім середовищем. Різні комбінації елементів електроніки, датчиків, програмного і апаратного забезпечення допомагають нам створювати все нові багатофункціональні пристрої, що враховують потреби кожного з нас. А безмежні фантазія і винахідливість дозволяють нам розробляти сотні інноваційних, спеціалізованих і персоніфікованих систем і послуг.

Зростання масштабів і ускладнення способів організації діяльності по створенню інженерних об'єктів, підвищення ступеня відповідальності за її результати, швидке зростання складності виникають при цьому наукових, технічних і управлінських проблем призвели до появи в середині ХХ століття нової прикладної системної методології - системної інженерії (Systems Engineering).

При розробці інтелектуальних пристроїв головне завдання - це правильна організація: ми повинні максимально ефективно інтегрувати складні комбінації різних технологій, щоб реалізувати інтелектуальну і комплексну "систему систем", яка буде відповідати поставленим цілям. Все це стає можливим завдяки такому поняттю, як системна інженерія.

Системна інженерія, включаючи інженерію програмних систем, представляє сьогодні швидко розвивається прикладну наукову дисципліну. Роботи в цій області виконуються за підтримки цілого ряду великих міжнародних професійних організацій, серед яких Інститут інженерів електротехніки та електроніки (The Institute of Electrical and Electronics Engineers - IEEE), Міжнародна рада з системної інженерії (International Council on Systems Engineering - INCOSE), Рада університетів, що реалізують освітні та дослідні програми в області створення інженерних систем (Council of Engineering Systems Universities - CESUN), група з управління об'єктами (Object Management Group - OMG) і ряд інших.

Вражаючі перетворення, що відбуваються сьогодні в області створення складних інженерних об'єктів і зумовлені революцією у сфері інформатизації, глобалізацією систем і швидким впровадженням інновацій; поява нових класів інженерно-насичених систем, включаючи соціотехнічні системи, транспортні, оборонні та комунікаційні системи масштаба країни, а також розвиток мегасистем привели і в нашій країні до розуміння необхідності проведення робіт і підготовки кадрів в області системної інженерії.

У сучасних розробках зарубіжних фахівців системна інженерія розглядається як комплексний, мультидисциплінарний підхід і методика створення складних систем і визнається в якості фундаменту, на основі якого

можна забезпечити і гарантовано підтримувати надійний і стійкий зв'язок між місією, стратегічними цілями, конкретними завданнями і вимірними результатами інженерної діяльності. Недарма один з видних зарубіжних фахівців з системної інженерії Дерек Хітчінс (Derek K. Hitchins) назвав системну інженерію системною методологією XXI століття.

В історії розвитку системної інженерії можна виділити два великих етапи. Етап становлення, зайнятий формуванням ядра методології і основоположних практик системної інженерії, почався на рубежі 40-х і 50-х років і завершився до середини 90-х років XX століття. Основним результатом цього періоду можна вважати створення науково-методичних і нормативно-технічних основ проектування і розробки складних інженерно-технічних об'єктів. За етапом становлення слідував період, який можна назвати етапом координації. Він триває до сьогоднішнього дня, його характерною особливістю є підвищена увага до зв'язків і гармонізації положень системної інженерії з досягненнями і рекомендаціями, отриманими у сфері управління якістю, управління проектами, програмної інженерії та в інших областях. На сучасному етапі розвитку системної інженерії в центрі уваги системних інженерів виявляються не стільки класичні, в основному апаратні системи з зосередженими параметрами, скільки розподілені системи, насичені різноманітним програмним забезпеченням, а також інженерія великомасштабних програмних продуктів. Відмінна риса цього концепту полягає в тому, що при викладі матеріалу обрано підхід життєвого циклу, з позицій якого послідовно розглянуті всі аспекти діяльності системного інженера. Стадії та етапи життєвого циклу, характерні для цієї моделі, стали по суті основою, на який будується все інше. Але при цьому в концепті вказується на те, що системний інженер зобов'язаний вибрати або побудувати самостійно визначену модель життєвого циклу, яка краще за інших відповідає особливостям предметної області (авіабудування, енергетика, будівництво, військова справа і т. п.).

Характерна особливість роботи по створенню складних систем полягає в тому, що існує безліч шляхів, які можуть, як здається, привести до успіху, але вибір правильного шляху - дуже непросте завдання. Даний концепт допоможе студентам і фахівцям із системної інженерії відповісти на питання, що таке правильний шлях, як його знайти системному інженерові і яких кроків треба дотримуватись, щоб з цього шляху не збитися.

## Лекція 1. ВСТУП В СИСТЕМНУ ІНЖЕНЕРІЮ

1. Що таке системна інженерія?
2. Системна інженерія і традиційні інженерні дисципліни.
3. Системна інженерія і управління проектом.
4. Походження системної інженерії.

**1. Що таке системна інженерія?** Призначення системної інженерії полягає в тому, щоб *керувати створенням складних систем*. Слово «керувати» означає «очолювати, управляти або направляти, зазвичай на основі багатого досвіду по запровадженню в життя вибраного курсу», а також «показувати шлях». Тут підкреслюється ідея вибору одного з безлічі можливих шляхів, по яких підуть інші люди, - це і є першочергове завдання системної інженерії. У тлумачному словнику інженерія визначається як «застосування наукових принципів до практичних завдань, наприклад: проектування, зведення і експлуатація ефективних і економічних конструкцій, устаткування і систем». У цьому визначенні слова «ефективний» і «економічний» - ознаки хорошої системної інженерії.

Слово «система», як і багато поширених, загальновідомих слів в англійській мові, трактується дуже широко. Часто система визначається як *«сукупність взаємозв'язаних компонентів, які працюють спільно для досягнення загальної мети»*. Це визначення має на увазі наявність декількох взаємодіючих частин, які спільно виконують якусь значущу функцію. Слово «складний» обмежує це визначення системами, у складі яких є різноманітні елементи, зв'язані між собою нетривіальним чином. Так, побутовий прилад, наприклад пральну машину, не слід вважати достатньо складним пристроєм, що складається з неоднорідних елементів, для створення якого необхідна системна інженерія, не дивлячись на те що в склад пральної машини входять сучасні елементи автоматики. З іншого боку, в розряд систем, створення яких пов'язане з *інженерною діяльністю*, не потрапляють такі складні об'єкти, як живі організми і екосистеми. Обмеження терміну «система» лише складними системами, створення яких вимагає серйозних інженерних зусиль, дозволяє підкреслити його застосовність до функції системної інженерії в загальноприйнятому сенсі. У наступному розділі перераховані приклади систем, для розробки яких необхідна системна інженерія.

**2. Системна інженерія і традиційні інженерні дисципліни** З приведеного вище визначення виходить, що системна інженерія відрізняється від механіки, електротехніки і інших інженерних дисциплін в декількох істотних відносинах:

1. В центрі уваги системній інженерії знаходиться система в цілому, це надає особливе значення функціонуванню системи як єдиного цілого. Системний інженер дивиться на систему як зовні (тобто оцінює її у взаємодії з іншими системами і оточенням), так і зсередини. Він цікавиться не тільки технічним проектом системи, але і зовнішніми чинниками, які можуть накладати істотні обмеження на проектні рішення. Сюди входить виявлення потреб

замовника, визначення умов експлуатації, сполучення з іншими системами, логістичні вимоги, вимоги до кваліфікації обслуговуючого персоналу і інші чинники, які повинні бути належним чином відбиті в документах, що містять вимоги до системи, і враховані при її проектуванні.

2. Хоча основна мета системної інженерії полягає в тому, щоб керувати, це не означає, що системні інженери самі не приймають істотної участі в проектуванні системи. Навпаки, вони грають провідну роль на початковій стадії створення нової системи (розробка концепції), яка завершується функціональним проектуванням системи, здатної задовольнити потреби користувача. Важливі проектні рішення, що приймаються на даній стадії, не можуть в тій же мірі, як це прийнято в традиційних інженерних дисциплінах, спиратися на кількісне знання, але швидше повинні ґрунтуватися на якісних оцінках, що враховують необхідність досягнення балансу між безліччю різнорідних показників, а також досвід, накопичений в різноманітних дисциплінах, особливо якщо мова йде про новій технології.
3. Системна інженерія наводить мости між традиційними інженерними дисциплінами. Елементи складної системи гетерогенні, тому для її проектування і розробки потрібне знання різних інженерних дисциплін. Щоб система функціонувала правильно, кожен її елемент повинен коректно працювати в зв'язці з іншими елементами - одним або декількома. Реалізація таких взаємозв'язаних функцій залежить від складного комплексу фізичних і функціональних взаємодій між елементами, які проектувалися окремо. Таким чином, елементи системи не можна розробити незалежно, а потім просто з'єднати, сподіваючись отримати при цьому працездатну систему. Насправді системний інженер повинен направляти і координувати проектування окремих елементів, стежачи за тим, щоб взаємодії і сполучення між елементами системи забезпечували сумісність і взаємну підтримку пристроїв у складі системи. Така координація особливо важлива, коли окремі елементи системи проектуються, випробовуються і поставляються різними організаціями.

**3. Системна інженерія і управління проектом.** Розробка нової складної системи зазвичай починається із стадії дослідження. На цій стадії з метою задоволення виявленої потреби або використання технологічної можливості, що відкрилася, розробляється концепція нової системи. Після того, як ухвалено рішення про втілення нової концепції в працюючу систему, настає черга крупних заходів, в яких, як правило, задіюється безліч людей з різними знаннями і навиками. З моменту розробки концепції до введення системи в експлуатацію деколи проходять роки.

Масштаб і складність завдань, що виникають при створенні нової системи, вимагають формування спеціальної групи для управління роботою і координації зусиль. Таке підприємство називається «проектом»; на чолі його керівник проекту, в підпорядкуванні якого знаходиться проектна група. Системна інженерія є невід'ємною частиною управління проектом - тією частиною, яка пов'язана з керівництвом власне інженерними роботами: визначенням цілей, контролем виконання, оцінкою результатів і вживанням



необхідних заходів, що коректують, для того, щоб проект розвивався за планом. Управління, пов'язане з плануванням і контролем фінансових і договірних зобов'язань, а також взаємин із замовниками, хоча і підтримується системною інженерією, але зазвичай не вважається її функцією. Ця тема детально розглядається в розділі 5.

Визнання важливості системної інженерії всіма учасниками проекту розробки системи - необхідна умова його ефективної реалізації. Для досягнення цієї мети часто корисно призначити керівника групи системної інженерії на посаду, що має на увазі відповідальність за технічні аспекти і певні владні повноваження в проекті.

**4. Походження системної інженерії.** Розробка високоманеврених літаків, військових радарів, дистанційних детонаторів, ракет і особливо атомної бомби зажадали революційних проривів в області енергетики, матеріалознавства і інформаційних технологій. Ці системи були складні, поєднували в собі досягнення декількох технічних дисциплін, і при їх розробці виникли інженерні завдання куди складніші, ніж ті, з якими доводилося стикатися раніше. До того ж гранично стислі терміни, зумовили необхідність такого рівня організації і ефективності, який зажадав нових підходів до планування програм і проектів, технічної координації і управління інженерною діяльністю. Системна інженерія в тому вигляді, в якому ми її сьогодні знаємо, стала відповіддю на ці виклики.

За часів «холодної війни» в період 50-х, 60-х і 70-х років вимоги військових продовжували стимулювати технологічний прогрес в області реактивних двигунів, систем управління і розробки нових матеріалів. Проте значніший вплив на розвиток технологій надав, мабуть, інший напрям - напівпровідникова електроніка. Саме завдяки ній став можливий що триває і понині «століття інформації», коли обчислювальна техніка, мережі і засоби зв'язку дозволили розширити можливості і сферу використання систем далеко за раніше мислимі межі.

В зв'язку з цим особливої значущості набуває створення цифрового комп'ютера і керівництва програмним забезпеченням, завдяки чому на зміну ручному почало приходити автоматичне управління системами. Комп'ютерне управління дозволяє якісно підвищити складність систем і представляє особливий інтерес для системної інженерії.

Зв'язок сучасної системної інженерії з її витокami найпростіше зрозуміти, взявши до уваги три основні чинники:

1. **Технічний прогрес**, який дає шанси для підвищення функціональних і інших можливостей системи, але одночасно приводить до появи ризиків, що відносяться безпосередньо до розробки, для управління якими потрібне керівництво з боку системного інженера; ні у якій іншій області це не виявляється так наочно, як у сфері автоматизації. Технологічні досягнення в області людино-машинної взаємодії, робототехнічних пристроїв і програмного забезпечення роблять саме цю область одним з місць, де з найбільшою швидкістю зростає вплив технологій на системне проектування.
2. **Конкуренція**, різні форми якої вимагають пошуку кращих (а значить і більш

передових) системних рішень, для чого необхідно на різних рівнях системної ієрархії шукати компроміси між альтернативними підходами.

3. **Спеціалізація**, яка вимагає розбиття системи на складові частини, відповідні конкретним типам виробів, які можуть бути спроектовані і виготовлені фахівцями, а також наявності строго встановлених правил сполучення цих частин і взаємодії між ними.

Приведене вище широке визначення системи як **сукупності взаємозв'язаних компонентів**, які працюють **спільно** як єдине ціле для досягнення загальної мети, підходить і для більшості побутових приладів. Пральна машина складається з головного барабана для завантаження білизни, електромотора, мішалки, насоса, таймера, внутрішнього барабана, що обертається, і різних клапанів, датчиків і органів управління. Вона виконує послідовність операцій, що хронометруються, і різні допоміжні функції, залежні від тривалості прання і режиму роботи, заданого оператором. Холодильник, мікрохвильова пічка, посудомийна машина, пилосос і радіоприймач - всі вони виконують корисні операції систематичним чином. Проте при створенні цих пристроїв використовуються одна-дві інженерні дисципліни, а їх конструкція визначається давно сталою технологією. Тому вони не відповідають критерію **складності**, і ми не розглядатимемо розробку нової моделі пральної машини або холодильника як завдання, для вирішення якої необхідно привертати системну інженерію в тому сенсі, в якому ми її розуміємо - хоча, звичайно, зусилля інженерів буде потрібно для забезпечення надійності і управління витратами. Зрозуміло, в побутових приладах все ширше застосовуються інтелектуальні автоматичні пристрої на основі нових мікропроцесорів, але подібні пристрої зазвичай автономні, а їх функціональні можливості, як правило, надмірні і не повною мірою використовуються для реалізації основної функції приладу.

Оскільки розробка нових сучасних систем в значній мірі стимулюється зміною технологій, ми додамо у визначення системи, створення якої вимагає участі системного інженера, ще одну характеристику, а саме: хоч би в деяких з ключових елементів застосовується передова технологія.

Система, для розробки, випробування і застосування якої потрібні системний інженер і системна інженерія, характеризується наступними ознаками:

- ♦ інженерна насиченість виробу і, як наслідок, можливість задоволення на цій основі певної потреби;
- ♦ гетерогенність, тобто система складається з різнотипних, різнорідних компонентів з нетривіальними взаємозв'язками і, як наслідок, її створення ведеться з використанням мульти-дисциплінарного підходу, а сама система відносно складна;
- ♦ використання передових технологій так, що саме ці технології життєво необхідні для досягнення найважливіших функціональних можливостей системи і, як наслідок, її створення зв'язане з ризиком і часто обходиться відносно дорого.

Тут і далі під **інженерною насиченою** системою\*, ми розумітимемо суть, що володіє трьома перерахованими вище властивостями, а саме: інженерною

насиченістю, гетерогенністю і використанням для її створення передових технологій<sup>1</sup>. Зрозуміло, ці ознаки повинні розглядатися тільки як доповнення до приведеного раніше широкому визначенню системи. В сукупності вони дозволяють виділити такі системи, при проектуванні, розробці, комплексированні, випробуванні і атестації яких необхідна участь системного інженера. Система складається з безлічі елементів, причому деякі з них самі можуть бути складними і розглядатися як самостійні системи. Наприклад, автоматична телефонна станція може вважатися системою, а вся телефонна мережа - «системою систем». У тій мірі, в якій подібні питання важливі для розуміння системної інженерії, вони обговорюватимуться в розділах 2 і 4.

### **Приклади систем, що потребують системного інженера**

До інженерних насичених комплексних систем належать, зокрема:

- метеорологічні супутники;
- системи управління повітряним рухом в зоні аеропорту;
- системи стеження за вантажівками;
- системи бронювання авіаквитків;
- медичні інформаційні системи;
- пасажирський літак;
- сучасний прибиральний комбайн;
- нафтопереробний завод;
- автоскладальний завод;
- електростанція.

**Приклад: сучасний автомобіль.** Прикладом більш простій і знайомої системи, що відповідає, проте, критерію інженерної насиченості, є пасажирський автомобіль в повній комплектації. Можна вважати, що це молодший представник складніших транспортних систем. Він складається з великого числа різнорідних компонентів, і для створення автомобіля потрібно використовувати досягнення різних областей техніки і технологій. Для правильного функціонування автомобіля його компоненти винні безпомилково і ефективно працювати спільно. Хоча принципи дії автомобіля давно і добре відомі, сучасні автомобілі необхідно проектувати, так щоб вони працювали ефективно в умовах ретельного контролю вихлопних газів, а для цього потрібні складні сучасні датчики і керовані комп'ютером механізми уприскування легко-паливної суміші. Ще один приклад ретельно настроєної автомобільної підсистеми - автоматична анти-блокіровочна гальмівна підсистема. Для захисту пасажирів, круїз-контроля автоматичної навігації, автономного водіння і парковки застосовуються передові матеріали і комп'ютерні технології. Строгі вимоги, що пред'являються до вартості, надійності, функціональності, комфортабельності, безпеки і десятка інших параметрів, ставлять перед системним інженером цілий ряд непростих проблем. Таким чином, автомобіль відповідає даному раніше визначенню системи, що потребує системного інженера, а значить, може служити корисним прикладом. Автомобіль також є прикладом широкого класу систем, в яких потрібна активна взаємодія з оператором (управління). У тому або іншому ступені така взаємодія має місце

---

\* комплексна система

в будь-якій системі, але в даному випадку необхідне безперервне управління. Оператор (водій) в як там не є найпрямішому сенсі виступає в ролі невід'ємної частини автомобільної системи, здійснюючи функцію зворотного зв'язку - він рулює, тобто виявляє і виправляє відхилення автомобіля від бажаної траєкторії на дорозі. Тому в проекті повинні бути враховані як критичне обмеження можливості сприйняття і швидкість реакції оператора. Ці обмеження слід брати до уваги разом з іншими обмеженнями, обумовленими особливостями людино-машинної взаємодії, зокрема розташуванням органів управління і індикаторів, положенням водійського крісла і так далі Крім того, хоча пасажир, можливо, і не є невід'ємною частиною системи рульового управління автомобілем, інтерфейси (наприклад, вага, зручність крісел і огляду, безпека), що відносяться до них, також повинні бути враховані в процесі проектування. Не дивлячись на те що автомобілі збираються і поставляються без людського елемента, з погляду системної інженерії людей необхідно враховувати як повноправні елементи системи.

У міру того як складні системи все глибше проникають в сучасне суспільство, а роль системної інженерії в їх розробці росте, професія системного інженера отримує все більш широке визнання. І перш за все це відноситься до компаній, що спеціалізуються на розробці крупних систем. У багатьох таких компаніях створені відділи системної інженерії, а люди, що працюють в них, іменуються системними інженерами. Крім того, для вирішення глобальних проблем, що виникають в таких областях, як охорона здоров'я, зв'язок, охорона навколишнього середовища і багато інших, для отримання життєздатного рішення теж потрібно привертати методи системної інженерії.

Недостатньо швидке визнання системної інженерії як професії пояснюється тим, що їй не відповідає ніяка інженерна дисципліна, що традиційно викладається. У основі традиційних інженерних дисциплін лежать кількісні співвідношення, відомі фізичні закони і що піддаються вимірюванню властивості матеріалів, енергії або інформації. З іншого боку, системна інженерія має справу переважно із завданнями, які вивчені не повністю, для яких не існує відомих рівнянь, що зв'язують змінні; завданнями, при вирішенні яких потрібно відшукати баланс між конфліктуєчими цілями з несумірними атрибутами. У минулому відсутність кількісних знань перешкоджала становленню системної інженерії як самостійної дисципліни.

Не дивлячись на всі перепони, потреба в системній інженерії, усвідомлена промисловістю і державою, стимулювала розробку ряду академічних програм для отримання магістерського і докторського ступеня по системній інженерії. Росте також число університетів, де можна отримати ступінь бакалавра по цій дисципліні.

Визнання системної інженерії як професії привело до утворення професійного суспільства - Міжнародної ради з системної інженерії (International Council on Systems Engineering - INCOSE), однією з головних цілей якого, є пропаганда системної інженерії і визнання її як сфери професійної діяльності.

## Орієнтація технічних фахівців

Відношення системних інженерів до технічних дисциплін буде простіше зрозуміти, якщо взяти до уваги, що технічні фахівці істотно розрізняються не тільки за своїми професіями, але і за інтелектуальними цілями, інтересами і схильностями. Типовий учений прагне зрозуміти природу і поведінку матеріального світу. Учений ставить собі питання «Чому?» і «Як?». Математик зазвичай займається виведенням логічних послідовностей з набору припущень, які можуть ніяк не співставлятися з реальним світом. Для математика представляє інтерес вислів: «якщо А, то Б». Інженера цікавить в першу чергу створення корисного продукту.

Ці орієнтації принципово різні - і тут криється відповідь на питання, чому технічні фахівці зосереджують увагу на тих або інших аспектах науки і техніки. Проте у більшості фахівців професійна орієнтація не задана раз і назавжди: нерідко ученому доводиться ставати інженером, щоб сконструювати якийсь апарат, а інженерові необхідно застосовувати математику для вирішення виниклих завдань управління. Тому в загальному випадку модель орієнтації технічного фахівця можна умовно представити у вигляді суми трьох ортогональних векторів, що відображають схильність людини до наукової роботи, математики або інженерної діяльності.

Після того, як студенти закінчують вуз, тенденція до розділення по спеціальностях і інтересах зазвичай посилюється, тому що вчорашні випускники прагнуть до визнання в своїй наочній області. Більшість технічних фахівців опираються перетворенню на фахівців широкого профілю, тому що побоюються втратити лідируюче положення на професійному терені і не отримати відповідного визнання.

Така вузька спеціалізація заважає спілкуванню професіоналів один з одним; мовний бар'єр - одна з серйозних перешкод, але відмінності в базових цілях і способах мислення є ще глибшою проблемою. Ось і виходить, що вирішення складних завдань на стику декількох дисциплін залежить від осіб, що порівняно рідко зустрічаються, які з тієї або іншої причини після твердження в основній професії зацікавилися системними проблемами і навчилися працювати спільно з фахівцями з інших областей.

В процесі розробки системи системні інженери мають справу з найбільш важливими питаннями. Вони від початку до кінця проектують архітектуру системи, вибирають технічні підходи і керують іншими членами колективу, зайнятими проектуванням компонентів. Спільно із замовником вони ранжирують вимоги до системи, добиваючись того, щоб різні її властивості належним чином враховувалися при урівноваженні різноманітних технічних чинників. Вони вирішують, які ризики прийнятні, а які - ні, і як застрахуватися від ризиків, щоб привести програму до успішного завершення.

Саме системні інженери прокладають курс програмі розробки, в якій чітко визначені види і терміни проведення випробувань і імітаційного моделювання. За ними залишається останнє слово в рішенні питання про те, як забезпечити потрібні функціональні можливості системи за прийнятною ціною.

Коли в ході розробки виникають несподівані проблеми - а це неминуче, - саме системні інженери вирішують, як з ними справитися. Вони визначають, чи

знадобиться для усунення проблеми абсолютно новий підхід або досить повернути більше сил; чи слід для компенсації труднощів внести зміни до якоїсь, що можливо знаходиться за межами безпосереднього розгляду, частина системи або краще пом'якшити вимоги, із-за яких проблема виникла.

Системний інженер набуває здатності управляти розробкою системи не тому, що посідає свою посаду, а завдяки чудовому знанню системи в цілому, її призначення, особливостей роботи її частин, а також всіх технічних чинників, що підлягають обліку при розробці. Він набуває цієї можливості ще і тому, що зумів на практиці довести своє уміння проводити складні програми крізь лабіринт труднощів до успішного фіналу.

Системному інженерові належить працювати на стику різних дисциплін, і від нього вимагається уміння схопити суть кожній з них. Саме тут здібності до занять наукою і інженерна кмітливість опиняться дуже до речі, тому що таким людям набагато легше дається вивчення нових дисциплін. Справа не в тому, щоб глибоко знати вищу математику, а швидше в тому, що людині із слабкою математичною підготовкою важко буде розбиратися в предметах, що мають в своїй основі математичні концепції.

У системного інженера повинні бути творча жилка і смак до вирішення практичних завдань. Інтерес до роботи повинен переважати над інтересом до кар'єрного зростання.

Системна інженерія - це швидше пошук, чим швидкий шлях до вершини. Нижче перераховані якості, характерні для більшості успішних системних інженерів. Подібні фахівці:

1. Отримують задоволення від вивчення нового і вирішення завдань.
2. Люблять, коли їм кидають виклик.
3. Скептично відносяться до недоведених тверджень.
4. Відкриті для нових ідей.
5. Мають солідну підготовку в природно-наукових і інженерно-технічних науках.
6. Продемонстрували технічні досягнення в певній наочній області.
7. Володіють знаннями в декількох інженерних областях.
8. Швидко засвоюють нові ідеї і інформацію.
9. Володіють хорошими навиками міжособового спілкування і комунікативними здібностями.

Загалом і в цілому системна інженерія - це дисципліна, що володіє високим потенціалом, вимагає мультидисциплінарних знань, що дозволяє розробити різноманітні системні елементи.

Системний інженер повинен уміти виконувати наближені розрахунки стосовно складних випадків, щоб перевірити результат на істинність. І нарешті, він повинен володіти скептичним, але разом з тим позитивним складом розуму - необхідна умова при розробці складних систем.

## **КОРОТКІ ВИСНОВКИ**

### **Що таке системна інженерія?**

Призначення системної інженерії полягає в керівництві розробкою

складних, комплексних систем. Система визначається як сукупність взаємозв'язаних компонентів, які працюють спільно для досягнення загальної мети. Далі, комплексна інженерна насичена система - в тому сенсі, в якому ми розуміємо її в цій книзі - 1) складається з декількох різнорідних, різнотипних елементів, які зв'язані між собою складним чином і 2) вимагає застосування системної інженерії для керівництва розробкою.

Системна інженерія відрізняється від традиційних дисциплін тим, що 1) предметом її розгляду є система в цілому; 2) її цікавлять потреби замовника і умови експлуатації; 3) вона направляє розробку концепції системи; 4) вона наводить мости між традиційними інженерними дисциплінами і долає нерозуміння між окремими фахівцями. Системна інженерія є невід'ємною частиною керівництва проектом, оскільки необхідна для планування і напряму інженерних зусиль.

### **Походження системної інженерії**

Сучасна системна інженерія з'явилася в результаті того, що передові технології у поєднанні із зростанням ступеня автоматизації принесли з собою ризики і підвищення складності; при цьому конкурентна боротьба вимагала йти на ризик після ретельної оцінки можливих наслідків, а поглиблення спеціалізації диктувало необхідність міждисциплінарних зв'язків і побудови інтерфейсів.

### **Приклади систем, що потребують системного інженера**

До інженерних насичених комплексних систем належать, зокрема:

- метеорологічні супутники;
- системи управління повітряним рухом в зоні аеропорту;
- системи стеження за вантажівками;
- системи бронювання авіаквитків;
- медичні інформаційні системи;
- пасажирський літак;
- сучасний прибиральний комбайн;
- нафтопереробний завод;
- автоскладальний завод;
- електростанція.

### **Модель розвитку кар'єри системного інженера**

Професія системного інженера важка, але ці труднощі окупаються. Типовий системний інженер із задоволенням береться за абстрактних, таких, що допускають неоднозначне рішення задачі, і нагородою йому служить усвідомлення своєї ключової ролі в проекті. Отже, успішний системний інженер володіє наступними якостями:

- добре уміє вирішувати завдання і вітає складні проблеми;
- має хорошу технічну підготовку і широкий кругозір;
- володіє аналітичним, системним складом розуму і при цьому проявляє творчі здібності;
- чудово володіє навиками спілкування, природжений лідер.

## Лекція 2. ЛАНДШАФТ СИСТЕМНОЇ ІНЖЕНЕРІЇ

1. Успішні системи.
2. «Якнайкраща» система.
3. Збалансована система.
4. Підходи системної інженерії
5. Сфери діяльності, пов'язані з системною інженерією.

### **1. Успішні системи.**

З самого початку розробки в центрі уваги системного інженера знаходиться успіх системи, тобто задоволення вимог і досягнення поставлених цілей, успішна промислова експлуатація і довгий термін корисної служби. Точка зору системного інженера охоплює всі ці аспекти. Системний інженер намагається за очевидним і лежачим на поверхні розгледіти дійсні потреби користувача, а також зрозуміти, в яких умовах система експлуатуватиметься на практиці. Системний інженер ставить за мету знайти таке технічне рішення, яке спростило б технічне обслуговування і ремонт системи в процесі експлуатації і разом з тим дозволило б провести модернізацію, яка рано чи пізно буде потрібна. Він намагається передбачити проблеми і вирішувати їх на можливо ранішій стадії циклу розробки; а в тих випадках, коли це практично неможливо, системний інженер ратує за підготовку резервних планів для реалізації їх в міру необхідності.

З метою успішної розробки систем в організаціях необхідно послідовно і усвідомлено застосовувати підхід системної інженерії. Це має на увазі реалізацію систематичних, добре формалізованих і старанно контрольованих заходів, що включають ретельне планування, аналіз, критичні огляди і документування. Але не менш важлива та сторона системної інженерії, якій часто приділяють недостатньо уваги, а саме: інновації. Щоб нова складна система була конкурентоздатна в обстановці швидких технологічних змін і зберігала лідируюче положення впродовж багатьох років служби, в її ключових компонентах обов'язково повинні бути реалізовані останні технічні досягнення. Це неминуче спричиняє за собою ризики - як відомі, так і ще невідомі, а значить, буде потрібно додаткові зусилля, щоб довести нові підходи до проектування до стану зрілості і згодом виконати валидацію подібних проектних рішень стосовно компонентів системи. Вибір самих багатообіцяючих технічних рішень, оцінка пов'язаних з ними ризиків, планування вирішальних експериментів і продумування варіантів дій на випадок невдачі - все це важливі обов'язки системного інженера. Таким чином, точка зору системного інженера враховує одночасно як ухвалення ризиків, так і пом'якшення їх наслідків.

### **2. «Якнайкраща» система.**

При обговоренні точки зору системного інженера часто доводиться чути, що «краще - ворог хорошого» і що «системна інженерія - це мистецтво вибору *достатньо хорошого*». Ці твердження можна інтерпретувати помилково,



вважаючи, ніби системна інженерія готова задовольнитися другосортним. На самій же справі все йде інакше: системний інженер прагне побудувати **якнайкращу** систему з можливих, але це часто зовсім не система з якнайкращими функціональними характеристиками. І суперечності тут немає: вся річ у тому, що розуміти під словом «кращий». У приведених вище твердженнях епітети «краще» і «достатньо хороше» відносяться до показників функціонування системи, тоді як з погляду системного інженера ці показники лише елемент безлічі базових характеристик системи; не менш важливі цінова доступність, зручність використання, витрати часу, необхідні на приведення в працездатний стан, простота обслуговування, а також можливість виконання робіт відповідно до узгодженого графіка.

Таким чином, системний інженер шукає **якнайкращий баланс** критично важливих властивостей системи з погляду успішності програми розробки і цінності системи для користувача. Взаємозалежність між показниками функціонування і вартістю можна зрозуміти виходячи із закону убуючої прибутковості.

Припустимо, що прийнятий якийсь технічний підхід до досягнення заданого показника функціонування системи, що розробляється. На рис. 2.1а представлений типовий графік залежності між значенням показника функціонування гіпотетичного компонента системи і витратами на його розробку. Верхня горизонтальна лінія представляє теоретичний максимум значення даного показника при вибраному технічному підході. При витонченішому підході максимум може опинитися вище, але ціною великих витрат. Пунктирні лінії представляють мінімально допустиме і бажане значення параметра.

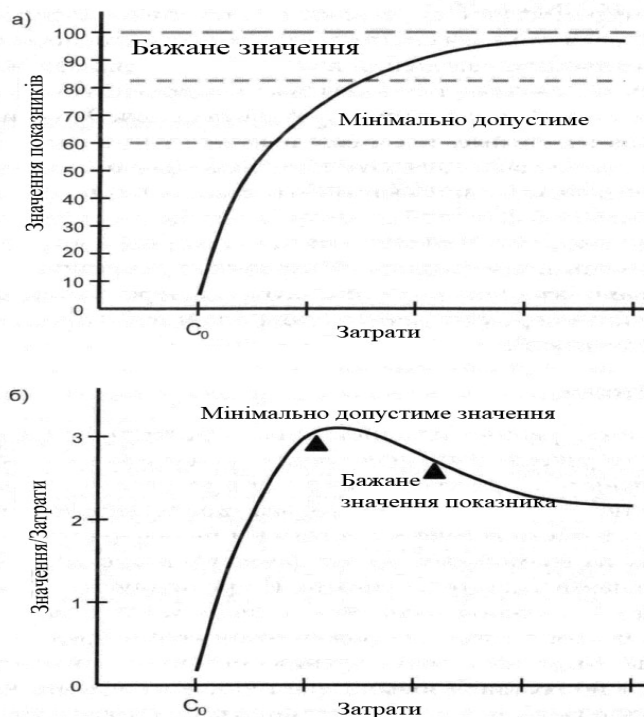


Рис. 1.1. а) Залежність значення показника функціонування від вартості розробки  
 б) Залежність відношення «значення показника/витрати» від об'єму витрат.

Крива на рис. 2.1а починається в точці  $C_0$ , яка характеризує витрати на досягнення хоч якого-небудь прийняттого значення показника функціонування. Спочатку крива круто йде вгору, але у міру асимптотичного наближення до теоретичного максимуму стає пологішою. Крутизна кривої є мірою приросту показника функціонування у відповідь на збільшення витрат. Поступове падіння крутизни обумовлене законом прибутковості, що зменшується, який застосовний практично до будь-якої діяльності, пов'язаної з розробками.

Як приклад розглянемо розробку автомобіля, максимальна швидкість якого буде вища, ніж у прототипу. Очевидне рішення - поставити потужніший двигун. Але, як правило, вагові і габаритні характеристики у такого мотора гірші, а крім того падає паливна ефективність. Крім того, із збільшенням швидкості росте і лобовий опір повітря, для подолання якого потужність двигуна повинна бути збільшена нелінійно. Якщо ставиться вимога залишити витрату палива на колишньому рівні і по можливості не збільшувати розмір і масу автомобіля, то слід подумати про використання або про розробку більш передового двигуна, про поліпшення обтічності корпусу, про застосування спеціальних полегшених матеріалів - і взагалі пошукати способи компенсації небажаних побічних ефектів збільшення швидкості. Все це підвищить вартість модифікованого автомобіля, причому чим ближче конструктори підходять до теоретичних меж вживаних технічних рішень, тим більше виявляється приріст витрат. Тому очевидно, що для досягнення розумного балансу ні для однієї експлуатаційної характеристики не можна вибирати граничне (або близьке до граничного) значення.

Підхід до відшукування такого балансу зображений на рис. 2.1 б, де представлена залежність між відношенням показника функціонування до витрат і власне цими витратами (тобто  $y/x$ ). Відношення значення технічного або функціонального показника до витрат еквівалентно поняттю **економічної ефективності**. Видно, що у кривої є максимум, за яким ефективність падає. Звідси витікає, що значення показника функціонування кращої в цілому системи, швидше за все, буде близьким до того, що досягається в крапці, де відношення показника до витрат досягає максимуму, - за умови що набутого значення буде вище мінімально допустимого.

### 3. Збалансована система

Одне із визначень слова «баланс», особливо доречне в контексті проектування систем, звучить так: «гармонійне або відповідне співвідношення або пропорція частин або елементів деякої конструкції або композиції». Найважливіша функція системної інженерії - знайти баланс між різними компонентами системи, які, як вже наголошувалося вище, проектувалися вузькими наочними фахівцями, що прагнули оптимізувати характеристики конкретних компонентів. Нерідко це завдання виявляється важким, що і продемонстровано на рис.2.2.

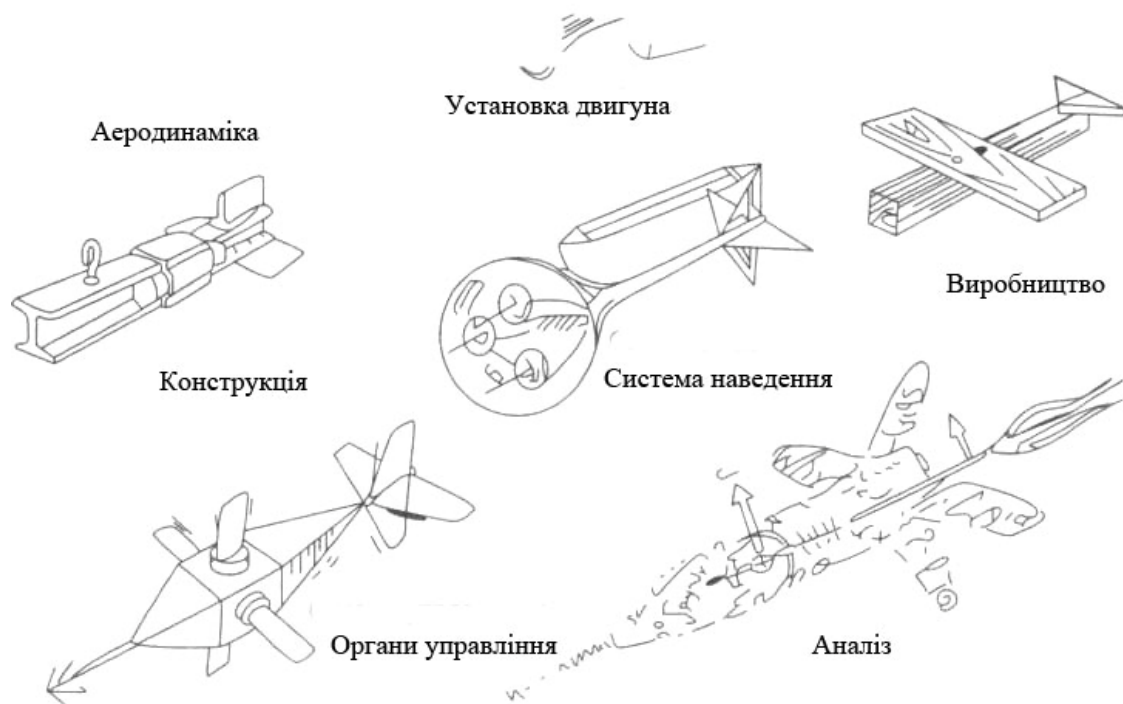


Рис.2.2. Проект ідеальної ракети з точки зору фахівців різних інженерних галузей.

Тут показано, як згідно представленням художника, могла б виглядати керована ракета, якби її проектував фахівець в одній з наочних областей, що мають відношення до ракетобудування. Звичайно, це перебільшення, але малюнки відображають той безперечний факт, що фахівець-наочник прагне оптимізувати саме ті характеристики системи, в яких краще всього розбирається. У загальному випадку слід чекати, що фахівець з проектування хоча і розуміє, що система - це сукупність компонентів, лише *сумісне* функціонування яких дозволяє добитися бажаної її поведінки, в ході розробки все одно приділятиме особливу увагу тим аспектам, які найбільшою мірою відносяться до його компетенції і службових обов'язків.

Навпаки, системний інженер повинен зосередитися на системі в цілому і займатися приватними питаннями тільки в тій мірі, в якій вони можуть вплинути на функціонування всієї системи, на властивих розробці ризики, на витрати або на життєздатність системи в довгостроковій перспективі. Коротше кажучи, системний інженер відповідає за управління розробкою, так щоб кожному компоненту було приділено належну увагу і виділені необхідні ресурси, але при цьому забезпечувалася оптимальна поведінка системи в цілому. Часто такий фахівець виступає в ролі «чесного технічного посередника», який вирішує, на які технічні компроміси в проекті можна піти, щоб добитися задовільної взаємодії ключових елементів системи.

### **Збалансована точка зору**

Отже, системний погляд має на увазі першорядна увага до балансу інтересів, прагнення до того, щоб жоден з показників системи не поліпшувався за рахунок іншого, не менше або навіть важливішого: наприклад, функціональні можливості не нарощувалися ціною збільшення вартості, швидкість - ціною дальності, пропускну спроможність - ціною збільшення

частоти помилок. Оскільки практично всі критично важливі параметри взаємозависимі, належний баланс слід шукати у всіх проектних рішеннях. Зазвичай, як у вищенаведених прикладах, подібні характеристики насилу піддаються порівнянню, тому судити про правильне їх співвідношення може тільки людина, що глибоко розбирається в тому, як система працює. Подібні думки системні інженери повинні виносити щодня, тому вони зобов'язані уміти мислити на рівні, що охоплює всі характеристики системи.

Хоча в останні декілька десятиліть рівень зрілості системної інженерії швидко зростає, в рамках цієї дисципліни існують і існуватимуть і використовуватимуться різні уявлення, підходи і точки зору. Це обумовлено розширенням наших знань про потенціал і користь системних підходів для вирішення все більш складних завдань, що виникають в світі. Про підйом системної інженерії свідчить зростаюча кількість академічних програм і випускників вузів по цьому напрямку. У деяких аналітичних оглядах наголошується, що системну інженерію багато хто вибирає як професію, що обіцяє відмінні кар'єрні перспективи. Працедавці у всіх секторах - як урядових, так і приватних - полюбляють за досвідченими системними інженерами.

Підхід *системної інженерії*, припускає, що в центрі уваги повинні знаходитися наслідки і вирішення проблеми з наміром розробити або побудувати систему, яка допоможе з нею справитися. Цей підхід носить в основному технічний характер і включає досвід потенційних користувачів і розробників системних рішень з метою з'ясування найважливіших потреб та вимог, а також концепції функціонування, лише після цього можна приступати до функціонального і фізичного проектування, розробки проектних специфікацій, виробництва і випробування системи. При цьому увагу слід зосередити на взаємодіях підсистем і необхідності отримання життєздатних і відчутних результатів. На практиці такий підхід застосовний до систем різного рівня складності, але у будь-якому випадку очікується, що кінцевий виріб успішно функціонуватиме в реальних умовах. Придатність підходу системної інженерії до розробки продукції доведена на численних прикладах створення систем комерційного і оборонного призначення.

#### **4. Підходи системної інженерії**

Системну інженерію можна також розглядати у вигляді послідовності процесів і методик, вживаних при проектуванні, розробці, комплектуванні і випробуванні системи (див. приклади на рис.2.3-2.5).

На рис.2.3 як логічний засіб досягнення узгодженості і життєздатності показаний процес лінійного виконання ряду дій, які часто здійснюються ітеративно. Невеликі відмінності характеризують каскадну схему; такий підхід надає додаткові, гнучкіші засоби для показу взаємозв'язків і взаємодії. Наявність великого числа кроків, що повторюються і взаємозалежних, породжує спіральні або циклічні схеми.

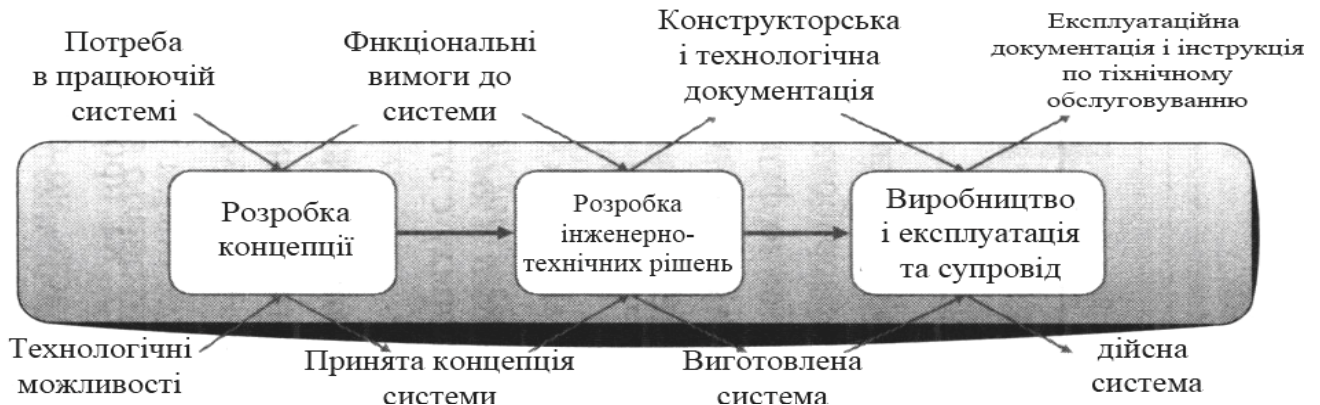


Рис.2.3. Процес лінійного виконання ряду дій.

Популярні в системній інженерії схеми V-подібні (рис.2.4) надають можливість показати життєвий цикл розробки, в якому виразно простежуються зв'язки між вимогами і описом системи, з одного боку, і виготовленим, таким, що пройшов випробування виробом - з іншою.

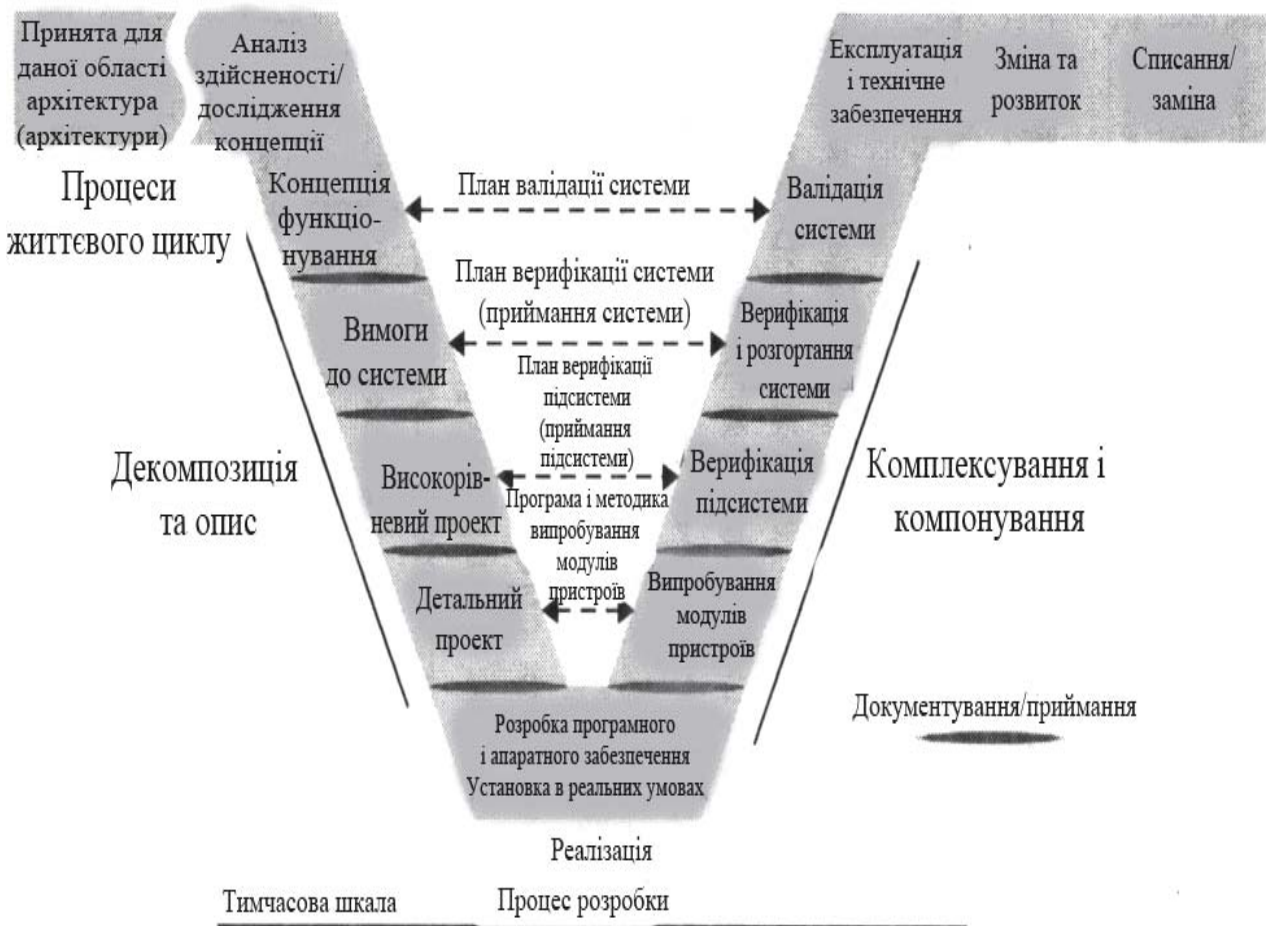


Рис.2.4. V-подібний підхід.

На рис.2.5 детально показаний повний життєвий цикл системи, включаючи управлінські дії на кожній його стадії. Малюнок ілюструє тісні зв'язки між управлінським плануванням і контролем, з одного боку, і процесами системної інженерії - з іншою.

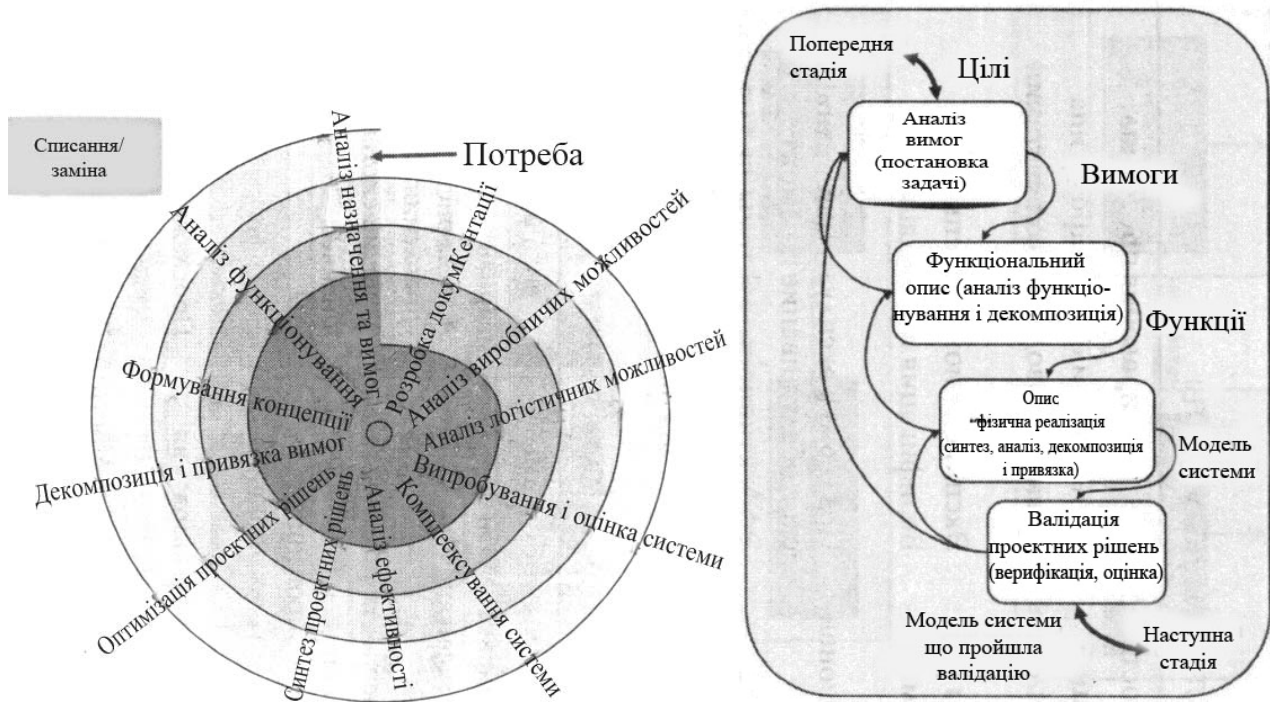


Рис.2.5. Повний життєвий цикл системи.

## 5. Сфери діяльності, пов'язані з системною інженерією.

Оскільки системна інженерія наводить мости між такими традиційними інженерними дисциплінами, як електротехніка, механіка, аеродинаміка, цивільне будівництво і багато інших, то слід чекати, що сприйняття системної інженерії фахівцем, зайнятим певною діяльністю, буде зміщено у бік його спеціальності. Аналогічно, оскільки системна інженерія - це керівництво по проектуванню систем, яке часто використовується в контексті конкретного проекту або програми, керівники різного рівня і профілю схильні рахувати адміністративно-управлінські елементи планування і контролю ключовими аспектами розробки системи. Функції підтримки управлінської діяльності, без якої немислимий успіх системної інженерії, - управління якістю, управління персоналом і фінансовий менеджмент - можуть претендувати на очолюючу, інтеграційну роль в розробці систем.

Ці сприйняття показані на рис.2.6 разом з додатковими областями, відповідними традиційним уявленням про методи і практичне застосування системної інженерії. Прикладом може служити область «Дослідження операцій», яка привносить в системну інженерію засоби, що дозволяють проводити кількісний аналіз альтернативних варіантів і ухвалювати оптимальні рішення. У проектуванні систем беруть також участь фахівці із структур і архітектури. У таких несхожих галузях, як виробничі і автономні системи, можлива ще одна інтерпретація системної інженерії, витікаюча від розробників систем автоматичного управління, які спираються на принципи системної інженерії, що відносяться до управління сполученнями і зворотним зв'язком. Нарешті, перекриття системної інженерії з областю «Математичне і імітаційне моделювання» відповідає уявленню, характерному для дослідження економічної ефективності систем з погляду задоволення потреб і вимог користувачів.

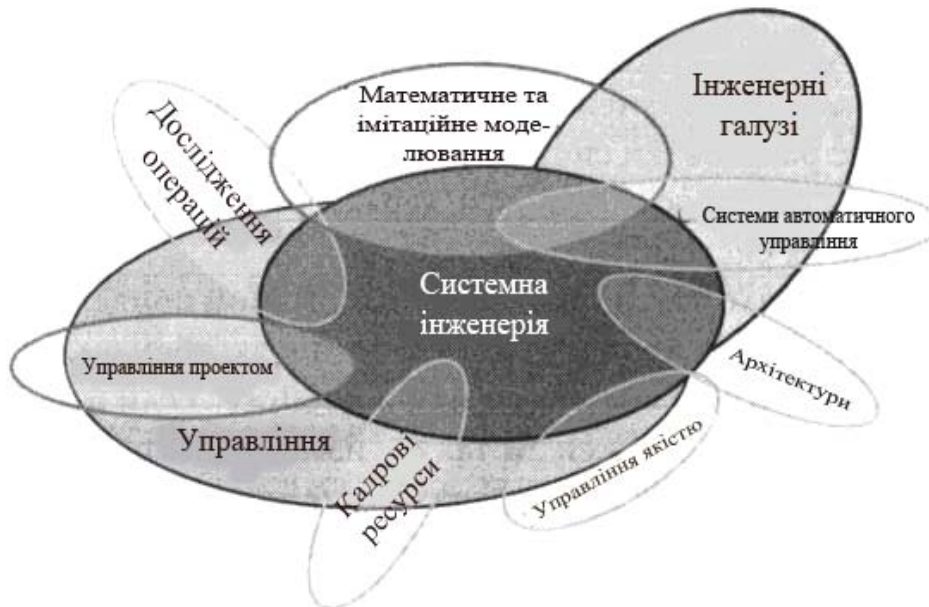


Рис.2.6. Приклади сфер діяльності, пов'язаних з системною інженерією

У міру розвитку системної інженерії і розширення її застосування в різних галузях примножуватимуться і сфери діяльності, пов'язані з системною інженерією.

Впродовж періоду розробки системний інженер концентрує увагу на всій системі, ухвалюючи ті або інші рішення в залежності того, як вони впливають на систему в цілому. Щоб забезпечити таке комплексне рішення, нерідко доводиться наводити мости між різними наочними областями. Співробітник, що спеціалізується, наприклад, на проектуванні, «одновимірний» - він глибоко знає свою технічну область, але йому бракує технічного світогляду і управлінського досвіду. Фахівець з планування і управління «двовимірний» - у нього є багатий досвід керівництва, помірно широкий технічний світогляд і поверхневі знання в конкретних технічних дисциплінах. Системний інженер «тривимірний» - у нього широкий технічний світогляд поєднується з достатньо глибокими технічними знаннями і керівним досвідом.

## **КОРОТКІ ВИСНОВКИ.**

### **Уявлення в системній інженерії**

Для розуміння системної інженерії використовується декілька уявлень: від загальноприйнятого підходу системного мислення до вирішення завдань до підходу, що бере за основу процес розвитку, характерного для системної інженерії і, в ширшій перспективі, - до системоскладання.

### **Наочні області, пов'язані з системами**

Погляд на системну інженерію як на системостворення охоплює не тільки традиційні інженерні дисципліни, але також технічну і управлінську області разом з обліком соціального, політичного/законодавчого і людського чинників. Через свою складність особливий інтерес представляють великомасштабні системи.

## **Сфери діяльності, пов'язані з системною інженерією**

Системна інженерія охоплює або частково зачіпає різні сфери діяльності і галузі знань, серед яких інженерна справа, менеджмент, системний аналіз і дослідження операцій, архітектурне проектування, математичне і імітаційне моделювання і багато інших.

## **Підходи системної інженерії**

У міру становлення системної інженерії і розширення спектру її застосування було розроблено декілька процесних моделей, зокрема лінійна, V-подібна, спіральна і каскадна.

## **Системна інженерія. Дії і результати**

Уявлення про повний життєвий цикл системи допомагає побачити його тісний взаємозв'язок з процесом управління, і виділити широкий спектр різних дій і їх результатів.



## Лекція 3. СТРУКТУРА СКЛАДНИХ СИСТЕМ

1. Складові частини і інтерфейси системи
2. Ієрархія в складних систем.
3. Модель складної системи
4. Елементи системи.
5. Межі системи: контекстна діаграма.
6. Складність в сучасних системах.

### **1. Складові частини і інтерфейси системи.**

Оскільки системному інженерові для розробки складної системи необхідні широкі пізнання в декількох взаємозв'язаних областях, виникає питання, наскільки *глибокі* повинні бути ці знання. Зрозуміло, вони не можуть бути такими ж глибокими, як у фахівця, що все життя працював у відповідній області. Але все таки їх повинно бути досить, щоб усвідомити такі чинники, як програмні і проектні ризики, технологічні обмеження, вимоги, що пред'являються з боку інтерфейсів, а також провести аналіз компромісів при виборі проектних альтернатив.

Очевидно, що відповідь на питання, поставлене вище, залежить від ситуації. Проте можна зробити важливі висновки, дослідивши ієрархічну структуру сучасних систем. Подібне дослідження виявить типові складові частини, які зустрічаються в більшості систем і визначають мінімально придатний для роботи рівень технічних знань, якими повинен володіти системний інженер для того, щоб виконувати свою роботу. На цьому рівні знаходять компроміс відносно технічних можливостей системи і вирішують конфлікти на рівні інтерфейсів, щоб добитися збалансованого проектного рішення відносно системи в цілому. У подальших розділах ми обговоримо природу цих складових частин в контексті їх уявлення як основних елементів системи, а також їх інтерфейси і взаємозв'язки.

### **2. Ієрархія в складних систем.**

Щоб з'ясувати собі, чим займається системна інженерія і що повинен вивчити системний інженер для виконання своїх обов'язків по управлінню розробкою складної системи, необхідно визначити межі і структуру подібної системи в загальному випадку. При цьому визначення *система* по суті своїй застосовано до різних рівнів агрегації складно взаємодіючих елементів. Наприклад, автоматична телефонна станція (АТС), лінії якої тягнуться до обслуговуваних будівель, з повною підставою може бути названа системою. Комутатори в готелях і офісних будівлях зі своїми місцевими лініями можна назвати підсистемами, а телефонні апарати - компонентами системи. В той же час АТС можна розглядати як підсистему системи міського телефонного зв'язку, а останню, у свою чергу, - як підсистему системи національного телефонного зв'язку.

Інший приклад - пасажирський авіалайнер, безумовно, заслуговує назви системи, підсистемами якої є планер, двигуни, органи управління і так далі Але

авіалайнер можна також назвати підсистемою системи повітряного транспорту, яка включає аеродром, авіадиспетчерську службу і інші елементи інфраструктури, в рамках якої функціонує лайнер. Тому часто говорять, що будь-яка система є підсистемою якої-небудь системи більш високого рівня, і, навпаки, всяку підсистему можна розглядати як самостійну систему.

Розглянуті вище зв'язки породили такий термін, як *суперсистема* для позначення систем типу глобальної телефонної мережі і системи повітряного транспорту, що охоплюють. У мережевих військових системах інтегровану систему, що складається з розподілених датчиків і комплексів озброєння, називають *системою систем* (systemofsystems - SOS). Ця термінологія перекочувала і в діловий світ, проте визначення і застосування терміну міняються залежно від галузі і спеціалізації.

### 3. Модель складної системи.

Невизначеність в розумінні того, що є система, може викликати утруднення у студентів, що вивчають основи системної інженерії. Тому для ілюстрації типових обов'язків системного інженера буде корисно створити конкретнішу модель типової системи. З матеріалу, що викладається нижче, стане ясно, що техніка моделювання - один з основних інструментів системної інженерії, особливо за ситуації, коли зібрати факти, що піддаються однозначній інтерпретації і що мають кількісний вираз, не виходить. У нашому випадку ми скористаємося цією технікою, щоб описати логічну структуру моделі типової складної системи в термінах її складових частин. Мета такої моделі - відобразити відносно просту і зрозумілу архітектуру системи, яка може використовуватися як відправна крапка при обговоренні розробки нової системи і ролі системної інженерії впродовж цього процесу. По своїх масштабах ця модель не розповсюджується на суперсистеми або системи систем, але з її допомогою можна описати більшість систем, що розробляються шляхом комплексирования компонентів, що набувають на основі договору або угоди, - таких, наприклад, як новий літак або система управління повітряним рухом в зоні аеропорту.

За своєю природою складна система володіє ієрархічною структурою, в якій можна виділити ряд крупних взаємодіючих елементів, званих, як правило, *підсистемами*. Останні у свою чергу складаються з простіших функціональних об'єктів і т. д., аж до таких примітивних елементів, як шестерня, трансформатор або електрична лампочка, які зазвичай називають *деталлями*.

У число стандартних термінів, вживаних при описі різних архітектурних рівнів в структурі системи, входять лише «система» і «підсистема» для позначення самих верхніх рівнів і «деталь» - для самого нижнього.

По причинах, які стануть зрозумілими з подальшого, в моделі системи використовуються ще два проміжні рівні - *компоненти* і *субкомпоненти*. У деяких моделях проміжних рівнів може бути на один-два більше, але для наших цілей п'яти названих достатньо.

Грунтуючись на описаній вище ієрархічній структурі комплексних систем, можна визначити області компетенції як системного інженера, так і фахівця з

проектування. Системні компоненти проміжного рівня займають центральне місце в процесі розробки системи; вони є елементами, які здебільшого створюються фахівцями з промислового проектування, здатними адаптувати їх для конкретної мети виходячи із заданих технічних вимог. Але специфікація технічних вимог, особливо в частині визначення функціональних можливостей і сумісності інтерфейсів, - завдання системної інженерії. Це означає, що системний інженер повинен бути обізнаний про ключові характеристики компонентів, складових систему. Причому ці знання, що отримуються в значні степені в ході діалогу і взаємодії з фахівцями з проектування, повинні дозволити системному інженерові вибрати найбільш відповідні типи компонентів, а також визначити їх функції і способи сполучення з іншими компонентами.

Області компетенцій, відповідні діяльності системного інженера і фахівця по проектуванню, показані на рис.1 з урахуванням описаної вище ієрархії в системах. Видно, що знання, необхідні системному інженерові, охоплюють самий верхній рівень - систему і її оточення - і тягнуться до середнього рівня основних складових частин системи або компонентів. В той же час знання, необхідні проектувальникові, тягнуться від самого нижнього рівня (деталі) до рівня компонентів; саме тут ці дві області компетенцій «перекриваються». Саме на цьому рівні системний інженер повинен ефективно взаємодіяти з проектувальниками - виявляти і обговорювати технічні проблеми і виробляти працездатні рішення, які не піддають ризику ні процес проектування системи, ні її можливості в цілому.

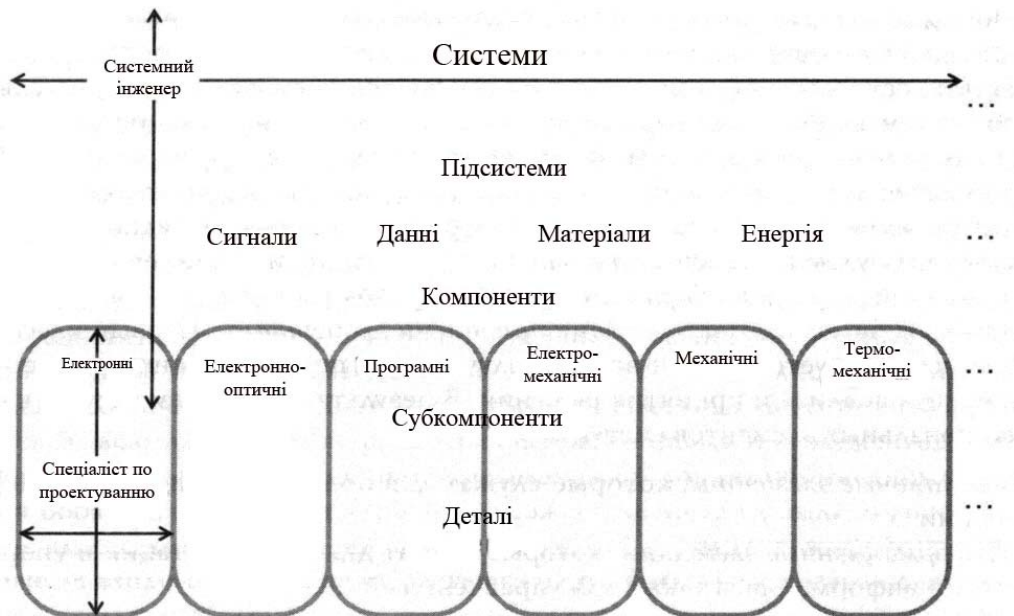


Рис. 3.1. Області компетенції системного інженера і спеціаліста по проектуванню

Горизонтальні межі областей спеціально показані стрілками, прямуючими до безкінечності; тим самим підкреслюється що ці області розширюються в залежності від складу конкретної системи. Якщо деякий субкомпонент або деталь виявляються критичними для роботи системи (наприклад, сумно відома

ізоляційна прокладка в ракеті-носії «Челленджера»), то системний інженер повинний бути готовим вивчити додатковий матеріал про проведення даного елемента в об'ємі достатньому для оцінки потенційного впливання на систему в цілому. Так часто буває у випадку розробки високовиробничих механічних і термомеханічних пристроїв, наприклад турбін і компресорів. Навпаки, якщо вказана в специфікаціях функція конкретного компоненту пред'являє незвичайні вимоги до його конструкції, фахівець з проектування повинен звернутися до системного інженера з пропозицією переглянути допущення системного рівня, що зумовили ці вимоги. Описана модель системи представляє системним інженерам просту методику розбиття системи на частини відповідно до функціональних і фізичних ознак, а саме: спочатку слід осмислити різні аспекти функціонування системи, а потім побудувати ієрархію її фізичних елементів.

#### 4. Елементи системи.

Кожен опис системи, як функціональний, так і фізичний, можна за допомогою декомпозиції розкласти на елементи. Нижче приведені опис обох категорій складових частин системи і рекомендований набір елементів, який слід використовувати при описі компонентів кожного типу. Середовище, в якому функціонують системи, включає три основну суть:

1. **Інформація** - зміст будь-якого знання і повідомлення.
2. **Речовина** - субстанція, з якої складаються всі фізичні об'єкти.
3. **Енергія**, яка приводить в працездатний стан і рух всі активні компоненти системи.

Оскільки будь-яка функція системи пов'язана з цілеспрямованою зміною характеристик однієї або декількох вказаної суті, то останні можна вважати природною основою для класифікації основних функціональних вузлів системи. Оскільки елементи, що мають справу з інформацією, використовуються для реалізації системних функцій як мінімум удвічі частіше, ніж елементи, що мають справу з матерією і енергією, їх зручно розділити на два класи: 1) елементи, що мають справу з передачею інформації (наприклад, радіосигналами), які ми називатимемо **сигнальними елементами**, і 2) елементи, що мають справу із стаціонарною інформацією (наприклад, комп'ютерними програмами), які ми називатимемо **інформаційними елементами**. Перший клас елементів асоціюється перш за все з отриманням і передачею даних, другою, - з процесами аналізу і ухвалення рішень. В результаті отримуємо чотири класи функціональних елементів системи:

1. **Сигнальні елементи**, які служать для отримання і передачі інформації.
2. **Інформаційні елементи**, які служать для інтерпретації і впорядкування інформації, а також для управління їй.
3. **Матеріальні елементи**, які служать для формування структури і перетворення матеріалів.
4. **Енергетичні елементи**, які служать для забезпечення енергією або рушійною силою.

Щоб читачі склали наочніше уявлення про особливості проектування кожного з цих чотирьох обширних класів функціональних елементів, ми

виділили набір характерних функціональних елементів, які можуть проілюструвати більшість найважливіших типів в кожному класі.

Бажаючи отримати узгоджений і достатньо представницький набір елементів, не дуже простих і не дуже складних, а також що мають широку область застосування, ми визначили три критерії відбору:

1. **Значущість.** Кожен функціональний елемент повинен виконувати особливу і важливу функцію, як правило, що включає декілька елементарних операцій.
2. **Унікальність.** Кожен функціональний елемент повинен належати по перевазі до технічної області окремої інженерної дисципліни.
3. **Уніфікована.** Функція, що виконується елементом, повинна використовуватися в безлічі систем.

Щоб зрозуміти, в яких умовах функціонує система, необхідно точно ідентифікувати її межі, тобто визначити, що знаходиться усередині системи, а що поза нею. Оскільки ми розглядаємо системну інженерію в контексті проекту розробки системи, як підмета розробці виробу береться система в цілому.

На перший погляд, визначення меж системи не складає труднощів, але на практиці дуже важко вирішити, що є частиною системи, а що частиною її оточення. Багато проектів виявилися провальними із-за неправильних допущень щодо того, що є зовнішнє, а що внутрішнє. Більш того, різні організації визначають межі системи по-різному - навіть для дуже схожих систем.

По щастю, є декілька критеріїв, що допомагають визначити, чи повинен деякий об'єкт визначатися як частина системи:

- ♦ **Контроль з боку розробника.** Чи контролює розробник системи розробку даного об'єкту? Чи може розробник вплинути на вимоги до об'єкту або ці вимоги визначаються незалежно від бажання розробника? Кошти виділяються з бюджету розробника або фінансування здійснює інша організація?
- ♦ **Контроль експлуатації.** Чи буде експлуатація даного об'єкту після впровадження системи знаходитися під контролем організації, що експлуатує її? Чи буде власник системи визначати цілі і завдання, що стоять перед цим об'єктом? Чи буде експлуатаційний контроль час від часу переходити до іншої організації?
- ♦ **Прив'язка функцій.** При функціональному описі системи чи може системний інженер прив'язувати функції до певних об'єктів?
- ♦ **Єдність мети.** Чи необхідний даний об'єкт для успішної роботи системи? Чи можна після впровадження системи видалити його без збитку для інших об'єктів?

Траплялося, що системні інженери допускали помилки, визначаючи об'єкт як частину системи, хоча ступінь контролю над ним (згідно приведеним вище критеріям) був дуже малий. І, як правило, на етапі розробки або експлуатації виявлялося, що подібний об'єкт нездібний виконати покладені на нього завдання.

На найраніших стадіях необхідно вирішити, чи є користувачі і оператори

частями системи або зовнішніми об'єктами. В більшості випадків їх слід розглядати як зовнішні об'єкти. Розробник системи і власник рідко володіють достатнім контролем над операторами, щоб включати їх в систему. Якщо оператор не є частиною системи, то системний інженер і розробник повинні приділити особливу увагу інтерфейсу оператора - критично важливому аспекту складної системи.

### 5. Межі системи: контекстна діаграма.

Один з важливих засобів обміну інформацією, доступних системному інженерові, - *контекстна діаграма*. На ній в наочному вигляді зображаються зовнішні об'єкти і їх взаємодії з системою. Типовий приклад контекстної діаграми показаний на мал. 3.2. Це так звана діаграма чорного ящика, на якій система показана у вигляді суцільної фігури в центрі, без яких би то не було деталей. Внутрішній устрій або принцип роботи приховані від читача. Діаграма складається з трьох компонентів:

1. **Зовнішні об'єкти** - це все об'єкти, з якими взаємодіє система.

Багато з них можна розглядати як джерела вхідних дій на систему або одержувачі вихідних дій з боку системи.

2. **Взаємодії**. Стрілками позначені взаємодії між системою і зовнішніми об'єктами. Напрямок стрілки указує, в яку сторону направлений конкретний зв'язок. Хоча допускається і використання двосторонніх стрілок, односторонні простіші для сприйняття. Тому системному інженерові не рекомендується застосовувати двосторонні стрілки, щоб не затемнити семантику взаємодії. У будь-якому випадку кожна взаємодія (стрілка) забезпечується влучною, яка означає, що саме передається через інтерфейс.

На цьому малюнку показані типові для контекстної діаграми взаємодії. На реальній контекстній діаграмі взаємодії були б помічені конкретними назвами, а не узагальненими поняттями. Мітки повинні чітко передавати сенс взаємодії, але при цьому бути досить лаконічними, щоб поміститися на діаграмі. Таким чином, слів «дані» або «зв'язок» краще уникати, тому що вони не несуть майже ніякого сенсу.

3. **Система**. Вона зображається суцільною фігурою - овалом, довкруги або прямокутником, в центрі якого знаходиться тільки назва системи без який-либo додаткової інформації.

Ми можемо класифікувати те, що передається через зовнішні інтерфейси, скориставшись приведеними вище визначеннями чотирьох основних елементів. Використовуючи ці елементи і додавши до них ще один, можна сформулювати п'ять категорій:

- ♦ дані;
- ♦ сигнали;
- ♦ матеріали;
- ♦ енергія;
- ♦ дії.

Таким чином, система взаємодіє зі своїм оточенням (точніше, із зовнішніми об'єктами), приймаючи або віддаючи один з перших чотирьох елементів або здійснюючи дію, яка тим або іншим чином впливає на систему або оточення.



Рис. 3.2. Контекстна діаграма

Побудова діаграми, подібній контекстній діаграмі для системи, може надати неоціниму допомогу при виділенні меж системи. На малюнку чітко і зрозуміло показані необхідні зовнішні інтерфейси з коротким поясненням, що передається всередину і назовні; тобто ми маємо наочне уявлення про входи і виходи системи.

## 6. Складність в сучасних системах.

Раніше в цьому розділі ми описали ієрархію систем - як система розбивається на підсистеми, потім на компоненти, субкомпоненти і, нарешті, деталі. У міру зростання складності сучасних систем ростуть також кількість, різноманітність і складність підсистем, компонентів і деталей, тобто об'єктів, що знаходяться на нижчих рівнях системної ієрархії.

Більш того, росте і складність взаємодій між ними. Принципи системної інженерії і практика їх застосування орієнтовані на боротьбу з цим ускладненням.

Все частіше зустрічаються окремі системи, які вже є або можуть стати частиною об'єкту великих масштабів. Хоча нині в ходу багато термінів, що описують поняття суперсистеми, термін «система систем» поширений особливо широко. Але в літературі можна зустріти і альтернативні терміни - одні означають в точності те ж саме, інші - щось інше.

Далі, ми коротко познайомимося з інженерією суті, яка вважається розташованою «вище» або складнішою, ніж окремі системи, - **системами систем**.

Нам будуть цікавими два визначення терміну «**система систем**» (system of systems - SOS); обидва пришли з Міністерства оборони США. Перше найпростіше:

Набір або впорядкована сукупність систем, що виникає в результаті комплексування незалежних і придатних до роботи систем в крупнішу систему, що володіє новими можливостями.

По суті справи, всякий раз, коли ряд незалежних і придатних до роботи систем об'єднується для придбання можливостей, що виходять за межі суми

можливостей окремих систем, ми отримуємо систему систем. Зрозуміло, рівень комплексирования може істотно розрізнитися. На одному кінці спектру знаходяться SOS, повністю інтегровані на найраніших стадіях розробки, коли окремі системи, хоча і здатні функціонувати незалежно, проектувалися мало не виключно для SOS. На іншому кінці ми зустрічаємо системи, слабо зв'язані для тимчасового вирішення локального завдання без яких-небудь формальних підстав, окрім згоди своїх власників. Тому для повноцінного опису різних нюансів SOS необхідна методика, що дозволяє охопити весь спектр комплексирования.

У 2008 році Міністерство оборони США випустило керівництво по системній інженерії спеціально для SOS, в якому було виділено чотири категорії подібних систем. Вони представлені в порядку зростання зв'язаності систем, що становлять, - від слабо до сильно зв'язаних:

- ♦ **Віртуальна.** У віртуальній SOS немає центрального пункту управління і єдиної узгодженої мети. Поведінка, характерна для крупномасштабних систем ймовірно і, можливо, навіть бажано, але передбачається, що в SOS такого типу для його підтримки повинні використовуватися сранительно слабо виражені механізми.
- ♦ **Коллаборативна.** Вхідні до складу коллаборативної SOS окремі системи взаємодіють на більш менш добровільній основі для досягнення узгоджених загальних цілей. Стандарти застосовуються, але немає ніякого центрального органу, який контролював би їх дотримання. Основні гравці спільно вирішують, чи потрібно надавати (і якщо потрібно, то як надавати) обслуговування, забезпечуючи тим самим деяку міру проходження стандартам регулювання і обслуговування.
- **Загальноновизнана.** У загальноновизнаній SOS є усвідомлені цілі, призначений керівник і виділені ресурси. Проте у складових її систем як і раніше є незалежні власники, цілі, фінансування, підходи до розробки і забезпечення функціонування. Для внесення змін до кожної окремої системи необхідна добровільна співпраця між нею і SOS.
- ♦ **Цільова.** Цільовими називаються інтегровані SOS, які створюються і управляються для досягнення конкретних цілей. Вони централізований управляються впродовж тривалого терміну служби для виконання як раніше поставлених, так і нових завдань, які можуть представляти інтерес для власників системи. Системи, що становлять, зберігають можливість працювати незалежно, але в нормальному режимі їх робота підпорядкована загальній меті.

Можна заперечити, що цільову SOS у вищеописаному сенсі швидше варто розглядати як одну складну систему, ніж систему систем; проте дані визначення охоплюють весь спектр ситуацій, що є на сьогоднішній день, коли системи комплексуються для виконання функції або володіння можливістю, якою не надасть жодна окремо узята система.

Майер (Maier) став таким, що започаткував формальне обговорення SOS, першим визначивши їх характеристики в 1998 році. З тих пір було опубліковано декілька робіт, в яких ці характеристики уточнювалися, проте в цілому вони виявилися на подив стабільними. Сейдж (Sage) і Цупан (Cuppen)



узагальнили їх таким чином:

1. **Експлуатаційна незалежність окремих систем.** SOS складається з систем, які незалежні і придатні до роботи окремо. Якщо розібрати SOS на системи, що становлять, то кожна зможе виконувати корисні функції незалежно від останніх.
2. **Адміністративна незалежність окремих систем.** Системи, складові SOS, не тільки здатні функціонувати незалежно, але, взагалі кажучи, так і працюють ради досягнення поставленої мети. Часто вони отримуються і комплексуються окремо і продовжують безперервно підтримувати своє існування і виконувати свої функції, які можуть відрізнятися від функцій, призначених SOS.
3. **Територіальна розподіленість.** Нерідко системи, що входять до складу SOS, знаходяться далеко один від одного і можуть обмінюватися між собою тільки інформацією і знаннями.
4. **Емерджентна поведінка.** SOS виконує функції і переслідує цілі, не обов'язково властиві якій-небудь з вхідних в її склад систем. Подібна емерджентна поведінка властива SOS в цілому і не характерний ні для однієї з вхідних в неї окремих систем.
5. **Еволюційний розвиток.** Розробка SOS зазвичай ведеться еволюційно. Компоненти структури, функції і цілі додаються, віддаляються і змінюються у міру накопичення досвіду роботи з системою. Таким чином, створення типової SOS ніколи не можна вважати повністю завершеним.

З часом ці характеристики піддавалися уточненням. І хоча уточнення не торкнулися приведені вище основні характеристики, все ж таки були додані дві важливі властивості.

1. **Самоорганізація.** SOS має динамічну організаційну структуру, здатну реагувати на зміни в оточенні і зміни поставлених цілей і завдань.
2. **Адаптація.** Як і організація, що розвивається, сама структура SOS може бути динамічною і реагувати на зовнішні зміни і сприйняття середовища.

Інженерія колаборативної або загальноновизнаною SOS повинна враховувати все сім основних характеристик SOS. Тому базових інструментів системної інженерії може опинитися недостатньо. У зв'язку з цим для інженерії таких складних структур були розроблені (і продовжують розроблятися) додаткові методи, інструменти і практичні прийоми.

Деякі з цих інструментів запозичені з інших розділів математики і інженерії, таких як теорія складності. Властивості, подібні емерджентності, самоорганізації і адаптації вивчалися в рамках даної теорії, і були розроблені засоби і методи уявлення внутрішньо властивою цим властивостям невизначеності. Проблема в тому, як зробити математичний апарат досить простим для застосування в системній інженерії.

Серед інших областей, в яких і до цього дня ведуться дослідження, пов'язані з інженерією системи систем, можна назвати соціальну інженерію, динаміку поведінки людини і хаотичні системи (теорію хаосу). Інженерія системи систем за своєю природою збільшує складність розробки окремих систем. Проте це ще не вищий рівень складності. Пригадаємо табл. 3.1, де була представлена ієрархія з *системою* на вершині. Її можна розширити, додавши

*систему систем і підприємство, що стоїть ще вище.* Ця ієрархія зображена на рис. 3.3.

Над SOS розташовано підприємство, яке, як правило, включає декілька SOS. Більш того, підприємство може складатися з різнотипних систем, необов'язково фізичних. Наприклад, підприємство включає системи, що складаються тільки з людей або соціальні системи, які необхідне комплексировать з фізичними системами. Формальне підприємство визначається як «освіта, що складається з людей, процесів, технологій, систем і інших ресурсів, розподілених організаційно і територіально і що взаємодіють між собою і з оточенням для досягнення загальної мети або рішення загальної задачі». Рівень взаємодії між цією суттю може мінятися, як і склад SOS. Приведеному визначенню відповідає багато суті, зокрема майже всі середні і крупні організації. Насправді структурні підрозділи деяких крупних корпорацій самі є підприємствами у вищеписаному сенсі.



Рис. 3.3. Піраміда ієрархії систем

Підпадають під це визначення також урядові агентства і міністерства, а рівно крупні соціальні і фізичні структури, наприклад міста і держави.

Складність інженерії систем масштабу підприємства обумовлена перш за все необхідністю комплексирования різнорідних системи і процесів. Типове підприємство включає наступні компоненти, які необхідно об'єднати в умовах властивої сучасному підприємству невизначеності:

- ◆ стратегія бізнесу і стратегічне планування;
- ◆ бизнес-процессы;
- ◆ служби підприємства;
- ◆ адміністративне управління;
- ◆ технічні процеси;
- ◆ управління людьми і їх взаємодіями;
- ◆ управління знаннями;
- ◆ інформаційно-технологічна інфраструктура і інвестиції в неї;
- ◆ управління основними засобами і устаткуванням;
- ◆ управління запасами;
- ◆ управління даними і інформацією.

**Під інженерією систем масштабу підприємства** (enterprise systems engineering) розуміється застосування принципів і практичних прийомів

системної інженерії до інженерії систем, що входять до складу підприємства. Даний термін позначає розробку окремих систем підприємства, що становлять. З'явився також ще один, ширший термін, в якому відсутнє слово «система» - *інженерія масштабу підприємства* (enterprise engineering). Зазвичай під цим розуміється розробка архітектури підприємства, а також розробка, реалізація і експлуатація підприємства як цілого. Деякі автори користуються вищенаведеними термінами як синонімами, проте насправді вони відносяться до різних рівнів абстракції.

Причина, по якій інженерія систем масштабу підприємства виявляється складнішою за інженерію SOS, полягає в тому, що багато компонентів підприємства налічують одну або декілька SOS. Тому підприємство можна було б розглядати як результат комплексирования багато SOS.

Так само як і для інженерії системи систем, ведеться активна розробка інструментів, методів і прийомів для цієї порівняно молодій області.

## **КОРОТКІ ВИСНОВКИ**

### **Складові частини і інтерфейси системи**

Оскільки системному інженерові необхідні обширні пізнання в декількох взаємозв'язаних областях, що стосуються створення складної системи, виникає питання, наскільки глибокими повинні бути ці знання.

### **Ієрархія складних систем**

Складну систему можна представити у вигляді ієрархічної структури, що складається з підсистем, компонентів, субкомпонентів і деталей.

Область компетенцій системного інженера охоплює самий верхній рівень і тягнеться до рівня компонентів, розповсюджуючись на декілька областей знань. Знання ж фахівця з проектування тягнуться вгору від рівня деталей до рівня компонентів і, як правило, обмежені окремою технологічною областю.

### **Складові частини системи**

Складові частини, з яких складаються всі комплексні системи, розташовуються на рівні компонентів і характеризуються функціональними і фізичними ознаками. Вони виконують в системі чітко певну значущу функцію і є специфічними, тобто відносяться до області окремої інженерної дисципліни.

Функціональні елементи - це функціональні еквіваленти компонентів; серед них за ознакою фізичного носія виділяються чотири категорії компонентів:

- ◆ сигналні елементи, які служать для отримання і передачі інформації;
- ◆ інформаційні елементи, які служать для інтерпретації, впорядкування інформації і управління їй;
- ◆ матеріальні елементи, які служать для формування структури і перетворення матеріалів;
- ◆ енергетичні елементи, які служать джерелами енергії або рушійної сили.

Компоненти - це фізичне втілення функціональних елементів; серед них по фізичних і конструктивних ознаках ми виділили шість категорій:

- ◆ електронні;
- ◆ електронно-оптичні;
- ◆ електромеханічні;
- ◆ механічні;
- ◆ термомеханічні;
- ◆ програмні.

Модель складових частин системи може бути корисною для визначення дій, здатних принести необхідні результати функціонування системи, полегшити функціональну деталізацію і опис окремих функціональних можливостей, а також ідентифікацію інтерфейсів між підсистемами і компонентами і візуалізацію фізичної архітектури системи.

### **Оточення системи**

Під оточенням системи розуміється все, що знаходиться поза системою, але взаємодіє з нею, а саме: 1) оператори системи (частина функціональних можливостей системи, що не входить в комплект постачання); 2) технічне обслуговування і ремонт, укриття і системи забезпечення; 3) упаковка, доставка і зберігання; 4) погодні і інші фізичні умови навколишнього середовища; 5) погрози.

### **Інтерфейси і взаємодії**

Інтерфейси - найважливіша турбота системного інженера; від них залежать взаємодії між компонентами. Можна виділити три категорії інтерфейсів: з'єднувачі, ізолятори і перетворювачі. Інтерфейси необхідно виявити і специфікувати, а також координувати і контролювати. Крім того, для комплексирования, а також технічного обслуговування і ремонту зазвичай надаються контрольні інтерфейси.

### **Складність в сучасних системах**

Будь-яка система - частина крупнішої освіти. Іноді це освіту само можна назвати системою (а не просто навколишнім середовищем, або «природою»). У таких випадках говорять про системи систем (БОБ), у яких є сім відмітних характеристик: експлуатаційна незалежність окремих систем, адміністративна незалежність окремих систем, територіальна розподеленность, емерджентное поведінка, еволюційний розвиток, самоорганізація і адаптація.

Інженерія систем масштабу підприємства аналогічна по складності, але увага тут сосредотачивается на організації як окремій суті. Оскільки до складу підприємства входять не тільки технічні, але і соціальні системи, його складність може виявитися непередбачуваною.

## Лекція 4

# ПРОЦЕС РОЗРОБКИ СИСТЕМ

Еволюція нової системи з моменту усвідомлення потреби в ній та ідентифікації технічного підходу, придатного для реалізації задуму на всьому протязі розробки і подальшого введення в експлуатацію, являє собою складну послідовність перетворень, яку ми далі будемо називати *процесом розробки системи*.

Розробка типової великої системи характеризується наступними ознаками:

- ◆ це комплексна і складна діяльність;
- ◆ має на меті задоволення важливої потреби користувача;
- ◆ займає багато часу (від початку до завершення розробки зазвичай проходить кілька років);
- ◆ має на увазі рішення великого числа взаємопов'язаних завдань;
- ◆ використовує досягнення декількох різних дисциплін;
- ◆ зазвичай здійснюється кількома організаціями;
- ◆ має конкретні терміни і бюджет.

Розробка і подальший введення в експлуатацію складної системи, по суті, вимагають все більших і більших ресурсів у міру еволюції цієї системи від появи задуму до подальшої розробки інженерно-технічних рішень і далі до виробництва та використання за призначенням. До того ж, впровадження нових технологій неминуче пов'язане з ризиками, які необхідно якомога раніше виявити і спробувати усунути. Ці фактори змушують вести розробку крок за кроком, коли перед прийняттям рішення про перехід до наступного кроку демонструється успішне завершення попереднього і перевіряється правильність обґрунтування продовження робіт.

**Модель життєвого циклу для системного інженера.** При побудові моделі життєвого циклу, яка відображала б істотні переходи від одного виду діяльності системного інженера до іншого протягом існування системи, виявилось, що найзручніше розбити життєвий цикл на три великих стадії, а їх, у свою чергу, - на вісім етапів. Ця структура показана на рис. 4.1 і розглядається нижче. Назви стадій обрані з тим розрахунком, щоб у них відбивалася принципово важлива діяльність, яка має місце на кожному етапі процесу. Деякі назви неминуче збігаються або, у всякому разі, корелюють з назвами аналогічних частин в інших існуючих моделях життєвого циклу.

**Моделі життєвого циклу програмного забезпечення.** Стадії життєвого циклу системи і складові їх етапи, представлені в розглянутих вище моделях, застосовні до більшості складних систем, у тому числі зі значною часткою програмного забезпечення на рівні компонентів. Однак життєвий цикл систем, в яких практично всі функціональні можливості втілені в програмному забезпеченні (ПО) (наприклад, в сучасних фінансових системах, системах резервування авіаквитків, веб- та інших інформаційних системах), хоча і схожий за формою, але в загальному випадку включає декілька ітерацій і розробку прототипу.



Рис.4.1. Модель життєвого циклу для системного інженера.

### Стадії в моделі життєвого циклу для системного інженера

Як було сказано вище і показано на рис. 4.2, модель життєвого циклу системи включає три стадії, з яких перші дві охоплюють частину життєвого циклу, що відноситься до розробки, а остання - період по завершенні розробки. Ці стадії відзначають найбільш важливі переходи в життєвому циклі системи, а також зміни типу та предмета діяльності системного інженера. У цій книзі ми будемо називати стадії наступним чином: 1) стадія розробки концепції - вона охоплює початковий період, коли формується і вибирається концепція системи, реалізація якої дозволяє найкращим чином задовільнити встановлені потреби; 2) стадія розробки інженерно-технічних рішень - вона охоплює перехід від концепції системи до проекту фізичної реалізації системи який успішно пройшов валідацію, що відповідає експлуатаційним, вартісним і тимчасовим вимогам; 3) построзробна стадія - вона включає виробництво, розгортання, експлуатацію та супроводження системи протягом усього терміну її служби. Назви стадій обрані у відповідності з основним типом діяльності.

#### Основні цілі стадії розробки концепції такі:

1. Переконалися в тому, що потреба в новій системі, так само як і ринок для неї, дійсно існують, а також у тому, що є технічні та економічні можливості її створення.
2. Дослідити можливі концепції системи, сформулювати і схвалити сукупність вимог до її поведінки.
3. Вибрати найкращу концепцію системи, визначити її функціональні характеристики і скласти детальний план подальших етапів розробки, виробництва і введення системи в експлуатацію.
4. Розробити нові технології, необхідні для реалізації прийнятої концепції, і переконалися в їх здатності задовольнити вимогам.

#### Основні цілі стадії розробки інженерно-технічних рішень такі:

1. Здійснити інженерно-технічну розробку системи-прототипу, яка задовольняє функціональним вимогам, а також вимогам по надійності, ремонтпридатності і безпеки.
2. Створити систему, економічну у виробництві та використанні і продемонструвати її придатність до експлуатації.

Построзна стадія включає види діяльності, що не відносяться власне до розробки, але тим не менш потребують суттєвої підтримки з боку системного інженера, особливо в ситуації, коли виникають непередбачені проблеми, що вимагають термінового вирішення. Крім того, безперервний технічний прогрес нерідко викликає необхідність в модернізації системи без виведення з експлуатації, а для вирішення цього завдання системна інженерія може знадобитися не в меншій мірі, ніж на стадіях розробки концепції та інженерно-технічних рішень. Дана стадія частково співвідноситься зі стадією виробництва та розгортання і повністю - зі стадією експлуатації та супроводу у моделі МО США.

### Етапи розробки концепції

В життєвому циклі системи виділяються три основні стадії, які були описані вище. Однак всередині кожної стадії можна виділити окремі етапи, які характеризуються чітко визначеними цілями і діяльністю.

**Етап аналізу потреб.** На цьому етапі виявляється потреба в новій системі. Тут ставлять запитання типу «Чи дійсно потрібна нова система?» І «Чи існує практично здійснений спосіб задовольнити цю потребу?». Для відповіді на такі питання необхідно критично проаналізувати, якою мірою поточні та передбачувані в подальшому потреби можуть бути задоволені шляхом фізичної та функціональної модернізації наявних коштів, а також зрозуміти, чи здатні існуючі технології забезпечити бажане нарощування можливостей. У багатьох випадках стимулом до створення нової системи може послужити результат аналізу практичних потреб або поява інноваційного продукту, але чітко зафіксувати момент народження нової системи зазвичай не вдається.

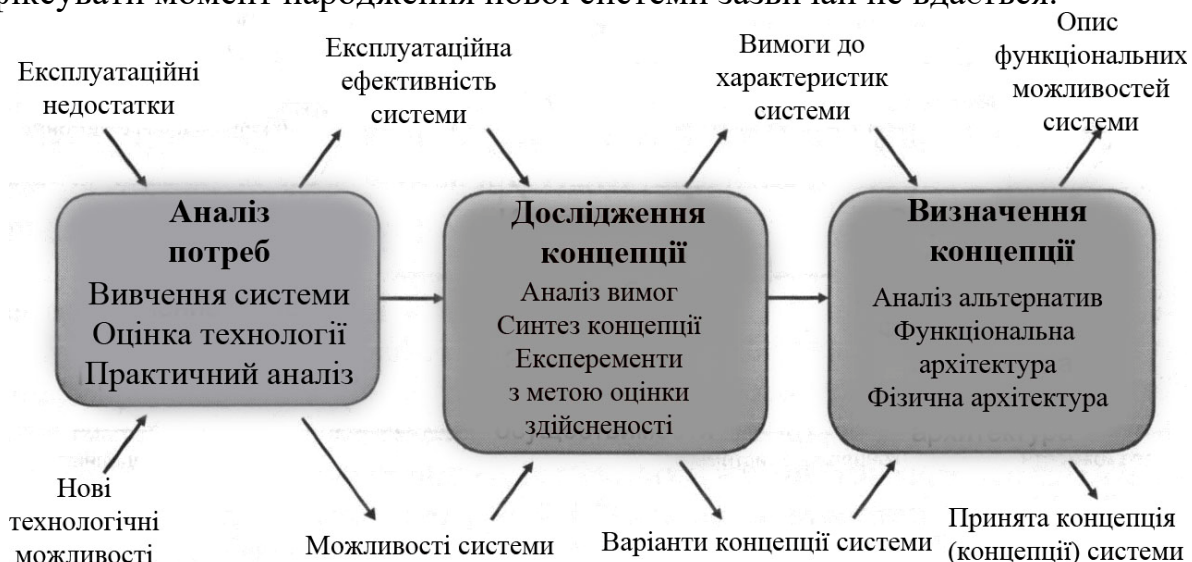


Рис. 4.2. Етап розробки концепції в життєвому циклі системи.

**Етап дослідження концепції.** На цьому етапі досліджуються можливі варіанти концепцій системи і ставлять запитання типу «Якими мають бути характеристики нової системи, щоб задовольнити виявлення потреби?» І «Чи існує хоча б один спосіб досягнення таких характеристик з прийнятними витратами?». Якщо відповідь на ці питання позитивний, то ще до витрати значних сил і засобів на розробку проекту нової системи відомо, що у нього є чітко визначена і досяжна мета.

*Етап визначення концепції.* На цьому етапі визначається краща концепція. Слід відповісти на запитання: «Які ключові характеристики концепції системи, при яких досягається сприятливий баланс між функціональними можливостями, терміном служби і вартістю?» Для цього потрібно розглянути кілька альтернативних концепцій і порівняти їх характеристики, практичну корисність, ризики розробки і вартість. Отримавши задовільну відповідь, можна приймати рішення про виділення серйозних ресурсів на розробку нової системи.

*Етап ескізного проектування.* Успішне завершення стадії розробки інженерно-технічних рішень істотно залежить від міцності фундаменту, закладеного на стадії розробки концепції. Але оскільки концептуальне проектування - головним чином аналітична робота, на яку виділяються обмежені ресурси, то неминуче залишаються невідомі, які ще належить визначити і прояснити. Важливо виявити ці «невідомі невідомі» і прийняти по них рішення на ранніх етапах стадії розробки. Зокрема, слід вжити всіх можливих заходів, щоб мінімізувати число невиявлених проблем ще перед тим, як від функціонального проекту і зв'язаних з ним вимог до системи буде здійснено перехід до технічних специфікацій (проектним, конструкторським, виробничо-технологічними) на окремі апаратні і програмні елементи системи.

На етапі ескізного проектування стоять два основні завдання: 1) ідентифікація і зниження ризиків розробки та 2) розробка проектною документації на систему. Цей етап особливо важливий, коли концепція системи передбачає використання передової технології, яка раніше не застосовувалася у подібних розробках, або коли для досягнення необхідних характеристик доводиться піддавати компоненти більшому навантаженню, ніж було прийнято раніше. Мета ескізного проектування полягає в тому, щоб отримати уявлення про конструкцію складових частин системи, рішення по яких не опрацьовані, і продемонструвати практичну можливість виконання вимог до цих частин, а також закласти основу для перетворення функціональних вимог до системи в документацію на систему і в технічні вимоги до компонентів. Системна інженерія займає центральне місце в рішеннях про те, що потребує валідації і як її провести, а також в інтерпретації результатів.

*Етап технічного проектування.* На даному етапі виконується детальне технічне проектування системи. Через масштабність завдання по ходу процесу зазвичай кілька разів проводиться формальний аналіз стану проекту. У цього заходу є важлива функція - дати замовнику або користувачеві можливість ознайомитися з проектом на ранніх етапах, проконтролювати виконання бюджету і графіку і висловити розробнику корисні критичні зауваження.

*Етап комплексування та атестації.* Процес комплексування компонентів складної системи в працююче ціле і оцінки функціонування системи в реальних умовах номінально є частиною процесу розробки інженерно-технічних рішень, оскільки в цій точці програми розробки немає ніякого формального розриву. Однак ролі та обов'язки системного інженера під час технічного проектування елементів системи і в процесі комплексування та атестації системи істотно розрізняються. Оскільки ця книга присвячена



системній інженерії, ми розглядаємо процес комплексування та атестації системи як окремий етап життєвого циклу.

*Етап виробництва.* Це перший з двох етапів, які складають поствиробничу стадію, які є точними аналогами стадій «виробництво і розгортання» та «експлуатація та супровід» в моделі МО США. Як би ретельно не був пристосований до потреб виробництва проект системими, в процесі виробництва неминуче виникають проблеми. Завжди існують несподівані перешкоди, які керівник проекту не в змозі контролювати, наприклад страйк на заводі постачальника, непередбачувані труднощі із забезпеченням інструментами, помилки в критично важливих програмах або несподіваний збій у ході заводських комплексних випробувань. Такі ситуації загрожують дорогим зривом виробничого графіка і, відповідно, вимагають швидкого і рішучого виправлення.

*Етап експлуатації та супроводу.* На даному етапі потреба в підтримці з боку системного інженера відчувається ще гостріше. Оператори системи і фахівці з технічного обслуговування і ремонту, швидше за все, недостатньо досвідчені в тонких деталях функціонування та обслуговування системи. Хоча спеціально навчені інженери з експлуатації в загальній ситуації знають, що робити, на випадок, якщо проблема вийде за рамки їх кваліфікації, повинна бути передбачена можливість запросити досвідченого системного інженера. Процес створення нової системи можна описати, не посилаючись на її схожість з існуючими системами, призначеними для задоволення таких же або схожих потреб. Концепцію системи в цілому і всіх її елементів часто представляють так, ніби все починається з чистого аркуша, чого насправді практично ніколи не буває.

### **Попередня система.**

У більшості випадків, коли нова технологія використовується для досягнення радикальних змін у таких галузях, як транспорт, банківська справа або системи бойового застосування, якась система вже існує. У знову створюваній системі зміни зазвичай зосереджені в декількох підсистемах, тоді як архітектура системи в цілому і інші підсистеми модифікуються не сильно. Навіть впровадження автоматизації звичайно змінює лише технічну сторону, але не істота процесу. Отже, якщо не враховувати таких проривів, як поява ядерних реакторів або космічних кораблів, при розробці нової системи майже напевно існує якась попередня система, яка може послужити відправною точкою.

Попередня система впливає на розробку нової системи трьома способами:

1. Виявлені недоліки старої системи часто стають рушійною силою нової розробки. При цьому в центрі уваги знаходяться найбільш важливі характеристики та функції, які має забезпечити нова система.
2. Якщо недоліки не настільки серйозні, щоб повністю відмовитися від існуючої системи, то її концепція і функціональна архітектура можуть розглядатися як найкраща вихідна точка при дослідженні альтернатив.
3. Якщо значні частини існуючої системи справляються зі своїми функціями задовільно і не застаріли у зв'язку з появою нових технологій, то їх

використання з мінімальними змінами може замінно скоротити витрати (і знизити ризики).

**Метод системної інженерії** можна уявити собі як систематичне застосування наукового методу до інженерної діяльності по створенню складної системи. Можна вважати, що він включає чотири основних види діяльності, які застосовуються послідовно, як показано на рис. 4.11.

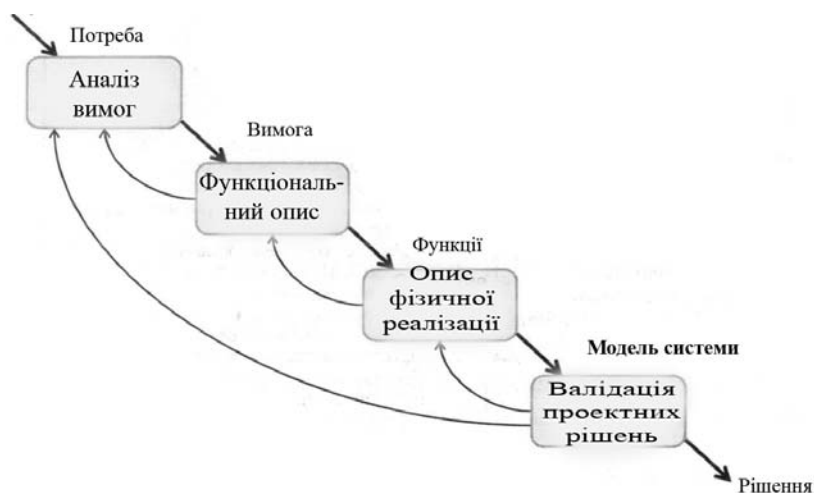


Рис. 4.3. Метод системної інженерії.

1. Аналіз вимог.
2. Функціональний опис.
3. Опис фізичної реалізації.
4. Валідація проектних рішень.

Характерні особливості кроків можуть відрізнятися в залежності від типу системи і стадії її розробки. Але основні принципи застосування досить

**1. Аналіз вимог (постановка завдання).** Типові дії включають:

збір та систематизацію всіх вхідних умов, у тому числі вимог, планів, точок прийняття рішень і моделей, отриманих на попередньому етапі; відповідь на питання «навіщо» стосовно всіх вимог - в термінах практичних потреб, обмежень, оточення та інших високорівневих цілей; прояснення вимог до того, що, наскільки добре і в рамках яких обмежень повинна робити система; виправлення невідповідностей і вираз вимог у вимірних показниках там, де це можливо.

**2. Опис функцій (аналіз функціонування і прив'язка функцій).** Типові дії: переклад вимог (навіщо, чому) на мову функцій (дій і робіт), які повинна виконувати система (що); декомпозиція вимог з прив'язкою до функціональних складових частин; опис взаємодій між функціональними елементами, що дозволяють закласти основу для побудови модульної конфігурації.

**3. Опис фізичної реалізації (синтез, аналіз фізичної реалізації та розміщення елементів).** Типові дії: синтез кількох альтернативних компонентів системи, що представляють різноманіття проектних підходів до реалізації необхідних функцій і

дозволяють найбільш простим чином здійснити взаємодію і реалізувати інтерфейси між елементами структури;  
 вибір найбільш пріоритетного підходу, де в основі прийняття рішення лежить досягнення компромісу на основі аналізу сукупності заздалегідь визначених критеріїв із заданими пріоритетами (показниками ефективності) в інтересах отримання найкращого «балансу» між показниками функціонування, ризиками, витратами і термінами;  
 опрацювання проектних рішень з необхідним ступенем деталізації.

**4. Валідація проектних рішень (верифікація та оцінка).** Типові дії:

проективання моделей оточення системи (логічної, математичної, імітаційної і фізичної), що відображають всі істотні аспекти вимог та обмежень;

імітація чи випробування і аналіз системного рішення (рішень) на моделях оточення;

при необхідності - виконання ітерацій для коригування моделі системи або моделей оточення або для послаблення надто жорстких вимог, до тих пір, поки проектні рішення і вимоги не будуть повністю узгоджені.

Випробування та атестація невіддільні від проектування і є невід'ємною частиною проекту. У простих випадках, наприклад при написанні картини, функція випробування та атестації реалізується художником як частина процесу перенесення художнього задуму на полотно. Якщо отримана картина не відповідає задуму, автор змінює її, накладаючи мазки пензлем, щоб глядацький ефект (показники функціонування) відповідав поставленій меті. Таким чином, проектування - це процес зі зворотнім зв'язком, в якому випробування та атестація якраз і грають роль зворотнього зв'язку, що дозволяє скоректувати результат, так щоб він задовольняв заданим вимогам.

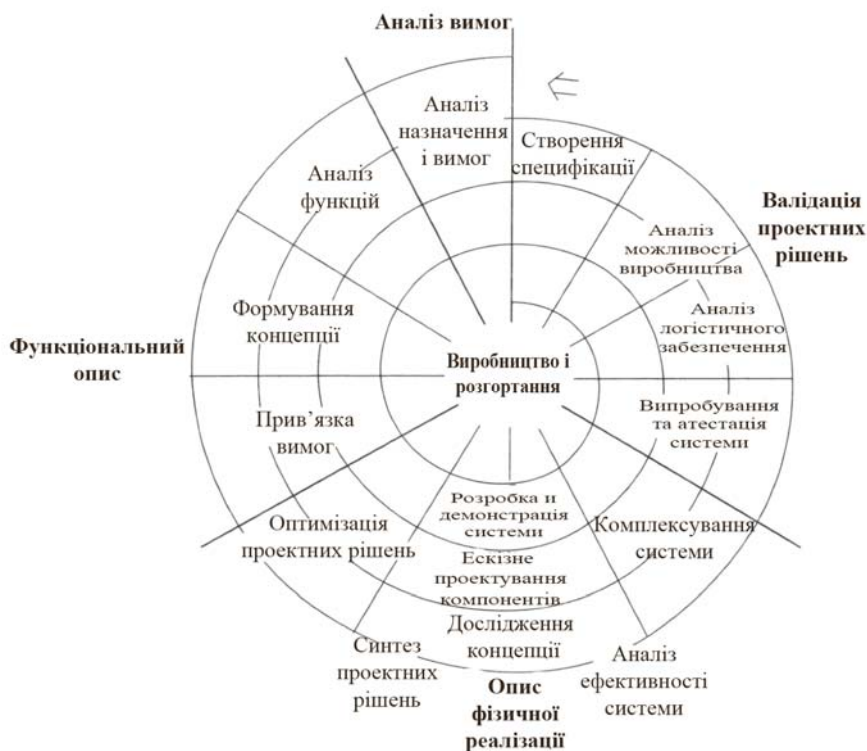


Рис. 4.4. Спіральна модель життєвого циклу системи.

## **Невідомі**

У будь-якому проекті розробки нової системи існує безліч невідомих, проблеми з якими повинні бути дозволені у ході виробництва виробу. Всякий раз, коли в процесі створення системи відбувається помітний відхід від сформованої практики, результат стає непередбачуваним. Вартість проекту залежить від різних чинників, жоден з яких в точності не відомий. Для усунення несумісності інтерфейсів часто доводиться вносити в проект коригування по обидві сторони інтерфейсу, а це нерідко тягне за собою несподівані та іноді важкозрозумілі технічні проблеми.

Важливе завдання системної інженерії - керувати розробкою системи, так щоб невідомі ставали відомими якомога раніше. Будь-які несподіванки на пізніх етапах програми чи проекту можуть обійтися в багато разів дорожче, ніж при їх виявленні на початкових етапах.

Багато невідомих очевидні із самого початку, так що їх можна назвати «відомими невідомими». Вони відразу ж ідентифікуються як потенціальні проблеми, які уважно розглядаються і вирішуються. Зазвичай цьому сприяє серія вирішальних експериментів з використанням можелювання та / або спеціального експериментального обладнання, а також програмного забезпечення. Однак є ряд проблем, які виявляються пізніше, в ході розробки системи. Ці непередбачені проблеми часто називають «невідомими невідомими», щоб відрізнити їх від невідомих, про існування яких всі знали спочатку, так що проблеми були дозволені ще до того, як вони встигли зробити негативний вплив на процес розробки.

### **Перетворення невідомого у відоме**

Існування невідомих невідомих значно ускладнює усунення всіх невідомих чинників. Доводиться активно шукати підводні камені в місцях, де вони часто зустрічаються. Системний інженер повинен очолити ці пошуки, ґрунтуючись на досвіді, накопиченому при розробці інших систем, полягаючись на свою технічну проникливість і ставлячи собі питання «Що якщо ..?».

Оскільки кожне невідоме підриває впевненість у кінцевому успіху, воно являє потенційний ризик. Скажемо більше: невідомі складають основні ризики в будь-якій програмі розробки. Тому завдання оцінки та пом'якшення ризиків - це фактично і є виявлення та усунення невідомих.

Вирішити подібні проблеми дозволяють аналіз, імітаційне моделювання та випробування, оскільки з їх допомогою можна визначити і кількісно оцінити критично важливі характеристики системи. Ця робота починається на самих ранніх етапах формування концепції і триває протягом всієї розробки; при цьому змінюються лише предмет і характер, але не мета і підхід.

При проектуванні нової системи або нового елемента системи з при-трансформаційних змін підходу, який раніше ніколи не випробували в подібних умовах (наприклад, для виготовлення елементів конструкції, що зазнають підвищене навантаження, використовуються нові матеріали), фахівець з проектування стикається з цілою низкою невідомих, в тому що стосується поведінки готової конструкції (наприклад, елемент, виготовлений з нового матеріалу, не вдається відлити в потрібну форму традиційними способами). У таких випадках тільки іс-питаніє дозволяє дізнатися, ускладнюють чи невідомі

фактори роботу настільки, що потрібно суттєво змінити проект або навіть повністю відмовитися від обраного підходу.

Якщо запропонований новий підхід до проектування, нерозумно чекати завершення проекту, щоб зрозуміти, чи було корисним таке нововведення. Для початку слід провести випробування на теоретичній або експериментальній моделі елемента конструкції, яку можна створити швидко і з мінімальними затратами. При цьому знадобиться з усією ретельністю оцінити оптимальне співвідношення вигоди, обумовленої більшою реалістичністю моделі, і тимчасових та фінансових витрат на досягнення такої реалістичності. Часто рішення, прийняте за підсумками цієї оцінки, відноситься до рівня системи, а не компонента, особливо якщо робота елемента може вплинути на систему в цілому. Якщо невідомі в основному пов'язані з функціональним поведінкою елемента, то рекомендується обчислювальна або імітаційна модель. Якщо ж невідомі відносяться до властивостей матеріалів, потрібно експериментальна модель.

### **Підхід системної інженерії до випробувань**

Підхід системної інженерії до випробувань можна проілюструвати, порівнявши точки зору інженера-конструктора, інженера-випробувача і системного інженера. Конструктор хоче бути впевнений, що компонент успішно пройшов випробування, тобто отримати позитивну відповідь на питання «Чи все в порядку?». Інженер-випробувач хоче знати, наскільки ретельно було організовано випробування, щоб бути впевненим у тому, що компонент отримав достатнє навантаження. Системному інженеру цікаво знайти та ідентифікувати всі дефекти компонента. Якщо компонент не зумів успішно пройти випробування, то системний інженер хоче з'ясувати причини, щоб зрозуміти, як усунути дефект.

Зі сказаного ясно, що системний інженер приділяє увагу не тільки умовам проведення випробувань, але і збору даних, що точно показують, як працювали або не працювали різні частини системи. Але одного лише збору даних недостатньо - необхідно ще мати процедури для їх аналізу. Найчастіше такі процедури складні і вимагають розвиненої аналітичної техніки, і це також необхідно передбачити.

Крім того, можна зробити висновок, що системний інженер повинен брати активну участь у розробці програми і методики випробувань і у виборі контрольовано-вимірювальної апаратури. Насправді ініціатива підготовки плану випробувань повинна належати системному інженерові, який тісно співпрацює з інженерами-випробувачами. Для системного інженера випробування - все одно що експеримент для вченого, тобто засіб збору важливих даних про поведінку системи в контрольованих умовах.

## **КОРОТКІ ВИСНОВКИ**

### **Застосування системної інженерії протягом життєвого циклу системи**

Програма розробки більшої системи - складний комплекс робіт, направлених на задоволення важливої потреби користувача. Цей комплекс включає безліч дисциплін, передбачає застосування нових технологій, вимагає

поступово наростаючого виділення ресурсів та виконується покроково відповідно до виділеного бюджету і затвердженого графіка.

### **Життєвий цикл системи**

Життєвий цикл системи можна розділити на три основні стадії.

#### **Розробка концепції.**

У завдання системної інженерії входять виявлення потреби в системі, дослідження практично здійснених концепцій і визначення кращою концепції системи. Стадію розробки концепції можна далі розбити на три етапи:

1. *Аналіз потреб:* визначення та валідація потреби в новій системі, демонстрація її технічної здійсненності та визначення вимог до функціональних можливостей системи.
2. *Дослідження концепції:* дослідження реалізованих концепцій і визначення вимог до показників функціонування.
3. *Визначення концепції:* вивчення альтернативних варіантів, вибір переважаючої концепції виходячи з показників функціонування, стоїмості, термінів і ризиків, а також визначення функціональних специфікацій системи (А-специфікацій).

#### **Розробка інженерно-технічних рішень.**

У завдання системної інженерії входять валідація нової технології, перетворення обраної концепції в проектні рішення щодо апаратного та програмного забезпечення, створення і випробування готових моделей. Стадію розробки інженерно-технічних рішень можна розбити на три етапи.

1. *Ескізне проектування:* ідентифікація зон ризику, зниження цих ризиків за допомогою аналізу, розробки і випробувань, а також визначення проектних специфікацій системи (В-специфікацій).
2. *Технічне проектування:* попереднє і остаточне проектування, створення та випробування апаратних і програмних компонентів, наприклад елементів конфігурації.
3. *Комплексування та атестація:* збірка компонентів в готовий дослідний зразок, атестація дослідного зразка та виправлення недоліків.

#### **Поствиробнича стадія.**

У завдання системної інженерії входять участь у виробництві та розгортанні системи, а також надання технічної підтримки в процесі її експлуатації, обслуговування та ремонту. Стадія ділиться на два етапи:

1. *Виробництво:* розробка інструментів і оснащення; виробництво системної продукції, поставка системи користувачам і введення її в експлуатацію.
2. *Експлуатація та супровід:* забезпечення експлуатації, технічного обслуговування і ремонту системи, а також підготовка і здійснення заходів з модернізації системи без її виведення з експлуатації.

## **Еволюційні характеристики процесу розробки**

Як правило, нові системи виростають з попередніх їм, оскільки функціональна архітектура і навіть деякі компоненти останніх можуть бути використані повторно.

Нова система поступово «матеріалізується» в ході розробки. Опису системи і проектні рішення еволюціонують від концепції до реальності. Документи, графіки, діаграми, моделі і продукція також зазнають відповідних змін. Більш того, ключові учасники розробки змінюють один одного; проте системна інженерія відіграє ключову роль протягом усіх етапів.

## **Метод системної інженерії**

Метод системної інженерії включає чотири основні кроки:

1. *Аналіз вимог* - визначається, чому необхідні ті чи інші вимоги.
2. *Функціональний опис* - вимоги переводяться на мову функцій.
3. *Опис фізичної реалізації* - синтезуються альтернативні варіанти фізичної реалізації.
4. *Валідація проектних рішень* - моделюється оточення системи.

Ці чотири кроки застосовуються на кожному етапі розробки. Конкретний спосіб застосування методу системної інженерії залежить від етапу життєвого циклу; у міру того як система матеріалізується, фокус зміщується в напрямку зверху вниз - з рівня системи (етап аналізу потреб) на рівень компонентів і деталей (етап технічного проектування).

## **Випробування на протязі розробки системи**

Випробування - це процес виявлення невідомих дефектів проекту. У ході випробування перевіряється, що проблеми з усіма відомими невідомими вирішені, і виявляються невідомі невідомі і їх причини. Якщо невідомі будуть виявлені із запізненням, це може обійтися надзвичайно дорого, тому планування випробувань і аналіз їх результатів - одна з найважливіших обов'язків системного інженера.

## Лекція 5. УПРАВЛІННЯ СИСТЕМНОЮ ІНЖЕНЕРІЄЮ

1. Управління розробкою системи.
2. Ієрархічна структура робіт.
3. План управління системною інженерією.
4. Управління ризиками.

### 1. Управління розробкою системи і ризиками.

Створення складної системи включає безліч взаємопов'язаних робіт, виконуваних десятками або сотнями людей, а також низкою субпідрядників та інших юридичних осіб. І це не тільки власне процес розробки, а й все, що зазвичай необхідно для забезпечення функціонування системи, в тому числі технічне обслуговування, документація, навчання та багато іншого. Потрібно розробити, придбати і забезпечити доступність випробувального та іншого необхідного обладнання, приміщень та транспортних засобів. Не можна випустити з уваги жодну сторону управління проектом і системної інженерії, в тому числі планування, складання графіків і кошторисів робіт, управління конфігурацією (рис.5.1).



Рис.5.1. Роль системної інженерії в управлінні проектом.

Розділи цієї глави відносяться до управління будь-якими видами діяльності у сфері системної інженерії для всіх типів складних систем. Однак у програмних системах, де практично всі функціональні можливості визначаються програмним забезпеченням, існує ряд особливостей, які



потребуються окремих розгляд. Про них ми поговоримо в розділі 11, зокрема в розділі «Управління програмної інженерією».

### **Підготовка пропозиції та технічне завдання**

Розробку системи часто ініціює сторона, що має деяку потребу, наприклад замовник, у якого в конкурентному середовищі звернення за допомогою нерідко відбувається у формі запиту пропозиції (request for proposal - RFP). Після того як організація прийме рішення відгукнутися на RFP, призначається керівник програми або група професіоналів, які повинні підготувати пропозицію. Хоча системний інженер може офіційно й не входити до складу групи, важливо мати впевненість, що технічні ідеї, а також проектні рішення і інтерфейси практично здійсненні. Тому навіть на ранніх стадіях проекту необхідна тісна співпраця між системним інженером і керівництвом проекту.

Найважливіший елемент пропозиції - технічне завдання (statement of work - SOW). Це словесний опис робіт, які належить виконати для того, щоб створити систему, що відповідає потребам замовника.

Системний інженер приділяє особливу увагу розробці виробу і стежить за тим, щоб в описуваній обсяг робіт були включені всі необхідні продукти і послуги. Точніше, системний інженер зосереджений на потребах замовника і відповідає за те, щоб в основу SOW була покладена концепція функціонування, яка заслуговує довіри. Він аналізує, чи використовуються в неявній проектного рішення успадковані компоненти, і якщо так, то чи доступні вони; перевіряє, чи включені в систему комерційні COTS-продукти, і визначає рівні технологічної готовності важливих підсистем, що згадуються в попередньому проекті системи. На цій ранній стадії планування закладається фундамент, на якому згодом належить зводити будівлю проекту всім його технічним учасникам.

## **2. Ієрархічна структура робіт.**

Для успішного керівництва розробкою системи необхідні спеціальні методики, які гарантують, що всі підлягають вирішенню завдання правильно поставлені і розподілені між виконавцями, що для них визначені терміни і забезпечений належний контроль. Одна з найбільш важливих методик - системна організація проектних завдань у формі ієрархічної структури робіт (work breakdown structure - WBS) або, у більш рідкісних випадках, ієрархія структури проекту системи. У цьому форматі всі окремі роботи описуються в термінах їх результатів (виробів і послуг), які повинні бути отримані в ході виконання проекту і представляються у вигляді ієрархічної структури. Побудова ієрархічної структури починається з самого початку етапу розробки концепції і служить вихідною точкою при аналізі альтернатив. На наступних стадіях WBS деталізується і використовується в якості основи для оцінки витрат протягом життєвого циклу системи. У конкурентному середовищі наявність документа з описом WBS часто є вимогою при укладанні договору на розробку системи.

Звичайно за допомогою WBS структурується і визначається все, що має безпосереднє відношення до системи, яка повинна бути розроблена, виготовлена, випробувана, розгорнута і надалі підтримана, в тому числі

технічне і програмне забезпечення, послуги і дані. Ця структура задає каркас, або концептуальну основу, на якій надалі буде реалізований проект.

Для планування і контролю виконання проекту часто застосовується техніка мережевого планування. Мережі складаються з подій і дій, необхідних для виконання проекту. Події еквівалентні точкам прийняття рішень, що означають моменти початку і закінчення дії. Дії уособлюють роботу або задачу, яка повинна бути виконана і, як правило, визначена в WBS. Аналіз критичного шляху - важливий інструмент управління проектом, що дозволяє простежити, як великі елементи системи з'являються у міру розробки їх складових частин. Оцінити можна не тільки вартість, але і тимчасові витрати на кожному кроці. Шлях, яким відповідають завдання, для виконання яких потрібно найбільший час для завершення, називається критичним.

Різниця між цим часом і часом, обчисленим для будь-якого іншого шляху, називається резервом цього шляху. Розраховані критичні шляхи - прямий наслідок застосування WBS, який системний інженер використовує, щоб зрозуміти, які залежності існують між завданнями, встановити пріоритетність різних робіт і графічно представити весь хід виконання програми.

### **3. План управління системною інженерією.**

При розробці складної системи важливо, щоб всі ключові учасники процесу знали не тільки свою сферу відповідальності, а й розуміли, як взаємодіяти один з одним. Як для контролю над системними інтерфейсами потрібна спеціальна документація, так і зв'язки між різними сферами відповідальності і керівними органами повинні бути визначені і підконтрольні. Зазвичай це досягається шляхом підготовки та поширення плану управління системною інженерією (SEMP) або еквівалентного документа. Обов'язок щодо створення такого плану управління інженерною діяльністю покладається на керівників проекту, зайнятих системною інженерією.

Важливість наявності затвердженого плану управління інженерною діяльністю визнана в США в програмах з оборонних закупівель, де підрядник зобов'язаний підготувати SEMP в ході розробки концепції. Основна функція SEMP - гарантувати, що всі численні учасники проекту (керівники робіт по підсистемах, розробники компонентів, інженери-випробувачі, системні аналітики, інженери, які ведуть спеціальну проектну діяльність, субпідрядники і т. д.) розуміють свої обов'язки один перед одним. Це точний аналог функції системної інженерії за визначенням взаємодій між частинами системи таким чином, щоб вони відповідали один одному і працювали безперебійно. Крім того, SEMP грає роль довідника з процедур, які необхідно виконати при вирішенні численних завдань системної інженерії. Місце SEMP в плануванні управління програмою (проектом) показано на рис. 5.2.

SEMP –документ що розвивається, який в початковий момент містить лише загальні принципи, але деталізується і актуалізується в процесі розробки системи. Наявність затвердженого SEMP дозволяє також контролювати повноту вирішення запланованих завдань.

**Елементи типового плану управління системною інженерією**

План управління системної інженерією (systems engineering management plan - SEMP) містить детальний опис того, як повинні бути реалізовані функції системної інженерії в ході розробки системи



Рис.5.2. Місце SEMP в планах управління програмою

Можна рахувати, що існують три види дій:

1. Планування і контроль програми розробки: обмовляються задачі системної інженерії з управління програмою розробки, у тому числі:
  - ◆ опис робіт;
  - ◆ організація;
  - ◆ календарне планування;
  - ◆ аналіз ходу робіт за програмою, проектних рішень та готовності до виконання;
  - ◆ управління ризиком.
2. Процес системної інженерії: описується процес системної інженерії в частині його застосування до розробки системи, у тому числі:
  - ◆ вимоги до функціональних можливостей;
  - ◆ аналіз функцій;
  - ◆ аналіз системи і стратегія прийняття компромісних рішень;
  - ◆ стратегія випробувань та атестації системи.
3. Інтеграція спеціальної інженерної діяльності: описується, яким чином включити в проектування та розробку основної системи питання, що відносяться до спеціальної інженерної діяльності, а саме:
  - ◆ надійність, ремонтпридатність, придатність до використання;
  - ◆ підготовка виробництва;

- ◆ забезпечення безпеки, в тому числі техніку безпеки;
- ◆ ергономічне проектування.

Структура типового SEMP адаптується до конкретної розробки системи, але в загальному випадку може включати наступні розділи:

#### Введення

Предмет, мета, загальний огляд, використовувані документи

#### Планування і управління програмою

Організаційна структура

Сфери відповідальності, процедури, керівні органи

WBS, контрольні точки, терміни

Важливі події в програмі

Аналітичні огляди програми, технічної готовності і готовності до випробувань

Контрольні показники - технічні і за термінами

Інтеграція з програмою розробки, плани організації взаємодії

#### Процес системної інженерії

Призначення, наочний опис системи

Аналіз вимог і функцій

Вивчення компромісів (аналіз альтернатив)

Аналіз і планування технічних інтерфейсів

Дерево специфікацій / специфікації

Математичне та імітаційне моделювання

Планування випробувань

Аналіз логістичного забезпечення

Інструменти системної інженерії

#### Комплексування системи

Проект / плани комплексування

Спеціальна проектна діяльність

Аналіз сумісності та взаємних перешкод

Вивчення можливості виробництва

## 4. Управління ризиками.

Розробка нової складної системи по суті своїй передбачає набуття знань про передові, але ще не до кінця розроблені пристрої і процеси. Це необхідно, щоб грамотно керувати проектуванням системи і в результаті отримати виріб, який виконує поставлене перед ним завдання надійно і з прийнятними витратами. Проте на кожному кроці можливі несподіванки, які чреваті ризиком неповноти функціональних можливостей, недостатньою стійкістю до впливів навколишнього середовища, непридатності до виробництва і цілим рядом інших неприйнятних наслідків, які можуть вимагати внесення змін, негативно позначаються на вартості і терміні завершення програми. Одне з найскладніших завдань, що стоять перед системним інженером, - прокласти курс, який веде до максимальних результатів при мінімальних ризиках.

На початку розробки невизначеності (а значить, і ризики) присутні всюди. Чи реалістичні виявлені вимоги до функціональних можливостей? Чи

залишаться вони такими протягом усього терміну служби нової системи? Чи будуть ресурси, необхідні для розробки і виготовлення системи, доступними, коли в них виникне потреба? Чи буде передова технологія, необхідна для досягнення заявлених функціональних можливостей, працювати, як очікується? Чи вдасться реалізувати передбачуваний рівень автоматизації виробництва? Чи не буде організація розробки страждати від затримок?

У завдання системної інженерії входить облік подібних можливостей і управління розробкою таким чином, щоб мінімізувати (пом'якшити) їх вплив у разі виникнення. Методологія, яка застосовується для виявлення та мінімізації ризиків при розробці системи, називається «управління ризиком». Управління ризиком слід вести з самого початку розробки системи і далі на всьому її протязі.

### **Зниження ризиків протягом життєвого циклу системи**

Зниження ризиків програми - безперервний процес, триває протягом усього життєвого циклу системи. Наприклад, на етапі аналізу потреб знижується ризик приступити до розробки системи, в якій немає конкретної необхідності. На етапі дослідження знижується ризик включення несуттєвих і нереалістичних вимог до показників функціонування системи. На етапі опису системи вибирається концепція, в якій задіяні технічні підходи, які не є ні занадто незрілими, ні надмірно витратними, але мають найкращі шанси задовільнити всі поставлені перед системою цілі<sup>90</sup>.

На рис. 5.3 схематично показано, як ризик програми розробки гіпотетичної системи (в довільно взятих одиницях виміру) зменшується в міру просування по етапах життєвого циклу. По горизонтальній осі відкладається час в прив'язці до етапів розробки. Тут же представлено типічний графік відносних трудових витрат на кожному етапі.

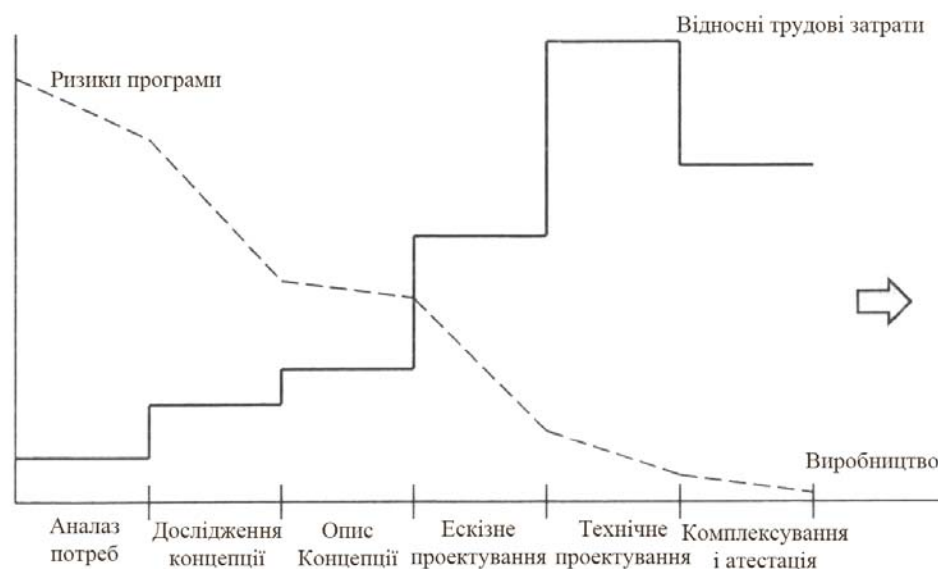


Рис.5.3. Зміни ризиків програми і трудових затрат в ході розробки системи.

Рис. 5.3 ілюструє кілька важливих принципів.

1. У міру наближення до завершальних етапів розробки обсяг інвестицій у програму зазвичай різко збільшується. Для забезпечення

продовження програми ризик невдачі необхідно відповідно знижувати так, щоб фінансові ризики залишалися на прийнятному рівні.

2. На початкових стадіях програми, коли приймаються основні рішення про вимоги до системи і її концепції, ризик знижується особливо відчутно. Це показує, наскільки важливо докласти максимум зусиль саме на цих визначальних етапах.
3. Найбільше зниження ризику зазвичай спостерігається на етапах дослідження концепції та ескізного проектування. У ході дослідження концепції закладається міцний концептуальний фундамент для формування підходу до розробки системи та побудові її архітектури. У ході ескізного проектування освоюються нові передові технології, щоб гарантувати, що з їх допомогою можна буде досягти необхідних робочих характеристик.
4. До моменту завершення розробки та готовності системи до виробництва та постачання залишковий рівень ризику повинен бути дуже низький - тільки в цьому випадку можна розраховувати на успіх.

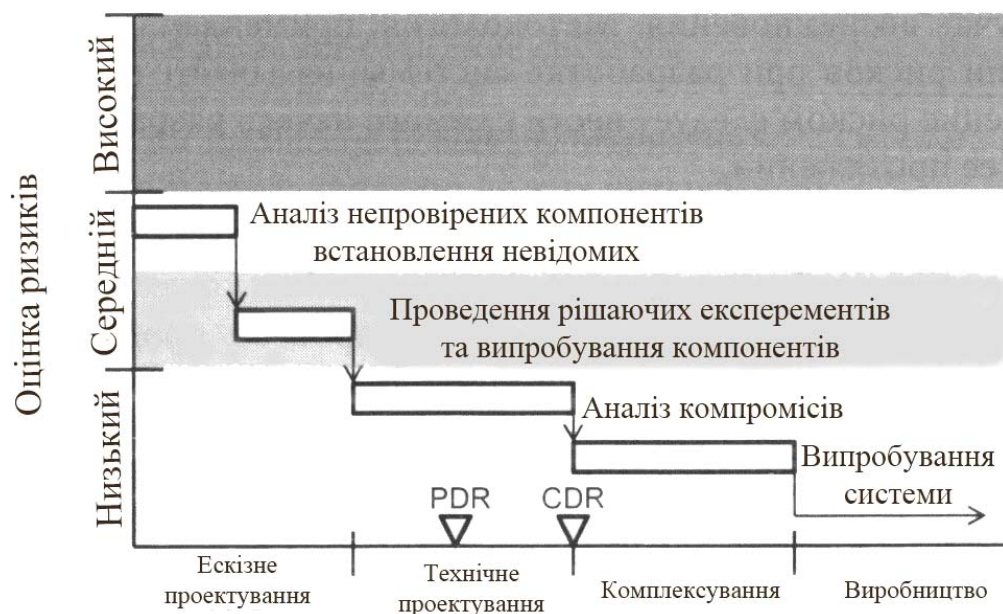


Рис.5.4. Приклад водопадів ризиків ( PDR-попередній аналіз проектних рішень, CDR-критичний аналіз проектних рішень).

Спадний характер кривої ризику обумовлений тим, що в міру розробки невизначеності (невідомі), які і є причиною непередбачуваних і небажаних подій, а значить, і ризиків, систематично усуваються або знижуються в результаті аналізу, експериментів, іспитів і змін напрямків робіт. Варіант цієї кривої іноді називається «водоспад пом'якшення ризиків» (рис. 5.4). Висхідний характер кривої трудових витрат означає, що витрати на кожному наступному етапі розробки системи збільшуються, оскільки у міру переходу від розробки концепції до проектування, а потім до комплексування та атестації обсяг робіт постійно зростає.

#### Складові частини управління ризиком

Управління ризиком формально визнано в стандартах системної інженерії і особливо в програмах державних закупівель. Передбачається, що кожна програма повинна включати план управління ризиком. При розробці великих систем для управління ризиком повинна бути створена окрема організація зі своїм штатом, базою даних, системою звітності та незалежній експертизі, причому управління ризиком має бути поширене на всі етапи виробництва, експлуатації та супроводу.

**Роль системної інженерії.** Завдання оцінки ризику (і наступна за нею задача управління ризиком), очевидно, потрапляє в сферу відповідальності системної інженерії. Пояснюється це тим, що для винесення експертних оцінок необхідна широта знань про характеристики системи і застосовуваних у ній технологіях, що виходить за рамки знань фахівців з проектування, а також тим, що оцінка критичності ризику проводиться на рівні системи і програми в цілому. Отже, процес оцінки ризиків допомагає системному інженеру виявляти ті особливості системи, які повинні бути осмислені найбільш ретельно і підняті на такий рівень зрілості проектування, який характерний для повномасштабної інженерно-технічної розробки.

### **Пом'якшення ризиків**

Нижче перераховані найбільш поширені методи вирішення проблем, пов'язаних з ризиками, в порядку зростання серйозності виявленого ризику:

1. Інтенсифікація аналізу процесу розробки з технічної та управлінської точок зору.
2. Особливий контроль над розробкою попередньо виділених компонентів.
3. Особливий аналіз та випробування критичних елементів конструкції.
4. Швидке створення дослідних зразків та облік результатів їх випробувань в подальшій розробці.
5. Розгляд можливості послабити критичні технічні вимоги.
6. Організація паралельної розробки на випадок невдачі в основному варіанті.

**Технічний і управлінський аналіз.** Формальному аналізу може бути піддана ціла підсистема, але найбільш глибоко розглядаються ті аспекти проекту, які вважаються найважливішими. Системні інженери повинні подбати про те, щоб істотні ризики були представлені повністю і всебічно обговорені з метою привернути увагу керівництва і направити ресурси на проблеми, які потребують додаткових сил і засобів. Завдання полягає в тому, щоб вирішити проблеми якомога раніше, тому важливо не приховувати правду про виниклі і очікувані труднощі. Процес аналізу проекту докладніше буде розглянуто в главі 12 (розділ 12.4).

**Контроль над розробкою попередньо виділених компонентів.** Регулярні планові аналізи проекту проводяться не настільки часто і є недостатньо детальними, щоб належним чином контролювати відомі зони ризику. Будь-якій виявленій проблемній області слід присвоїти особливий статус, завдяки якому вона досить часто розглядається на нарадах і знаходиться під контролем спеціально призначених старших конструкторів і системних інженерів. Там, де це доречно, до процесу слід залучати сторонніх консультантів. Необхідно

скласти план пом'якшення ризиків і неухильно слідувати йому, поки проблеми не будуть вирішені.

**Особливий аналіз та випробування.** Для компонентів, в конструкції яких виявлені проблеми, не вирішені на етапі ескізного проектування, потрібно провести додатковий аналіз і, якщо знадобиться, виготовити і випробувати їх, щоб отримати достатньо даних для валідації обраного технічного підходу. На обробку результатів аналізу та випробувань потребується виділити додаткові ресурси і внести зміни в графік виробництва.

**Швидке прототипування.** Для неперевіраних компонентів, результати аналізу і обмежених випробувань яких не можуть служити достатньою підставою для схвалення проектних рішень, може виявитися необхідно сконструювати і випробувати прототипи / дослідні зразки, щоб переконатися в придатності цих компонентів. Зазвичай такі дії приймаються на етапі ескізного проектування, але буває, що проблема в цей момент ще не виявлена; крім того, іноді вжитими заходами невдається її вирішити.

**Ослаблення надмірних вимог.** Досвід показує, що спроба задовільнити всім спочатку сформульованим вимогам часто завершується невдало: отримати на практиці повне рішення не вдається, і доводиться вносити корективи в деякі вимоги до показників функціонування або сумісності. Цю можливість слід розглянути, якщо всі зусилля для задоволення деякої вимоги повною мірою призводять до вирішення, виявляються надмірно складними, дорогими, ненадійними або ще в якомусь відношенні не відповідними з практичної точки зору. Подібна проблема знаходиться у винятковому віданні системної інженерії, оскільки необхідно взяти до уваги відразу всі фактори: функціональні можливості, вартість і терміни. Вдаватися до цього засобу слід тільки в крайніх випадках, але і відкладати його до моменту, коли в марних спробах задовольнити вимогу буде витрачено непропорційно багато часу і ресурсів, теж не варто.

**Резервні альтернативи.** Розробка альтернативних підходів до проектування найбільшою мірою стосується компонентів, в яких застосовується нова технологія з негарантованим результатом. У таких випадках на етапі ескізного проектування слід продумати резервні підходи, на які можна буде переключитися, якщо нова конструкція не виправдає очікувань. Майже завжди подібні альтернативи призводять до погіршення показників функціонування, підвищення вартості або ще якихось недомог у порівнянні з обраним підходом, але завдяки більш консервативному проекту вони мають більше шансів на успіх.

### **Команда проектування системи**

Управління та координація робіт в будь-якій великій програмі вимагають тісного співробітництва однієї або декількох груп ключових фахівців, що мають загальну думку з питань організації програм інженерно-технічних розробок. Команда проектування в проекті розробки складної системи повинно включати наступних фахівців:

- ◆ системний інженер;
- ◆ інженери по основних підсистемах;
- ◆ програмний інженер;



- ◆ фахівець із забезпечення;
- ◆ інженери-випробувачі,
- ◆ представник замовника;
- ◆ інженери по спеціальному проектуванню та фахівці з паралельної організації робіт.

Представник замовника відстоює вимоги до системи. Перевага командного підходу - в тому, що він формує корпоративний дух, посилює мотивацію учасників і розширює їх кругозір в частині стану розробки пов'язаних частин системи і зустрічних там проблем. В результаті члени команди починають усвідомлювати себе причетними до системи в цілому, а не обмежуються своєю вузькою сферою відповідальності, як буває в багатьох організаціях. Тому реакція на несподівані проблеми і інші зміни в програмі виявляється більш ефективною.

У конкретній ситуації керівництво розробкою системи має бути вибудовано з урахуванням організаційної структури головного підрядника і рівня залучення замовника до процесу. Найбільш важливі наступні загальні фактори:

1. Лідерські якості керівника команди.
2. Представленість осіб, наділених ключовими повноваженнями.
3. Участь основних інженерно-технічних колективів.

Без енергійного лідера члени команди проектування системи зіб'ються зі шляху і розбредуться по своїх кутках. Якщо з якоїсь причини особа, призначена системним інженером проекту, не володіє необхідними лідеру особистими якостями, то цю роль має взяти на себе головний інженер проекту або старший системний інженер.

## **КОРОТКІ ВИСНОВКИ**

### **Управління розробкою системи і ризиками**

Системна інженерія - частина управління проектом; у її сферу відповідальності входять технічне керівництво, комплексування системи і координація роботи технічних підрозділів.

### **Ієрархічна структура робіт**

У обов'язку системного інженера входять також участь в розподілі ресурсів, постановці завдань і взаємодії із замовником і на початковому етапі - розробка WBS, ієрархічно організованої безлічі завдань, що має на меті розбити усю майбутню роботу на елементи роботи, що послідовно зменшуються. Це створює основу для планування, складання бюджету і моніторингу, а також дозволяє здійснювати оцінку і контроль витрат.

Одним з ключових інструментів планування програми є метод критичного шляху. Цей метод, ґрунтований на елементах роботи, включених в WBS, дозволяє побудувати мережу послідовних дій. За допомогою аналізу цієї мережі системний інженер і керівник програми можуть ідентифікувати шляхи, для завершення яких вимагається більше всього часу.

## **План управління системною інженерією**

У плані управління системною інженерією SEMP визначені шляхи виконання усіх завдань системної інженерії, у тому числі виявлені ролі і сфери відповідальності усіх учасників.

## **Управління ризиком**

Управління ризиком - одна з найбільш складних проблем, що стоять перед системною інженерією, оскільки будь-яка розробка нової системи таїть в собі невизначеності і ризики. Зниження ризиків програми - це безперервний процес, що триває упродовж усього життєвого циклу; крім того, ризик необхідно знижувати у міру збільшення об'єму що вкладаються в програму фінансових засобів.

Для здійснення управління ризиком важливо скласти план управління ризиком. В процесі оцінки ризику його значущість визначається в термінах вірогідності виникнення і критичності ризику (впливу і наслідку у разі реалізації ризику).

Для пом'якшення ризику в критичних ділянках можна застосовувати наступні методи: управлінський аналіз, особливий інженерний контроль (наприклад, над розробкою заздалегідь виділених компонентів), особливий аналіз і випробування, швидке створення дослідних зразків, послаблення надмірно строгих вимог, організація паралельної розробки на випадок невдачі.

## **Організація системної інженерії**

Організація системної інженерії охоплює різні дисципліни і задіяні в проекті організації, але також адаптується до організаційної структури компанії. Тому системна інженерія повинна ефективно довести до відома усіх зацікавлених сторони "що, коли і чому", а також надавати усім учасникам технічні звіти. У великих програмах системна інженерія підтримується фахівцями з системного аналізу.

Для великомасштабних програм необхідно створювати офіційно затверджені команди проектування системи, сфера відповідальності яких охоплює основні підсистеми, субпідрядників і результати роботи програмних інженерів. У ці команди входять представники служб інженерного забезпечення і випробувальної організації, а також, як правило, інженери, зайняті спеціальним проектуванням, і фахівці з паралельної організації робіт. При необхідності можуть бути включені і представники замовника. Ключова роль системної інженерії в таких командах - концентрувати їх увагу на успіху підприємства в цілому.

## Лекція 6. АНАЛІЗ ПОТРЕБ

Основна мета етапу аналізу потреб в життєвому циклі системи - ясно і переконливо продемонструвати, що дійсно існує практична потреба (чи потенційний ринок) в новій системі або в принциповій модернізації існуючої і що є практично здійснений підхід до задоволення виявленої потреби з розумними витратами і прийнятним рівнем ризику.

Частково досягти цього можна, подумки сконструювавши щонайменше одну уявну систему, про яку можна сказати, що вона має функціональні можливості, що дозволяють задовольнити знову усвідомлену потребу, а потім описати цю систему настільки детально, щоб переконати осіб, що приймають рішення, що вона технічно реалізовується і може бути розроблена і зроблена з прийнятними витратами.

Коротше кажучи, мета цього процесу - пред'явити переконливі і обгрунтовані аргументи на підтримку виявлених потреб і створити "образ успіху" в умах тих, хто може дати добро на початок розробки нової системи.

### **Місце етапу аналізу потреб в життєвому циклі системи**

Точний момент початку активної розробки нової системи частенько не так-то просто визначити. Пов'язано це з тим, що на самих ранніх етапах виникнення нової системи діяльність зазвичай носить дослідницький характер і по суті своїй неформальна, т. е. не має ні сформованої організаційної структури, ні конкретних цілей, ні погодженого графіку робіт. Сенс подібної діяльності - в тому, щоб, ґрунтуючись на оцінці існування дійсної потреби в новій системі і технічно здійсненого підходу до її реалізації, визначити виправдано або ні виділення сил і засобів на її створення.

При розробці системи, обумовленої появою нової технології, як правило нової комерційної системи, етап аналізу потреб вважається частиною стадії розробки концепції. Проте і в цьому випадку повинні виконуватися аналогічні дії, наприклад аналіз ринку, оцінка конкуруючих продуктів і оцінка недоліків існуючої системи в порівнянні з пропонованою. Вказана діяльність спрямована на те, щоб встановити, чи існує справжня нужда (потенційний ринок) в продукті, який передбачається розробити. Тому в подальшому обговоренні ми не проводитимемо відмінності між системами, розробка яких стимулюється потребами або наявністю нових технологій, якщо тільки це не буде обумовлено явно.

Місце етапу аналізу потреб в життєвому циклі системи показане на рис. 6.1. Як бачимо, входами є експлуатаційні недоліки і/або технологічні можливості, що відкрилися. На виходах, що є одночасно входами наступного етапу, - дослідження концепції, - ми маємо результати оцінки експлуатаційної ефективності, що дозволяють визначити, яких результатів повинна досягти нова система, для того, щоб задовольнити



Рис.6.1 Етап аналізу потреб у життєвому циклі системи.

**Аналіз цілей.** Під "аналізом цілей" розуміється процес формування і уточнення цілей, що стоять перед системою. Зазвичай результатом цієї роботи є дерево цілей, в якому цілі верхнього рівня, - одна або невеликий набір - розбиваються на безліч первинних і вторинних цілей. Таке дерево зображене на рис.6.2. Робіть декомпозицію до тих пір, поки не дійдете до мети, що допускає верифікацію, або доки не упіймаєте себе на тому, що почали визначати функції системи. У цей момент зупинитися. На малюнку функції показані світло-сірими прямокутниками - вони не увійдуть до дерева цілей. Наш досвід показує, що у більшості випадків дерева цілей мають один-два рівня; немає нужди поглиблювати їх далі.



Рис.6.2. Структура дерева цілей.

Як приклад аналізу цілей розглянемо новий автомобіль. Припустимо, що автовиробник хоче сконструювати новий пасажирський автомобіль, який зможе позиціонувати на ринку як "зелений", тобто екологічний. Розуміння поставлених цілей дозволило б визначити пріоритети кінцевої конструкції.

Тому керівництво компанії ініціює дослідження цілей. Аналіз цілей змусить компанію - як керівництво, так і інженерно-технічний персонал - оцінити і вирішити, що важливо при розробці нової системи. Отже, має сенс витратити деякий час, сили і засоби на визначення спільних цілей системи. До того ж наявність лаконічного формулювання, з яким усі згодні, допоможе команді розробників сконцентруватися на тому, що їм належить зробити.

Завершальний і найважливіший крок застосування методу системної інженерії - методична перевірка достовірності результатів, отриманих на попередніх кроках. У разі аналізу потреб в процесі валидації необхідно прийняти рішення про ваговитість аргументів на користь існування потреби в новій системі і можливості задовольнити цю потребу з розумними витратами і прийнятним ризиком.

### **Модель експлуатаційної ефективності**

На стадії розробки концепції аналітична діяльність, що ставить метою оцінити, якою мірою від цієї концепції системи можна чекати задоволення заданих вимог призначення, називається аналізом експлуатаційної ефективності. У основі цієї роботи лежить математична модель умов експлуатації і варіанту концепції системи, що підлягає розгляду.

В процесі аналізу ефективності умови експлуатації моделюються з використанням набору сценаріїв - передбачуваних дій, що містять діапазон можливих подій, на які система повинна реагувати. Зазвичай спочатку вибираються сценарії, що представляють найбільш вірогідні ситуації, а потім складніші випадки для перевірки граничних умов, заданих у вимогах призначення. Для кожного сценарію в ролі критерію оцінки виступає те, наскільки прийнятними виявляються відгуки системи в термінах результатів функціонування. Аналіз ефективності повинен включати не лише різні режими експлуатації системи, але і операції, не пов'язані безпосередньо з експлуатацією: транспортування, зберігання, установку, обслуговування і ремонт, а також і логістичне забезпечення. В сукупності усі значимі вимоги призначення і обмеження мають бути враховані в сценаріях практичного використання і в супровідній документації, що містить опис умов експлуатації.

Нарешті, описаний вище аналіз ефективності в першу чергу спрямований на вирішення питання про те, чи являється концепція системи, що оформилася в процесі опису функціональних можливостей або способу фізичної реалізації, що 1) здійсненою і 2), що відповідає практичним цілям, досягнення яких потрібне для задоволення даної потреби. Це припущення не завжди надійне, особливе у разі розробки систем, обумовлених технічним прогресом, коли потенційна сфера застосування системи нова і прийняття подібного припущення залежить від багатьох невідчутних чинників. Одна така ситуація, для якої існують сотні конкретних прикладів, - впровадження засобів автоматизації в сферу, де раніше застосовувалася переважно ручна праця (система резервування авіаквитків - приклад успішної реалізації великого проекту подібного роду). Валидація потреби в такій системі вимагає технічного аналізу разом з аналізом функціонування і аналізом ринку з метою обліку безлічі складних чинників, які можуть вплинути на вирішення питання про

прийнятність автоматизованої системи і вірогідний прибуток від її впровадження.

У таких складних випадках від валідації можна чекати лише дуже предвари-тельних результатів, які мають бути підтвержені солідними дослідницькими розробками і експериментами. Але навіть попередня валідація здатна виявити більшість критичних місць і іноді показує, що шанси на задоволення деяких передбачуваних потреб занадто проблематичні і не виправдовують великих інвестицій при поточному рівні розвитку технології.

### **Сценарії практичного використання**

Логічно послідовний метод розробки вимог призначення полягає в тому, щоб запропонувати ряд сценаріїв, які в сукупності представляють усю гамму ситуацій, очікуваних в процесі практичного використання. У основу сценаріїв має бути покладене ретельне вивчення умов застосування (експлуатації), бесіди з досвідченими користувачами попередньої системи і її аналогів, а також усебічне осмислення минулого досвіду і продемонстрованих недоліків існуючих систем. Особливо важливо правильно розставити призначені для користувача пріоритети відносно необхідних удосконалень, і, зокрема, тих, які здаються найбільш важкими для реалізації. Хоча склад і зміст сценаріїв істотно залежать від додатка, можна виділити п'ять основних компонентів, властивих майже усім сценаріям.

1. **Призначення і цілі.** Сценарій повинен в цілому виявити цілі, пов'язані з можливістю використання системи за призначенням, а також місце і роль системи (чи систем) в досягненні цих цілей. В деяких випадках цей компонент не залежить від системи, тобто роль якоїсь окремої системи не представлена, а даний лише загальний опис основного завдання і переслідуваних цілей. У разі комерційної системи призначення може полягати в за-хватові частки ринку, у разі державної - в наданні деякого набору послуг виборцям, а у разі військової - в отриманні контролю над якоюсь конкретною фізичною установкою.

2. **Архітектура.** Сценарій повинен виявити базову архітектуру системи. Сюди входить перелік систем, організацій і основні відомості про структуру. Якщо є зведення про методи управління, їх слід включити. Цей компонент повинен також містити загальні відомості про інтерфейси системи і опис інформаційно-технологічної інфраструктури. По суті, надається опис доступних ресурсів. Для комерційної системи в цьому сценарії описуються ресурси корпорації, для державної - організації і установи, що беруть участь в рішенні задачі, для військової - військові формування з усім їх оснащенням.

3. **Фізичне оточення.** Сценарій повинен виявити оточення, в якому він розгортається. Сюди входить як фізичне оточення (наприклад, рельєф місцевості, погодні умови, транспортна мережа і енергосистема), так і ділове (наприклад, період спаду або зростання економіки). У цьому розділі описуються "нейтральні" об'єкти. Зокрема, можна описати замовників і їх характеристики або нейтральні країни і їх ресурси.

4. **Конкуренція.** Сценарій повинен виявити наявних конкурентів. Це можуть бути сторони, прямо протидіючі успішному рішенню завдань, на-приклад хакер, або інший "супротивник", або ваш конкурент на ринку, або

зовнішні чинники, що роблять вплив на ваших замовників. Сюди ж можна включити стихійні лиха, такі як цунамі або ураган.

**5. Загальна послідовність подій.** Сценарій повинен виявити загальну послідовність подій в контексті завдання. Ми свідомо використовуємо розпливчате слово "загальну". Сценарій повинен залишати свободу дій учасникам. Оскільки ми застосовуємо сценарії для складання вимог призначення і оцінки ефективності системи, потрібна можливість змінювати різні параметри і події у рамках загального опису сценарію. Сценарій - не жорстко прописаний текст ролі; це інструмент аналізу, а не окови, що утрудняють розробку системи. Тому зазвичай в сценаріях описується лише загальна послідовність подій, а деталі оставляються аналітикові, який цим сценарієм користується. Іноді в сценарії може бути представлена детальна послідовність подій аж до деякого моменту часу, з якого розпочинається аналіз; після цього моменту дії можуть мінятися.

Сценарій може включати і значно більше, все залежить від додатка і переслідуваної мети. Об'єм сценарію також може мінятися: від короткого графічного опису, що складається з декількох малюнків, до сотень сторінок тексту з даними.

#### **Валідація здійсненності**

Аналіз ефективності нерозривно пов'язаний з функціональними якість системи і тому сам по собі не може використовуватися для валідації здійсненності фізичної реалізації системи. Особливо це справедливо у разі, коли для забезпечення певних функціональних можливостей застосовується неперевірена технологія.

Непрямий підхід до валідації здійсненності полягає в тому, щоб привести переконливу аргументацію на основі аналогії із вже наявними додатками пропонованої до використання техніки. Такий підхід може виявитися адекватним за умови, що узяті як приклад застосування дійсно представляє характерні особливості нової системи. Важливо, проте, щоб порівняння було кількісним, а не тільки якісним і доводило можливість досягнення заданих характеристик в результаті застосування пропонованої технології.

Прямий підхід до валідації здійсненності нової фізичної реалізації полягає в проведенні експериментальних досліджень даної технології з метою продемонструвати, що прогнозовані характеристики дійсно досяжні на практиці. Такі "вирішальні експерименти" для вивчення нових концепцій реалізації часто ставляться на ранні етапи програми.

Оскільки рішення про початок фактичної розробки системи ще не прийняте, на етапі аналізу потреб ресурси нерідко у край обмежені, і для виконання процесу валідації їх бракує. Тому якість процесу валідації у величезному ступені залежить від досвіду і винахідливості системних інженерів. Чинник досвіду особливо важливий, оскільки ця робота залежить від знання умов експлуатації, попередньої системи, досліджень, що раніше проводилися, і аналізів, технологічної бази і методів системного аналізу і системної інженерії.

## **КОРОТКІ ВИСНОВКИ**

### **Виникнення нової системи**

Завдання етапу аналізу потреб - ідентифікувати дійсно існуючу потребу в новій системі і розробити здійснений підхід до її задоволення. Такий стимульований потребами похід до розробки систем характерний для більшості оборонних і інших державних програм і, як правило, є результатом обмеженості можливостей існуючої системи. Розробка подібного типу вимагає використання технічного підходу, що гарантує здійсненність з прийнятними витратами.

Принципово інший підхід - це розробка системи, що стимулює технічним прогресом. Він характерний для розробки більшості комерційних систем і є результатом технологічної можливості, що відкрилася, для кращого задоволення якої-небудь потреби. При розробці такого типу необхідно продемонструвати доцільність і можливість збуту.

На етапі аналізу потреб виконуються наступні види діяльності :

- ◆ *системний аналіз* - осмислення потреби в новій системі;
- ◆ *аналіз функціонування* - визначення функцій, необхідних для того, щоб система могла успішно використовуватися за призначенням;
- ◆ *оцінка здійсненності* - виявлення здійсненого підходу до реалізації;
- ◆ *валідація потреб* - демонстрація економічної ефективності.

### **Системний аналіз**

Проводяться дослідження і аналіз для формулювання і осмислення практичних потреб в новій системі. В результаті будується дерево цілей, що описує ієрархію очікувань і результатів.

### **Аналіз функціонування**

Ідентифікуються і упорядковуються початкові функції системи, які дозволять досягти заявлених практичних цілей. Ці функції перевіряються шляхом аналізу і презентації користувачам і зацікавленим сторонам.

### **Оцінка здійсненності**

Приймається рішення про вибір підходу до розробки системи, воно доводиться до відома зацікавлених сторін, і дається приблизний кошторис витрат. Крім того, озвучується початкова здійснена концепція. Нарешті, починається розробка вимог призначення.

### **Валідація потреб**

Раніше перевірений набір практичних потреб тепер піддається валідації шляхом аналізу експлуатаційної ефективності, зазвичай на декількох рівнях аналітичної піраміди. Концепції системи, що задовольняють практичним потребам, оцінюються за допомогою погоджених показників ефективності і дають можливість отримати уявлення про повний життєвий цикл системи.



## Лекція 7.

### ДОСЛІДЖЕННЯ КОНЦЕПЦІЇ. КОНЦЕПТУАЛІЗАЦІЯ.

Основна мета етапу дослідження концепції в сенсі визначення, прийнятого в цій книзі, полягає в тому, щоб перейти від функціонального погляду на систему, етапу аналізу потреб, що є результатом, до інженерного погляду, необхідному для визначення концепції і подальших етапів розробки. В результаті має бути закладена явна основа, що піддається кількісній оцінці, для вибору прийнятної функціональної і фізичної концепції системи, а згодом і для керівництва їх перетворенням у фізичну модель системи. Проте слід пам'ятати, що вимоги до показників функціонування - це інтерпретація, а не заміна вимог призначення.

При розробці вимог до показників функціонування, як і у разі вимог призначення, необхідно одночасно розглядати концепції системи, які могли б задовольнити цим вимогам. Причому для упевненості в тому, що вимоги до показників функціонування досить широкі, щоб випадково не обмежити спектр можливих конфігурацій системи, слід досліджувати не одну, а декілька можливих концепцій.

На рис.7.1 показано місце етапу дослідження концепції в загальному процесі розробки системи. Видно, що високорівневі вимоги призначення є результатом аналізу потреб, в ході якого встановлюються обґрунтованість потреб і здійсненність програми розробки у рамках заданих обмежень.

На виходах етапу дослідження концепції виникає набір вимог до показників функціонування аж до рівня підсистем, а також ряд можливих концепцій того, як може бути побудована система, яка, згідно з проведеним аналізом, в змозі задовольнити цим вимогам.



Рис.7.1. Етап дослідження концепції в життєвому циклі системи.

Хоча формально етап дослідження концепції має чітко визначені почало і кінець, про багато допоміжних видів діяльності цього не скажеш. Наприклад, дослідницька розробка передових технічних підходів або кількісна оцінка складних умов експлуатації часто починаються до початку, а завершуються після закінчення заданих термінів цього етапу, будучи підкріплені незалежними НІОКР або іншими ресурсами, що не відносяться безпосередньо до

проекту. Крім того, значна частина попередньої роботи за визначенням концепції зазвичай має місце до офіційного початку цього етапу.

Дії на етапі дослідження концепції і їх взаємозв'язку - це результат застосування методу системної інженерії. Нижче приведений короткий огляд цих дій; назви чотирьох узагальнених кроків показані в дужках.

**1. Аналіз вимог призначення (аналіз вимог).** Типові дії включають:

- ◆ аналіз встановлених вимог призначення в плані їх цілей;
- ◆ уточнення або доповнення (залежно від необхідності) вимог для того, щоб підкреслити специфіку, незалежність і несуперечність різних цілей, визначених для системи, і тим самим гарантувати сумісність із спорідненими системами, а також надати іншу інформацію, необхідну для повноти і цілісності вимог.

**2. Визначення вимог до показників функціонування (функціональний опис).** Типові дії включають:

- ◆ перехід від вимог призначення до функцій системи і її підсистем;
- ◆ визначення значень показників функціонування, які потрібні для задоволення встановлених вимог призначення.

**3. Дослідження концепції реалізації (опис фізичної реалізації).** Типові дії включають:

- ◆ дослідження сукупності здійснених концепцій і технологій реалізації, що пропонують різні потенційно корисні додаткові можливості;
- ◆ розробку функціональних описів і ідентифікацію відповідних компонентів системи для більшості випадків, що представляють інтерес;
- ◆ визначення необхідного і достатнього набору показників функціонування, що відбивають функції, істотні для задоволення вимог призначення.

**4. Валідація вимог до функціонування (валідація проектних рішень).** Типові дії включають:

- ◆ проведення аналізу ефективності, мета якого - визначити вимоги до функціонування, що охоплюють увесь спектр концепцій передбачуваної системи;
- ◆ валідацію погодження цих вимог з поставленими практичними цілями і при необхідності уточнення вимог.

Як і на усіх етапах процесу розробки системи, насамперед треба в усіх подробицях зрозуміти і, якщо необхідно, уточнити і доповнити вимоги до системи, визначені на попередньому етапі (в даному випадку - вимоги призначення). При цьому важливо бути уважним і не пропустити вад, які часто є присутніми в спочатку певних вимогах призначення. Ми застосовуємо загальний процес, що іменується аналізом вимог, щоб ідентифікувати і виявити вимоги до показників функціонування, синтезувати і мінімізувати початкові набори вимог і, нарешті зробити валідацію остаточного набору вимог. Проте основна робота доводиться на етап дослідження концепції, де вимоги призначення перетворюються у вимоги до показників функціонування з тими, що допускають вимір пороговими значеннями. Ці вимоги до показників функціонування ляжуть в основу договорів між замовником і розробником і тому мають бути сформульовані точно і лаконічно.

На рис.7.2 показаний загальний процес розробки вимог. Зрозуміло, його необхідно адаптувати стосовно конкретної ситуації. На першому етапі створюється набір вимог. Це рідко відбувається на порожньому місці - зазвичай існує джерело потреб. На етапі дослідження концепції вже відомі практичні потреби і вимоги призначення, проте, як правило, вони виражені на мові і в контексті оператора або користувача. Тому їх необхідно перетворити в набір вимог, що відносяться не посередньо до системи і описують її характеристики.

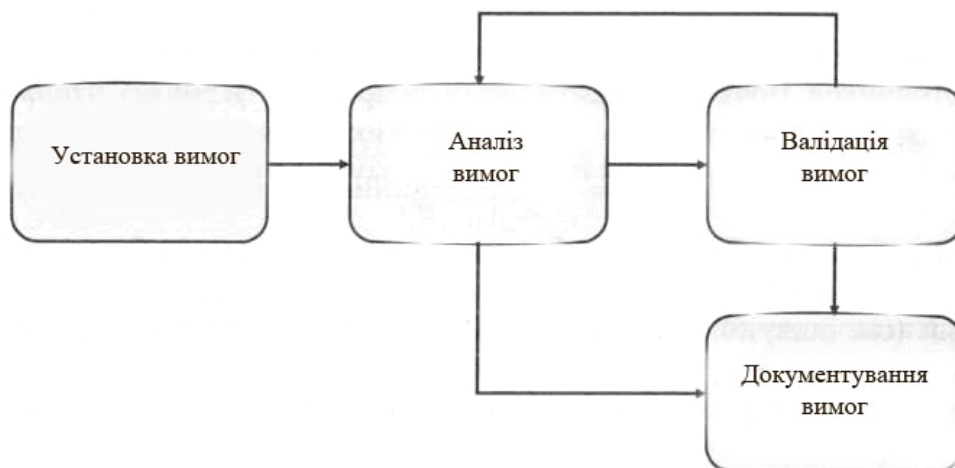


Рис.7.2. Простий процес розробки потреб.

Дослідження потенційно можливих конфігурацій системи робиться на функціональному і на фізичному рівнях. Спектр різних функціональних підходів, які породжують поведінку, що відповідає вимогам призначення системи, взагалі кажучи, набагато вужчий, ніж спектр різних фізичних реалізацій. Проте частенько є способів досягнення заявлених практичних можливостей системи, що дещо помітно відрізняються один від одного. Важливо, щоб показники функціонування, які характеризують ці різні функціональні підходи, були враховані при встановленні меж вимог до показників функціонування системи.

Щоб було простіше ідентифікувати функції системи, що відповідають за її робочі характеристики, згадаєте про класифікацію функціональних середовищ: сигнали, дані, матеріали, енергія. Даний процес ставить за мету отримання відповідей на наступні питання:

1. Чи існують практичні цілі, для досягнення яких потрібні датчики або комунікації? Якщо так, то мають бути присутніми функції, що відносяться до введення, обробки і виведення сигналів.

2. Чи потрібна системі інформація для управління її роботою? Якщо так, то як породжуються, обробляються, зберігаються і взагалі використовуються дані?

3. Чи потрібні для роботи системи конструкції або машини для забезпечення матеріалами або зберігання і обробки матеріалів? Якщо так, то які операції пов'язані зі зберіганням, забезпеченням, обробкою і маніпулюванням матеріальними елементами?

4. Чи потрібно системі енергію для приведення в дію, пересування, живлення або іншого виробництва необхідного руху або тепла?

Далі, функції можна віднести до однієї з трьох категорій : введення, трансформація і виведення. Функції введення відносяться до процесів виявлення і подання сигналів, даних, матеріалів і енергії в систему; функції виведення - до процесів інтерпретації, відображення, синтезу і виведення сигналів, даних, матеріалів і енергії з системи; функції трансформації - до процесів перетворення входів у виходи, яке б не було функціональне середовище. Зрозуміло, в складних системах кількість функцій трансформації може бути дуже велика, причому декілька трансформацій можуть з'єднуватися в ланцюжки. На рис.7.3 зображена концепція такої двомірної конструкції : категорії функцій у поєднанні з функціональним середовищем.

При побудові початкового переліку функцій корисно ідентифікувати входи і виходи (як описано в лекції 3). Цей перелік дозволяє інженерові відразу ж скласти перелік функцій введення і виведення. Знаючи входи і виходи системи, простіше ідентифікуватиме функції трансформації.

Як приклад розглянемо стандартну кавоварку (найпростішу модель), усвідомлюючи те, що це зовсім не складна система. В результаті спостережень аналітик може ідентифікувати наступні необхідні входи:

- ◆ *сигнали*: команди користувача (ми просто назвемо їх "вкл". і "викл".);
- ◆ *дані*: немає;
- ◆ *матеріали*: мелена кава, змінний фільтр і вода;
- ◆ *енергія*: електрика;
- ◆ *сили*: механічна опора.

Виходи ідентифікувати також просто:

- ◆ *сигнали*: стан (ми просто назвемо їх "вкл" і "викл");
- ◆ *дані*: немає;
- ◆ *матеріали*: зварена кава, використаний фільтр, кавова гуща;
- ◆ *енергія*: тепло;
- ◆ *сили*: немає.

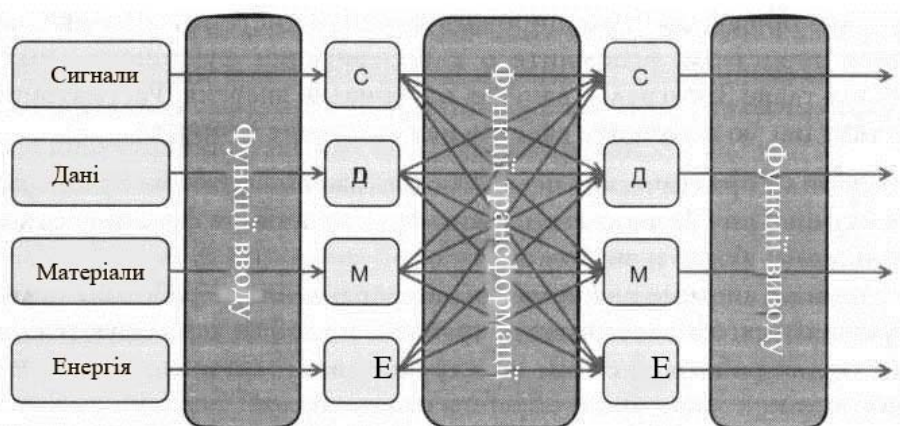


Рис. 7.3. Категорії функцій і функціонального середовища

Знання входів і виходів допомагає аналітикові ідентифікувати функції. Функції введення виявляються безпосередньо зі списку входів (дедуктивне міркування), функції виведення - зі списку виходів (ще одно дедуктивне міркування). Функції трансформації виявити складніше, тому що для цього

вимагається застосувати міркування по індукції. Проте тепер у нас є підказка: ми знаємо, що треба перетворити шість входів в п'ять виходів.

Такий підхід до дослідження зазвичай виявляє усі практично значимі функції і дозволяє згрупувати їх відносно конкретних практичних цілей. Потім це угруповання природно допомагає зібрати разом елементи різних підсистем, які є складовими частинами, розташованими на першому рівні самої системи. Описана стратегія застосовна навіть у тому випадку, коли базова конфігурація запозичена у передуючої системи, тому що що лежить в її основі узагальнений, методичний підхід сприяє виявленню елементів, які інакше можна було б прогледіти. У разі кавоварки ми можемо зосередитися на трансформації вхідних матеріалів і сигналів у вихідні матеріали і сигнали. Іншими словами, ми можемо ідентифікувати функції, відповівши на питання: "Як перетворити мелену каву, фільтр, воду і команду вкл/выкл в зварену каву, використаний фільтр, кавову гущу і стан?"

Прагнучи скласти мінімальний перелік функцій верхнього рівня і використовуючи синтаксис "дієслово - об'єкт", ми могли б прийти до такого переліку функцій кавоварки :

#### *Функції введення*

1. Прийняти команду користувача (вкл./викл.).
2. Отримати матеріал - мелену каву.
3. Подати електрику.
4. Подати воду.

#### *Функції трансформації*

5. Нагрівати воду.
6. Змішати гарячу воду з меленою кавою.
7. Відфільтрувати кавову гущу.
8. Зберегти гарячим зварену каву.

#### *Функції виведення*

9. Показати стан.
10. Забезпечити видалення матеріалів.
11. Відвести тепло.

Чи можете ви зв'язати входи і виходи з однією або декількома функціями? Чи можете ви визначити, як входи перетворюються у виходи? Оскільки кавоварка - дуже проста система, та кількість функцій трансформації невелика. Але майте на увазі, що яка б не була складність системи, від 5 до 12 функцій верхнього рівня завжди можна налічити. Можливо, для складної системи є розгалужена ієрархія функцій, але для будь-якої системи можна зробити агрегацію і отримати відповідний набір функцій верхнього рівня.

#### **Визначення показників функціонування.**

Вище відзначалося, що мета етапу дослідження концепції полягає в тому, щоб вивести необхідний і достатній набір показників функціонування системи. Це означає, що що має такі показники система повинна задовольняти наступним критеріям:

1. Система, яка відповідає вимогам призначення і є такою, що технічно реалізується і економічно виправданою, повинна відповідати заявленим показникам функціонування.

2. Система, яка має такі характеристики, повинна задовольняти вимогам призначення і може бути спроектована так, що виявиться такою, що технічно реалізовується і економічно виправданою.

Умова необхідності і достатності набору показників функціонування важлива, для того, щоб випадково не виключити з розгляду концепцію системи, що обіцяє особливий вигаш, в порівнянні з іншими, оскільки в ній застосовується незвичайний підхід до реалізації якоїсь функції системи. Так нерідко буває, коли показники функціонування частково визначаються з урахуванням характеристик попередньої системи і несуть з собою особливості, несуттєві для її роботи. Це трапляється і тоді, коли є упереджена думка про те, як певну можливість практичного використання сі-стеми слід перетворити в її функцію.

З указаних причин визначення показників функціонування має бути дослідницьким і ітеративним процесом, як показано на мал. 7.2. Зокрема, якщо є альтернативні функціональні підходи до реалізації деякої можливості практичного використання, то усі вони мають бути відбиті в показниках функціонування, принаймні до тих пір, поки деякі не будуть виключені в процесі реалізації і валидації.

Після того, як встановлені практично значимі показники функціонування для декількох придатних до реалізації концепцій, кожна з яких здатна задовольнити вимогам призначення, що пред'являються до системи, можна переходити до наступного кроку - уточнення і об'єднання їх в єдиний набір, який стане основою для підготовки офіційних вимог до показників функціонування системи. Як було відмічено вище, ці вимоги, виражені в значеннях показників, придатних для технічного виміру, утворюють однозначно певну базу для подальших етапів розробки системи, аж до стадії, на якій готова система може бути випробувана в реальних умовах.

Дії, що виконуються в ході уточнення і валидації вимог до показників функціонування системи, можна уявляти собі як два тісно пов'язаних між собою процесу - агрегація, коли порівнюються і комбінуються показники функціонування, характерні для альтернативних і придатних до реалізації концепцій, і аналіз ефективності, коли оцінюється придатність агрегованих показників в термінах вимог призначення.

### **Агрегація показників функціонування**

Процес агрегації дозволяє відібрати і уточнити ті характеристики, властиві різним концепціям системи, розглянутим в процесі дослідження, які потрібні і достатні для визначення системи, що має необхідні експлуатаційні характеристики. Незалежно від наявних аналітичних інструментів цей процес вимагає найвищого рівня кваліфікації в області системної інженерії.

Цей і інші процеси на цьому етапі можуть виграти від участі системних інженерів з досвідом роботи над попередньою системою, про що вже неодноразово згадувалося вище. Бази знань і даних про стару систему - безцінне джерело інформації для розробки нових вимог і концепцій. У багатьох випадках деякі ключові інженери і керівники, що відповідали за її розробку, все ще доступні і можуть внести свій вклад в розробку нових вимог і концепцій. Вони не лише можуть знати про існуючі недоліки, але, цілком

імовірно, думали про різні удосконалення. До того ж, накопивши за багато років знання про чинники, що впливають на експлуатацію, вони, напевно, розуміють, що дійсно треба замовникові. Всього один системний інженер з таким досвідом може надати величезну допомогу. До усього іншого, досвідчені люди такого типу "нутром чують" життєздатність даних вимог і концепцій. Їх сприяння, хоч би в ролі консультантів, не усуває потреби в аналізі вимог, але допоможе направити зусилля в правильному напрямі і уникнути безвиході.

**Валідація показників функціонування.**

Останні кроки процесу полягають в тому, щоб піддати валідації виведені показники функціонування, зіставивши їх з вимогами призначення і обмеженнями, і представити їх у вигляді документу, що містить опис вимог. У ідеалі показники функціонування, виведені на попередньому кроці уточнення, були отримані з концепцій, вже схвалених в процесі дослідження концепцій реалізації. Проте цілком могло статися, що при видаленні несуттєвих або надмірних показників на кроці агрегації і додаванні зовнішніх обмежень, які були відсутні в моделі ефективності, результуючий набір показників був істотно змінений. Тому необхідно ще раз піддати їх аналізу ефективності, щоб перевірити відповідність вимогам призначення. Взагалі кажучи, модель ефективності, використовувана на цьому кроці, має бути строгішою і детальнішою, чим колишні моделі, щоб гарантувати відсутність в кінцевому виробі дефектів, викликаних пропуском важливих критеріїв оцінки.

Описаний вище процес повторюється, поки не буде отриманий несуперечливий набір *показників функціонування системи*, що мають наступні властивості :

1. Вони визначають, що система повинна робити і наскільки добре, але нічого не говорять про те, як це повинно робитися.

2. Вони визначають характеристики в технічних термінах, так щоб їх можна було верифицировать аналітично або експериментально; ці характеристики повинні стати основою подальших проектно-конструкторських етапів розробки системи.

3. Вони повно і точно відбивають вимоги призначення, що пред'являються до системі, і обмеження, у тому числі зовнішні інтерфейси і взаємодії, так що система, що має вказані характеристики, свідомо задовольняє вимогам призначення.

**Документування вимог.**

Для перетворення показників функціонування системи в документ з вимогами потрібні уміла організація і редагування. Оскільки вимоги до показників функціонування ляжуть в основу етапу визначення концепції і що йдуть за ним, украй важливо, щоб цей документ був ясним, несуперечливим і повним. Проте не менш важливо розуміти, що він не висічений в камені, а продовжує змінюватися і покращуватися у міру розробки і випробування системи.

При розробці системи, обумовленої потребами, коли передбачається виконувати етап визначення концепції силами учасників тендеру, вимоги до показників функціонування системи стають найважливішою складовою частиною запрошення до участі в тендері разом з повним описом усіх інших

умов і обмежень. Таке запрошення часто поширюється у вигляді проекту серед потенційних учасників, щоб допомогти їм забезпечити повноту і ясність розуміння умов конкурсу.

Якщо розробка обумовлена технічним прогресом і етапи дослідження і визначення концепції виконуються однією і тією ж компанією, то кінцевий продукт зазвичай лягає в основу рішення про те, чи слід санкціонувати і фінансувати етап визначення концепції до початку конструкторської розробки. Для цього в документ з вимогами зазвичай включається детальний опис найбільш перспективних з досліджених альтернативних концепцій, аргументи на користь їх здійсненості, результати вивчення ринку, що підтверджують потребу в новій системі, і оцінка витрат на розробку, виробництво і просування на ринок.

## **КОРОТКІ ВИСНОВКИ**

### **Розробка вимог до системи**

Мета етапу дослідження концепції (у сенсі, визначеному в цій книзі) з-стоїть в тому, щоб розглянути альтернативні концепції, вивести загальні показники і перейти від функціонального погляду на систему до інженерного. Результатом дослідження концепції є: 1) вимоги до показників функціонування, 2) архітектура системи, що пропрацювала до рівня підсистем, і 3) альтернативних концепції системи.

В ході дослідження концепції виконуються наступні дії:

- ◆ *аналіз вимог призначення* - покликаний забезпечити повноту і несуперечність;
- ◆ *дослідження концепцій реалізації* - уточнення функціональних характеристик;
- ◆ *визначення вимог до показників функціонування* - встановлення функцій і параметрів;
- ◆ *валідація вимог до показників функціонування* - доказ придатності до практичного застосування.

Аналіз вимог призначення

Розробка вимог включає чотири основні кроки: встановлення, аналіз, валідацію і документування. Якщо ці кроки виконані правильно, то на виході виходить набір вимог, що добре пропрацювали.

Для породження вимог функціонального рівня зазвичай потрібний аналіз альтернативних концепцій, що включає, як правило, використання математичних і імітаційних моделей ефективності. Для проведення такого аналізу потрібні три речі: початковий набір вимог призначення, концепція функціонування даної системи і контекст функціонування - набір сценаріїв експлуатації, що дають уявлення про оточення.

### **Визначення вимог до показників функціонування**

Розробка системи - це недетермінований процес в тому сенсі, що включає процес міркувань, що повторюється, по індукції; при цьому є безліч різних рішень, здатних задовольнити вимогам призначення. Велику цінність може



представляти попередня система, оскільки на її основі можна визначити функціональну архітектуру нової системи і показники функціонування її окремих складових частин.

### **Дослідження концепцій реалізації**

Дослідження альтернативних концепцій реалізації проводиться для того, щоб:

- ◆ уникнути "синдрому зосередженого проектування";
- ◆ розглянути широкий спектр альтернатив;
- ◆ розглянути питання про адаптацію технології попередньої системи;
- ◆ розглянути інноваційні підходи із застосуванням передових технологій;
- ◆ оцінити показники функціонування, ризики, витрати і потенціал зростання кожної альтернативи.

Розробка технології також є важливим компонентом розробки системи. Промисловість і держава підтримують великі програми НІОКР, що ведуть до появи нових технологій. Технологічний фундамент, який зазвичай називають "технічною базою", є джерелом багатьох інноваційних концепцій.

Вимоги до показників функціонування розробляються в ході аналізу, мета якого - встановити показники функціонування для кожної концепції. Потім ці вимоги оцінюються на предмет відповідності вимогам призначення і обмеженням. До числа джерел обмежень можна віднести: 1) оператора системи, особливості технічного обслуговування, ремонту і випробувань; 2) вимоги до сполучення з іншими системами; 3) зовнішніх умов, в якій експлуатується система; 4) умов, пов'язаної з виготовленням, транспортуванням і зберіганням. Зрештою вимоги до показників функціонування системи визначають, що система повинна робити, але не як це повинно робитися. Ці вимоги представляють характеристики системи в технічних термінах, їх набір є необхідним і достатнім для відображення вимог призначення і наявних обмежень.

### **Валідація вимог до показників функціонування**

На етапі валідації вимог до показників функціонування виконуються два взаємозв'язані види діяльності : 1) агрегація вимог, встановлених в результаті розгляду альтернативних концепцій системи, і 2) аналіз ефективності для демонстрації задоволення вимог призначення. Вимоги до показників функціонування оформляються у вигляді динамічного документу; вони переглядаються і оновлюються на всьому протязі життєвого циклу системи.

## Лекція 8. ВИЗНАЧЕННЯ КОНЦЕПЦІЇ СИСТЕМИ

Етап визначення концепції в життєвому циклі системи знаменує початок серйозної, зосередженої роботи за визначенням функціональних і фізичних характеристик нової системи (чи значної модернізації існуючої), яка пропонується для задоволення практичних потреб, виявлених на попередніх етапах, що зачіпають розробку концепції. Завдання цього етапу - охарактеризувати систему досить детально, щоб можна було побудувати кількісний прогноз відносно показників функціонування, часу розробки і витрат упродовж усього життєвого циклу. Трудовитрати на етапі визначення концепції помітно вище, ніж на попередніх етапах, оскільки якщо раніше в проекті були зайняті переважно системні інженери і аналітики, то тепер до них додаються інженери-конструктори і фахівці в інших конкретних інженерних областях. У разі, коли розробка системи обумовлена потребами, на цьому етапі, як правило, беруть участь декілька конкуруючих розробників, що ґрунтуються на вимогах до показників функціонування, підготовлених на попередніх етапах самим замовником або за його дорученням. Результатом цього етапу є вибір серед декількох альтернативних концепцій системи конкретної конфігурації, яка стане основою для розробки і конструювання. Починаючи з цього етапу розробка системи полягатиме в основному в реалізації вибраної концепції (з необхідними модифікаціями) у вигляді програмно-апаратного комплексу, придатного до виробництва і експлуатації.

### **Метод системної інженерії при визначенні концепції**

У наступних розділах види діяльності на етапі визначення концепції розглядаються в термінах чотирьох кроків методу системної інженерії. Потім описуються планування роботи на наступній стадії розробки системи і визначення функціональних вимог до системи. Нижче наведено короткий огляд цих кроків у застосуванні до даного етапу; їх узагальнені назви показані в дужках.

*Аналіз вимог до показників функціонування (аналіз вимог).* Типові дії включають:

- аналіз вимог до показників функціонування системи і зіставлення їх з практичними цілями і зі сценарієм повного життєвого циклу;
- уточнення вимог, якщо це необхідно для врахування обмежень, які не враховувалися раніше, а також кількісне вираження якісних вимог там, де це можливо.

*Аналіз функціонування та визначення функціональних вимог {функціональний опис}.* Типові дії включають:

- прив'язку функцій підсистем до компонентів в термінах функціональних елементів системи та визначення взаємодій між ними;
- розробку артефактів функціональної архітектури;
- формування попередніх функціональних вимог, відповідаючим призначеним функціям.

*Визначення концепції (опис фізичної реалізації).* Типові дії включають:

- синтез альтернативних технологічних підходів і конфігурацій компонентів, спроектованих у відповідності з вимогами до показнику функціонування;
- розробку (включаючи виявлення серед наявних) артефактів фізичної архітектури;
- дослідження компромісів між функціональними можливостями, ризиками, витратами і термінами з метою вибору кращої концепції системи, визначеної в термінах компонентів і архітектур.

*Валідація концепції (валідація проектних рішень).* Типові дії включають:

- аналіз та імітаційне моделювання системи з метою підтвердження того, що вибрана концепція задовольняє вимогам і перевершує інші варіанти;
- уточнення концепції при необхідності.

**Категорії вимог до системи.** При обговоренні аналізу вимог акцент зазвичай робиться на те, які функції система повинна виконувати і наскільки добре. Відповідно, говорять про функціональні вимоги і вимоги до показників функціонування. У загальному випадку ці вимоги визначені чітко. Однак бувають вимоги інших видів, не менш важливі, але визначені набагато гірше або навіть взагалі не розглядалися до цього моменту. До них відносяться:

- *вимоги до сумісності:* як система повинна сполучатися зі своїм місцем експлуатації, зі своїм логістичним забезпеченням і з іншими системами;
- *вимоги до надійності, готовності та ремонтпридатності:* наскільки надійною повинна бути система, щоб її можна було використовувати за призначенням, як буде здійснюватися її технічне обслуговування та ремонт, які будуть потрібні засоби забезпечення та супроводу;
- *вимоги до навколишнього середовища:* які граничні параметри навколишнього середовища система повинна витримувати протягом свого терміну служби (експлуатації).

Вимоги до надійності, готовності та ремонтпридатності, якщо вони взагалі сформовані явно, часто виявляються довільними і не надто чітко визначеними. Що стосується двох інших категорій, вони в основному обмежуються робочими режимами функціонування систем; без уваги залишаються погрузка, зберігання, транспортування, складання та забезпечення. У таких випадках необхідно детально вивчити весь життєвий цикл системи - від поставки виробу до завершення терміну служби та ліквідації.

**Сценарій життєвого циклу системи.** Щоб усвідомити всі ситуації, в яких система може виявитися протягом терміну служби, потрібно розробити модель або сценарій, де перераховані всі можливі умови, в яких може опинитися система. При цьому повинні бути враховані як мінімум:

- зберігання системи та / або її компонентів;
- транспортування системи до місця експлуатації;
- збірка і підготовка системи до роботи;
- тривале перебування у польових умовах;

- експлуатація системи;
- планове та аварійне технічне обслуговування;
- модифікація та модернізація системи;
- ліквідація системи.

Модель цих етапів використання системи повинна бути достатньо детальною і відображати всі ті взаємодії системи з оточенням, які мають значення для її проектування. Наприклад, для технічного обслуговування і ремонту системи необхідний комплект запасних частин, спеціальне контрольно-випробувальне обладнання, спеціальні контрольні точки та інші види забезпечення, які слід брати до уваги.

**Функціональні складові частини.** Спільну природу задачі перетворення вимог до показників функціонування та у вимоги до функцій системи можна проілюструвати на прикладі використання концепції функціональних складових частин системи, перелічених у лекції 3. Продовжуючи обговорення, розпочате в лекції 7, виділимо наступні кроки:

1. *Ідентифікація функціонального середовища.* Тип фізичного середовища (сигнали, дані, матеріали, енергія, сила), яка використовується для організації взаємодії при здійсненні кожної з основних функцій системи, зазвичай легко асоціювати з одним з цих п'яти класів, застосовуючи запропоновані в лекції 7 критерії.
2. *Ідентифікація функціональних елементів.* Операції над кожною з п'яти різновидів функціональних середовищ представлені п'ятьма або шістьма базовими функціональними елементами, перерахованими в главі 3, кожен з яких виконує якусь істотну функцію і зустрічається в системах різних типів. Дії системи (функції) можна сконструювати з деяких підмножин цих функціональних складових частин.
3. *Співвіднесення вимог до показників функціонування з характеристиками елементів.* Кожен функціональний елемент володіє декількома ключовими характеристиками (наприклад, швидкість, точність і ємність). Якщо ці характеристики вдається співвіднести з істотними вимогами до показників функціонування, то тим самим підтверджується правильність вибору функціонального елемента.
4. *Конфігурація функціональних елементів.* Функціональні елементи, відібрані для досягнення необхідних показників функціонування, необхідно зв'язати між собою і згрупувати в підсистеми. Для зв'язування можуть знадобитися додаткові інтерфейсні елементи (введення-виведення).
5. *Аналіз і включення всіх зовнішніх взаємодій.* В заданому наборі вимог до показників функціонування часто не розглядаються важливі взаємодії системи зі своїм експлуатаційним (або іншим) оточенням (наприклад, із зовнішніми органами управління або джерелами енергії). Ці взаємодії необхідно включити в загальну функціональну конфігурацію.

Не рекомендується на даній стадії роботи спроби оптимізації. Початковий опис проектних рішень щодо функціонування системи ще належить модифікувати після чергового кроку - розробки фізичної архітектури - і на подальших ітераціях.

### Інструменти для графічного представлення функціональних блоків

Існує (і розробляються нові) кілька формальних методів та інструментів для представлення функціональних блоків системи і взаємодії між ними. У промисловості використовується діаграма, відома як схема функціональних потоків (Functional Flow Block Diagram - FFBD), на якій показуються не тільки функціональні можливості, але і потік управління (або будь-які з п'яти базових елементів). Цю техніку наочного представлення можна використовувати на декількох рівнях для освіти ієрархії функціональних блоків.

Порівняно недавно розроблений комплексний метод функціонального моделювання (Integration DEFinition - IDEF). Насправді цей метод виходить за рамки моделювання лише функціонування системи, але охоплює цілий спектр описів можливостей системи. Метод функціонального моделювання IDEF0 - це основна техніка, застосовувана для опису функціонування системи. Базовою конструкцією є функціональний блок, зображуваний у вигляді прямокутника, намальованого суцільними лініями (рис.8.1). Існують суворі правила позначення інтерфейсів, за допомогою яких блок взаємодіє з іншими блоками або із зовнішнім по відношенню до моделюючої системи середовищем. Іноді усередині прямокутника поміщаються деталі, наприклад список функцій, виконуваних даним об'єктом, а іноді прямокутник залишають порожнім. Входи завжди малюються зліва від прямокутника, виходи - справа. Керуючі сигнали (крім входів) малюються зверху від функціонального блоку, а вхід механізмів (або реалізації) – знизу<sup>1</sup>. (1 Опис комплексу засобів для наочного подання широкого спектру ділових, виробничих та інших процесів і операцій на будь-якому рівні деталізації, а також організаційні та методичних прийомів застосування цих засобів можна знайти в Рекомендаціях з стандартизації Р 50.1.028-2001 « Інформаційні технології підтримки життєвого циклу продукції . Методологія функціонального моделювання »).



Рис.8.1. Структура функціональної моделі IDEF0.

Один з найпростіших методів побудови графічних схем - це функціональна блок-схема (Functional Block Diagram - FBD). Вона схожа, з однієї сторони, на FFBD, але без структури потоків, а з іншого - на IDEF0, але без правил зображення схем. По суті, кожна функція відображається прямокутником. Інтерфейси між функціями позначаються спрямованими стрілками з мітками, в яких описується, що передається від однієї функції іншій. Якщо є інтерфейс між функцією і зовнішнім об'єктом, то цей об'єкт теж якось представляється (наприклад, прямокутник і окружність) і наноситься інтерфейсна стрілка.

Згадаймо приклад кавоварки з лекції 7. Ми ідентифікували 11 функцій, список яких для зручності повторений нижче.

#### *Функції введення*

1. Прийняти команду користувача (вкл / викл).
2. Отримати матеріали для приготування кави.
3. Подати електрику.
4. Подати воду.

#### *Функції трансформації*

5. Нагріти воду.
6. Змішати гарячу воду з меленою кавою.
7. Відфільтрувати кавову гущу.
8. Зберегти гарячою заварену каву.

#### *Функції виводу*

9. Показати стан.
10. Забезпечити видалення матеріалів.
11. Відвести тепло.

На рис. 8.2 представлена FBD для всіх 11 функцій. Показані також зовнішні об'єкти: користувач, джерело енергії (передбачається електрична розетка) і оточення. Відзначимо, що ні в списку функцій, ні на схемі технічне обслуговування не розглядається. Пояснюється це природою побутових приладів взагалі і кавоварок зокрема. У їх конструкції ремонт не передбачений, вони «одноразові».

Оскільки важко уникнути перетинання ліній, існує кілька способів відрізнити різні інтерфейсні стрілки. Найпоширенішим, мабуть, є колір, але застосовуються й інші, наприклад пунктирні лінії. У разі енергії ми просто перерахували потребуючі її функції (для прикладу, «F5»). Ми намагалися малювати досить ретельно, щоб читачеві було легше осмислити процес ідентифікації функцій і розробки функціональної структури системи. Було б неважко спростити цю схему, оскільки на даній стадії кілька функцій можна опустити, за умови, що в результаті ми про них не забудемо. Наприклад, функцію 10 «забезпечити видалення матеріалів» цілком можна поки що прибрати, аби в остаточному проекті можливість видалення відходів була присутня. Відзначимо також, що функції можна розбити на категорії за ознакою роботи з п'ятьма базовими елементами:

Матеріали	Отримати матеріали для приготування кави; Змішати гарячу воду з меленою кавою; Відфільтрувати кавову гущу; Забезпечити видалення матеріалів.
Дані	Показати стан.
Сигнали	Прийняти команду користувача.
Енергія	Подати електроенергію; Нагріти воду; Зберегти гарячою зварену каву; Відвести тепло.
Сила	Розподілити вагу.

Це не «чисте» розбиття, тому що деякі функції приймають на вході елемент одного типу і перетворюють його в елемент іншого типу. Наприклад, функція 2 «прийняти команду користувача» отримує на вході дані і перетворює їх в сигнали. Доводиться приймати суб'єктивне рішення.

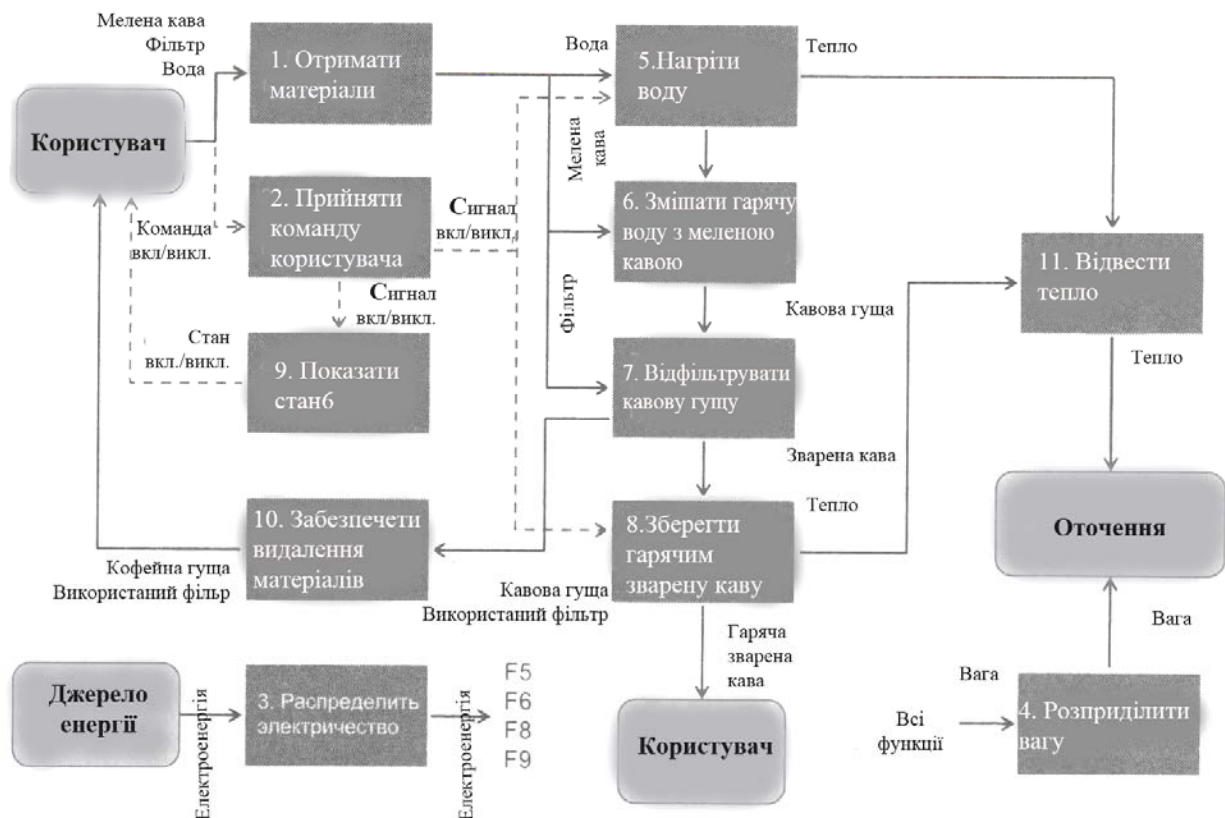


Рис.8.2. Функціональна блок-схема стандартної кавоварки.

### Уніфікована мова моделювання UML

Ми вже говорили, що при розробці складної системи дуже важливо створювати високорівневі моделі її структури та поведінки, щоб зрозуміти, як може бути налаштована система для того, щоб вона відповідала поставленим вимогам. При розробці методології OOAD кілька відомих вчених-практиків

незалежно запропонували подібні моделі. У середині 1990-х років троє з них (Градї Буч, Джеймс Рамбо і Івар Якобсон) запропонували єдину термінологію для моделювання, яку вони назвали UML. Спільнота розробників ПЗ прийняло цю мову в якості стандарту, після чого він став широко використовуватися в комерційних і державних проектах. Мова UML підтримана розвиненими інструментами, що розробляються декількома провідними виробниками інструментального ПЗ.

Якщо у структурній методології застосовуються три взаємодоповнюючих представлення системи, то UML пропонує аналітикам і фахівцям з проектування 13 різних способів схематичного представлення різних характеристик системи. Серед них виділяються шість статичних, або структурних, діаграм і сім динамічних, або поведінкових, діаграм. На рис.8.3 показані обидва набори цих діаграм.

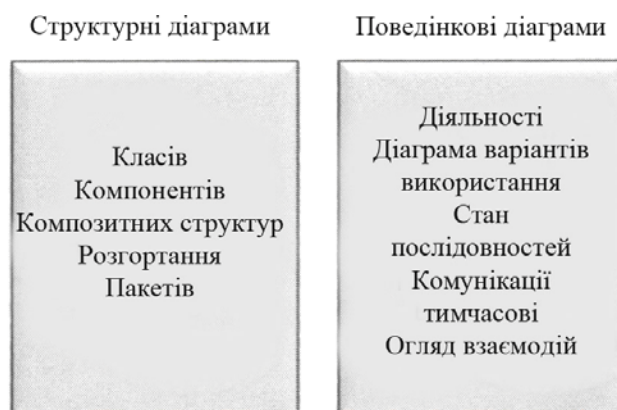


Рис.8.3. Моделі UML.

*Структурні діаграми* служать для відображення різних представлень про зв'язки між сутностями, що утворюють систему.

- На *діаграмах класів (class diagram)* показується набір класів, зв'язків між ними та їх інтерфейсів.
- На *діаграмах об'єктів (object diagram)* показуються екземпляри класів (об'єкти) системи і зв'язки між ними.
- *Діаграми компонентів (component diagram)* зазвичай використовуються для пояснень структури фізичних об'єктів і зв'язків між ними.
- *Діаграма розгортання (deployment diagram)* дає статичне представлення фізичних компонентів системи.
- На *діаграмі композитної/складової структури (composite structure diagram)* демонструється декомпозиція класів під час виконання завдання.
- На *діаграмах пакетів (package diagram)* представляється ієрархія компонентів.

*Поведінкові діаграми* служать для відображення різних представлень про динамічні характеристики системи.

- На *діаграмах варіантів використання (use case diagram)* показані відношення між варіантами використання, що представляють функції



системи, які відповідають за взаємодії з зовнішніми сутностями (акторами).

- На *діаграмах послідовностей (sequence diagrams)* показані хронологічні впорядковані взаємодії між об'єктами при реалізації сценарію системи.
- На *діаграмах станів (state machine diagrams)* моделюються події переходу станів та дії, які змінюють стан системи.
- *Діаграми діяльності (activity diagrams)* являють собою блок-схеми різних дій в деякій частині системи і потоки управління між ними.
- На *діаграмах комунікацій (communication diagrams)* визначаються зв'язки між об'єктами з акцентом на їх взаємодії.
- *Діаграми огляду взаємодій (interaction overview diagrams)* - це комбінація діаграм послідовності і діяльності.
- На *тимчасових діаграмах (timing diagrams)* представлені взаємодії між об'єктами разом з хронометричною інформацією.

Діаграми класів в UML приблизно відповідають діаграмам сутність-зв'язок в структурному аналізі, а діаграми станів - діаграмам переходу станів. Інші діаграми, особливо діаграми діяльності, - це різні види схем функціональних потоків. Нова мова була швидко прийнята на озброєння спільнотою системних інженерів як стандарт де-факто для представлення програмних концепцій і переважно програмних систем. Хоча витoki мови лежать в світі програмного забезпечення, останнім часом він успішно застосовувався для розробки систем, що включають як програмні, так і апаратні засоби.

Розвиток мови UML направляється всесвітнім консорціумом Object Management Group (OMG). UML продовжує розвиватися, виходять нові версії, збільшується складність. Замість того щоб детально розповідати про всі діаграми, ми наведемо декілька прикладів поведінкових діаграм - варіантів використання, діяльності та послідовності - і одну структурну - діаграму класів.

**Діаграма варіантів використання.** Ми вирішили почати з цієї діаграми, тому що вона знаходить застосування при описі функціонування та поведінки системи. У програмних, та й у деяких апаратних додатках діаграма варіантів використання застосовувалася для виявлення й аналізу вимог призначення і функціональних вимог.

Зовні діаграма варіантів використання виглядає, як показано на рис.8.3, де моделюється взаємодія діючої особи або суб'єкта дії - «актора» - ліворуч (представлений чоловічком) з одним варіантом використання (представлений овалом), яке веде до виконання підлеглого дії (окремий варіант використання), тоді як три інших варіанти взаємодіють із другим (зовнішнім) актором. Стрілки показують, хто ініціює варіант використання, а не напрямок потоку інформації. Наприклад, актор «бібліотекар» може ініціювати варіант використання «управління абонементом». Той же варіант може бути ініційований варіантом використання «повернути книгу».

Кожен зображений на діаграмі варіант використання представляє окрему послідовність дій і подій. У UML визначений стандартний набір компонентів для варіанту використання, а саме:

- назва;
- короткий опис;
- список акторів;
- початкові умови (передумови), що описують стан оточення до початку варіанту використання;
- кінцеві умови (постумови), що описують стан оточення після завершення варіанту використання;
- послідовність подій - список подій або дій, що відбуваються в певній послідовності.



Рис.8.3. Діаграма варіантів використання.

**Діаграма діяльності.** Як інший приклад поведінкової діаграми візьмемо діаграму діяльності. На таких діаграмах можна представити будь-який тип потоку в системі, включаючи процеси, операції та управління. На діаграмі це досягається за допомогою послідовності дій та подій, яка регулюється за допомогою різних керуючих вузлів. Нижче описані основні компоненти діаграми діяльності:

- дію (action): елементарний здійснений крок у рамках діяльності (прямокутник з округленими кутами);
- дуга діяльності (activity edge): лінія, що з'єднує дії, а також дії та вузли (зі стрілкою). Дуги діяльності бувають двох типів: потоки об'єктів і потоки управління;
- потік об'єктів (object flow): дуга діяльності, по якій переміщаються об'єкти (або символічні позначення об'єктів);
- потік управління (control flow): дуга діяльності, яка представляє напрям управління (крім того, по ній переміщуються маркери управління);
- з'єднувач, або скріпка (pin): сполучна ланка між параметрами дії і потоком (прямокутник, з'єднаний з дією і потоком). З'єднувач приймає явні входи або породжує явні виходи дії;

- початковий вузол (initial node): початкова точка потоку управління (суцільний круг);
- кінцевий вузол (final node): кінцева точка потоку управління (суцільний круг всередині кола більшого радіусу);
- вузол прийняття рішення (decision node): точка розгалуження потоку, кожна гілка містить умову, за якої вона вибирається (ромб);
- вузол злиття (merge node): точка, в якій кілька потоків об'єднуються в один (ромб);
- вузол поділу (fork node): точка, в якій один потік розгалужується на декілька паралельно виконуваних (жирна лінія);
- вузол з'єднання (join node): точка, в якій кілька паралельно виконуваних потоків синхронізуються і з'єднуються в один потік (жирна лінія).

На рис. 8.4 показана проста діаграма діяльності для нашої бібліотечної системи, аналогічна схемі функціональних потоків. Тут одна діяльність розгалужується на дві, що відповідають двом логічним шляхам: видача і повернення книги.

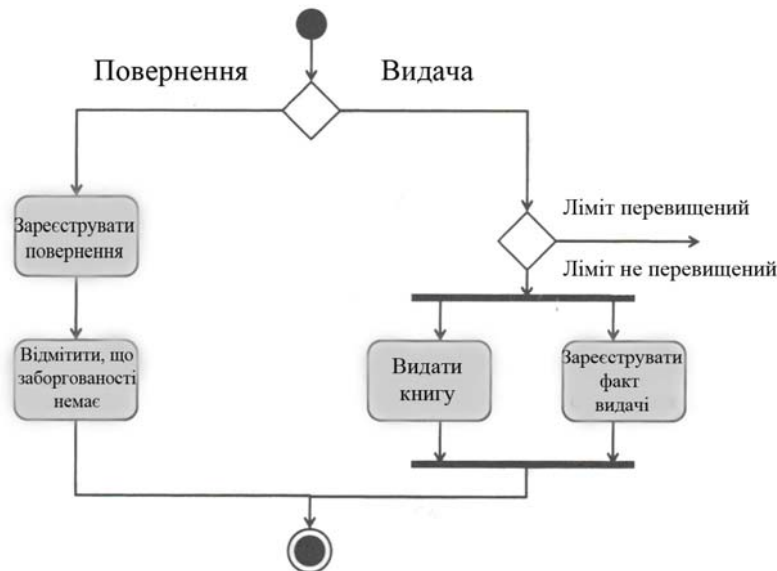


Рис.8.4. UML-діаграма діяльності.

**Діаграма послідовності.** В якості останньої з поведінкових діаграм ми розглянемо діаграму послідовності. Такі діаграми звичайно зв'язуються з варіантами використання, в яких дії або події слідують в певному порядку. На діаграмі послідовності цей порядок зображується у вигляді послідовності подій, кожна з яких асоційована з виконуючим його актором або підсистемою.

На рис. 8.5 наведено приклад діаграми послідовності для операції видачі книги. Ця діаграма пов'язана з представленим вище варіантом використання, але містить ще й додаткову інформацію.

**Діаграма класів.** В основі UML лежить поняття класу; для його представлення призначені діаграми класів. Можна вважати, що клас - це просто безліч об'єктів (реальних чи віртуальних), що володіють однаковими характеристиками і семантикою. Об'єкт може бути практично чим завгодно,

уявленим програмними засобами (якщо мова йде про UML). Типовий клас описує структуру і поведінку об'єктів класу.

Визначення класу містить три основні компоненти (у числі інших):

- *атрибути*: структурні властивості класу;
- *операції*: поведінкові властивості класу;
- *обов'язки*: зобов'язання класу.

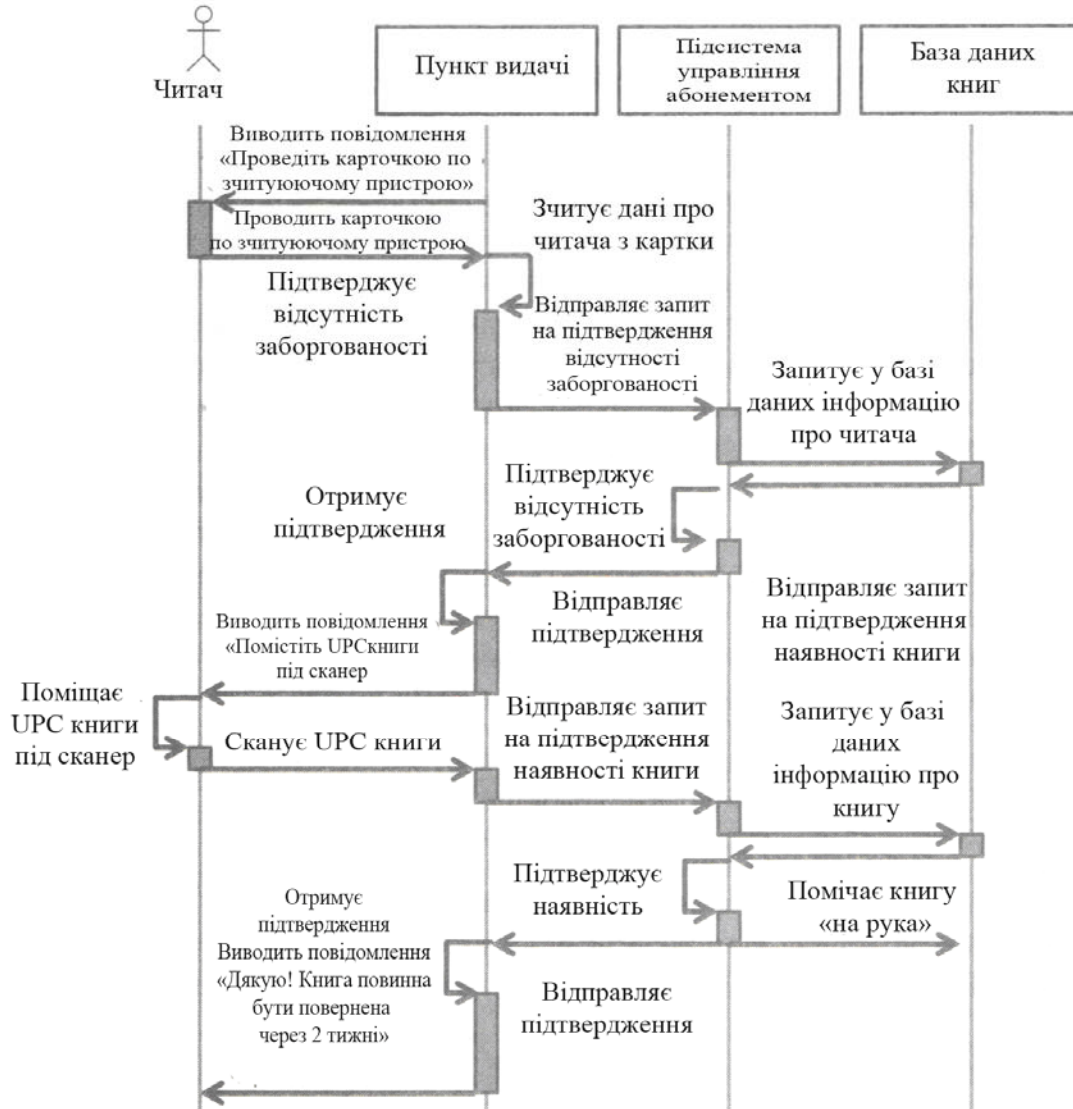


Рис.8.5. UML-діаграма послідовності.

Зазвичай між класами є зв'язки. Найпростіший структурний зв'язок називається *асоціацією*. На рис. 8.6 показана проста асоціація між двома класами, «Працівник» і «Компанія». Лінія, що з'єднує два класи, може бути забезпечена стрілкою; якщо ж стрілка відсутня, мається на увазі двосторонній зв'язок. Характер асоціації можна також описати за допомогою трикутника. У цьому випадку асоціація читається наступним чином: «Працівник працює в компанії» і «Компанія наймає працівника». Нарешті, автор може вказати *кратність* асоціації, яка визначає її чисельні аспекти і записується окремими числами або в тій чи іншій скороченій нотації. Наприклад, запис 0..2 означає будь-яке значення від 0 до 2, а символ зірочки (\*) інтерпретується як «багато». Таким чином, у нашому прикладі зірочка і число «1» означають, що працівник

працює тільки в одній компанії, а компанія може найняти багато робітників. Між класами можуть існувати зв'язки ще двох типів: *узагальнення* і *залежність*. Узагальнення описує таксономічний зв'язок між приватним і загальним класами.

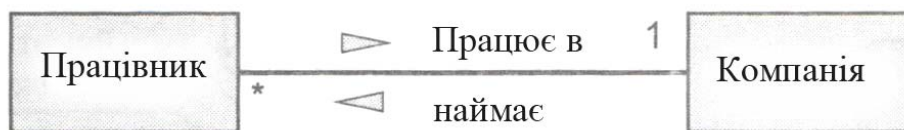


Рис.8.6. Приклад асоціації класів.

Приватні класи, пов'язані ставленням узагальнення з більш загальним класом, успадковують атрибути та операції свого батька. Отже, клас корпоративний клієнт володіє не тільки своїми власними атрибутами і операцією, але також атрибутами Name і Address і операцією getCreditRating (). Те ж саме справедливо і для класу клієнт - фізичне обличчя.

Третій тип зв'язку - залежність - застосовується в ситуації, коли в описі або реалізації одного класу використовується інший. Слід зазначити, що відношення залежності може існувати не тільки між класами, але і між іншими елементами UML.

### Специфікація функціональних вимог до системи

Етап розробки концепції не можна вважати завершеним, якщо не створена формальна основа для керівництва подальшою стадією розробки інженерно-технічних рішень. Її наріжним каменем є документ, в якому повно і лаконічно описані всі функції, які повинна виконувати система, що відповідає вимогам призначення. У великих програмах державних закупівель такий документ зазвичай називається «специфікацією системи», або «А-специфікацією»<sup>1</sup> (1 А-специфікація має багато спільного з технічним завданням, яке повсюдно використовується у вітчизняній практиці інженерних розробок. - *Прик. ред.*).

Специфікацію системи можна розглядати як документ, в якому представлена концепція системи у вигляді тексту і діаграм. У ньому нічого не говориться про те, як система повинна виконувати свої функції, але вказується, які функції повинні виконуватися, з якою точністю і за яких умов. Важливо, щоб всі визначення були сформульовані у вигляді, придатному для перевірки на основі вимірів, оскільки на них буде спиратися інженерно-технічна реалізація функцій.

Хоча підготовка специфікацій системи логічно є частиною етапу розробки концепції, в тендерній процедурі придбання це зазвичай робиться переможцем тендеру відразу після підбиття підсумків. При розробці комерційних виробів даний процес менш формальний, але переслідує подібні цілі.

У специфікації системи повинні бути відображені як мінімум наступні питання.

#### Опис системи

Призначення системи і концепція її функціонування.

Конфігурація і організація системних інтерфейсів.

#### Обов'язкові характеристики

Показники функціонування (обладнання та програмного забезпечення) і вимоги до сумісності.

Вимоги до надійності, ремонтпридатності та придатності до використання.

*Вимоги до забезпечення*

Упаковка, вантаження, транспортування, зберігання, підготовка персоналу.

Спеціальні споруди та обладнання.

*Особливі вимоги*

Охорона праці та безпека.

Відділ системної інженерії повинен взяти на себе керівництво підготовкою специфікацій системи і значну частину фактичної роботи.

## **КОРОТКІ ВИСНОВКИ**

### **Визначення концепції системи**

Завдання етапу визначення концепції - виділити надаючій перевагу конфігурацію системи, сформулювати функціональні специфікації системи і визначитися з календарним планом (розкладом) робіт і витратами.

Визначення концепції завершує стадію розробки концепції, на якій закладається фундамент для наступної стадії життєвого циклу – розробки інженерно-технічних рішень. Обрана краща концепція являється також еталоном, на який слід орієнтуватися при розробці та конструюванні.

На цьому етапі виконуються наступні дії:

- *аналіз вимог до показників функціонування* - співвіднесення з практичними цілями;
- *аналіз функціонування та визначення функціональних вимог* - привязка функцій до компонентів;
- *вибір концепції* - вибір кращої концепції шляхом аналізу компромісів;
- *валідація концепції* - підтвердження правильності та переваги вибраної концепції.

### **Аналіз вимог до показників функціонування**

У ході аналізу вимог до показників функціонування повинно бути показано, що система сумісна з місцем передбачуваної експлуатації та доступною там логістичною підтримкою. Повинні бути також розглянуті питання надійності, ремонтпридатності і засобів забезпечення, а також сумісності з навколишнім середовищем. Слід звертати увагу на весь життєвий цикл, від виробництва до ліквідації системи. Нарешті, в ході аналізу повинні бути уточнені формулювання вимог, що не допускають кількісного вираження.

### **Аналіз функціонування і формування функціональних вимог**

Для функціонального опису корисні функціональні складові частини системи (див. глав 3). Вибір кращої концепції - завдання системного інженера, який повинен розглянути і порівняти кілька альтернативних концепцій.

## Прив'язка функцій

Розробка альтернативних концепцій - це одночасно наука і мистецтво. Зрозуміло, точкою відліку для побудови подальших концепцій може служити попередня система (у припущенні, що вона існує). Можуть виявитися корисними також мозковий штурм та інші новаторські методики колективного вирішення завдань.

## Вибір концепції

Концепції системи оцінюються з точки зору: 1) експлуатаційної сумісності і показників функціонування; 2) термінів і вартості програми; 3) ризиків, пов'язаних з досягненням кожної із зазначених вище цілей. Можна рахувати, що ризик програми є комбінацією двох факторів: ймовірності, що цілі системи не будуть досягнуті, і впливу такого розвитку подій на успіх програми в цілому.

Ризики програми обумовлюються рядом причин:

- використання неперевіреної технології;
- високі вимоги до показників функціонування;
- несприятливі умови навколишнього середовища;
- недостатнє фінансування або відсутність кваліфікованих кадрів;
- не виправдано жорсткі терміни.

Аналіз компромісів лежить в основі будь-якого систематичного процесу прийняття рішень.

## Валідація концепції

Аналіз компромісів, що виконується при виборі концепції, повинен бути:

- *організованим* - планується як окремий процес;
- *вичерпним* - розглядається повний спектр альтернатив;
- *напівкількісним* - критеріям призначаються відносні ваги;
- *ретельним* - розглядаються всі основні характеристики;
- *документованим* - результати докладно описуються.

В обґрунтуванні практичної реалізації обраної концепції має бути відображено наступне:

- продемонстровано, що заявлена потреба дійсно існує;
- вказані причини вибору даної концепції з ряду альтернатив;
- описані ризики програми і заходи щодо їх обмеження;
- наведені докази наявності детальних планів, наприклад WBS, SEMP і т. п. ;
- представлені свідчення наявності попереднього успішного досвіду;
- дана оцінка витрат протягом життєвого циклу;
- розглянуті інші питання, які стосуються справи, у тому числі вплив на навколишнє середовище.

## **Планування розробки системи**

Ієрархічна структура робіт (WBS) - важлива частина програми розробки системи. У ній описуються всі підлягаючі виконанню завдання.

У плані управління системної інженерією - SEMP (або його еквіваленті) визначаються всі які відносяться до системної інженерії види діяльності протягом життєвого циклу системи.

## **Побудова архітектури системи**

Побудова архітектури системи зводиться насамперед до виділення і опису різних результатів розгляду або точок зору на систему. Практично всі системні архітектури розглядають систему щонайменше в трьох ракурсах:

- *операційному* - представлення системи з точки зору користувача, або оператора;
- *логічному* - представлення системи з точки зору замовника або керівника;
- *фізичному* - представлення системи з точки зору фахівця з проектування.

Методики опису архітектури визначають структуру і моделі, застосовувані для розробки і відображення системної архітектури. Вони покликані забезпечити однакове уявленнь в різних програмах.

## **Моделе-орієнтована системна інженерія**

Основоположна ідея моделе-орієнтованої системної інженерії полягає в тому, що модель системи розробляється на ранніх стадіях процесу і еволюціонує в ході життєвого циклу розробки системи, до тих пір поки не перетвориться, по суті справи, в прототип. На ранніх стадіях життєвого циклу модель ще недостатньо точна і використовується головним чином для прийняття рішень. У міру просування розробки рівень якості моделі зростає, і в кінці кінців вона стає придатною для використання в проектуванні. І нарешті, модель ще раз трансформується - на цей раз в прототип.

## **Специфікація функціональних вимог до системи**

Специфікація функціональних вимог до системи містить її функціональний опис, обов'язкові характеристики та вимоги до забезпечення.



## Лекція 9. ПІДТРИМКА ТА АНАЛІЗ РІШЕННЯ

Для полегшення процесу прийняття рішення, було запропоновано дві стратегії: застосування методу системної інженерії та розбиття життєвого циклу системи на послідовність чітко визначених етапів.

Форма і контекст прийняття рішень можуть істотно різнитися. Більш того, всі ми приймаємо рішення практично безперервно з моменту пробудження і до відходу до сну. Простіше кажучи, рішення рішенням ворог. І єдиного процесу прийняття рішень, придатного на всі випадки життя, не існує. Безперечно, рішення про те, що з'їсти на сніданок, незрівнянне з рішенням про місце розміщення нової АЕС.

Будь-яке рішення приймається з урахуванням контексту. У цій главі ми будемо розглядати рішення, які приймають системні інженери при розробці складних систем. Таким рішенням спочатку властива складність. Приймати їх важко. Як правило, доводиться працювати в умовах невизначеності - у системного інженера немає повної інформації, необхідної для прийняття оптимального рішення. Але навіть за наявності великого обсягу інформації людина, яка приймає рішення, можливо, не зуміє обробити і узагальнити її за той час, яким володіє до моменту, коли буде потрібно готове, обгрунтоване рішення.

Для простих рішень звичайно цілком достатньо якоїсь базової інформації та інтуїції. Наприклад, щоб вирішити, чим поснідати, потрібна інформація: які продукти є в наявності, наскільки добре людина вміє готувати і скільки часу у розпорядженні. Результатом прийняття цього простого рішення є вибір продуктів для приготування сніданку. Але у випадку складних рішень помітно збільшуються як кількість входів і виходів, так і обсяг планування. Крім того, зібрану інформацію необхідно організувати, узагальнити й представити відповідальним особам у вигляді, сприятливого прийняттю «хороших» рішень.

На рис. 9.1 представлений упрощений процес прийняття складних рішень. Нижче в цьому главі ми опишемо його більш детально.

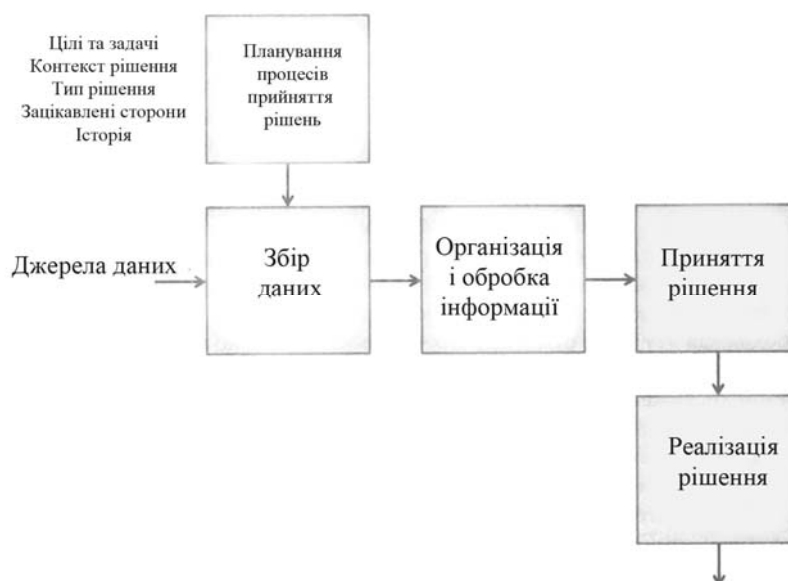


Рис. 9.1. Упрощена модель прийняття рішень.

На перший погляд, все це виглядає досить громіздко. Однак скільки часу, сил і ресурсів направляти на кожну стадію, залежить від типу, складності та сфери дії прийнятого рішення. На прийняття офіційних рішень, типових для великих програм державних закупівель, можуть піти роки, тоді як рішення про компоненти відносно простої системи можна прийняти всього за кілька годин або й того менше.

Кожна стадія займає кінцевий час. Навіть саме «прийняття рішення» - не обов'язково миттєва дія. Наприклад, якщо прийняття та затвердження рішення вимагає участі кількох осіб, то ця стадія може затягнутися. Якщо ж потрібен ще й консенсус, то вона може виявитися досить складною, що вимагає обліку не тільки технічних і організаційних, а й політичних міркувань. Зрозуміти, які ресурси залучені на кожному етапі, можна на прикладі роботи державних законодавчих органів. Планування, збір і організація інформації зазвичай виконуються помічниками, а також в ході відкритих і закритих слухань. Стадія прийняття рішення в дійсності є складним процесом, що включає політичне маневрування, укладання угод, рекламні акції, агітаційні кампанії і демонстрацію намірів. Бувало, що ця стадія займала багато місяців.

Але незалежно від типу рішення і площадки, на якій воно приймається, існує безліч факторів, які слід враховувати, перед тим як починати і завершувати стадію планування процесу прийняття рішення.

### **Фактори, що впливають на процес прийняття рішення**

Для належного прийняття складного - і при цьому корисного - рішення необхідно розуміти процес у всьому його різноманітті. На стадії планування належить враховувати такі фактори.

**Цілі і завдання.** До прийняття рішення потрібно поставити собі питання: які цілі і завдання зацікавлених сторін? На різних рівнях організації вони, швидше за все, будуть відрізнятися. Цілі лінійного майстра і керівника програми не однакові. Які цілі мають більш високий пріоритет? І які цілі керівництва, що стоїть вище особи, що приймає рішення? Прийняте рішення повинно (по мірі можливості) відповідати цілям і завданням ключових зацікавлених сторін.

**Тип рішень.** Людина, яка приймає рішення, повинна розуміти його тип. Багато невдалих рішень були прийняті через нерозуміння цієї сторони питання. Чи є рішення двійковим, наприклад пов'язаним з отриманням якогось дозволу? У таких випадках потрібно дати просту відповідь: так чи ні. В інших випадках результатом двійкового рівняння може бути не відповідь «так» чи «ні», а вибір з двох варіантів; класичний приклад - рішення про те, виготовляти самостійно або закуповувати. І нарешті, приймаючий рішення повинен розуміти, хто або що від нього залежить. Це рішення чисто технічна або зачіпає інтереси якихось людей? Неправильне визначення типу рішення неодмінно призведе до негативних наслідків.

Продовжуючи цю тему, відзначимо, що надзвичайно важливо розуміти, кого включати в процес прийняття рішення. Чи приймається рішення одноосібно? Або потрібно досягнення консенсусу в якійсь групі? Хто повинен затвердити рішення до його реалізації? Від відповідей на ці питання залежить, коли і як приймаються рішення.

**Контекст рішення.** Для прийняття правильного рішення необхідно також розуміти сферу його дії. Глобальне (або масштабу підприємства) рішення сильно відрізняється від рішення про компонент системи. Наприклад, невдале рішення, що зачіпає все підприємство, загрожує далекосяжними наслідками. Контекст включає, зокрема, розуміння того, яка проблема чи питання привели до необхідності приймати рішення. Це не так просто, тому що у контексті багато аспектів, що ставить перед особою, яка приймає рішення, різні цілі та завдання:

- технічні, що включають фізичні об'єкти, наприклад рішення про підсистеми;
- фінансові, що включають інструменти та обсяги інвестування;
- кадрові, включаючи окремих працівників;
- відносяться до процесів, включаючи ділові та технічні процедури, методи і прийоми;
- організаційні, включаючи виділення ресурсів (у тому числі часу, місця і фондів);
- тимчасові, в сенсі проміжку часу, протягом якого є необхідність в рішенні (він може бути рухомим);
- успадковані, включаючи рішення, що приймалися раніше.

**Зацікавлені сторони.** Під зацікавленою стороною розуміється будь-яка особа (фізична або юридична), інтереси якої зачіпає прийняте рішення. Усвідомити, які сторони зацікавлені в рішенні, необхідно до його прийняття. Часто це не робиться - зацікавлені сторони не ідентифіковані до моменту ухвалення рішення. Але навіть у цьому випадку, коли про рішення оголошено або розпочата його реалізація, ми можемо подбати про те, щоб думки всіх, кого воно стосується, були почуті.

**Успадковані рішення.** Знання того, які рішення за схожими питаннями приймалися в минулому, допомагає встановити як контекст (див. вище), так і оточення, в якому має бути прийнято нове рішення. Якщо особа, приймаюча рішення, має інформацію про історію, то йому простіше визначити зацікавлені сторони і наслідки.

**Додаткові дані.** Нарешті, необхідно вчасно надати додаткові дані для прийняття рішення. Для отримання впевненості в тому, що інформація, необхідна для прийняття рішення буде зібрана, потрібний ясний і своєчасно складений план збору даних. Ретельність, з якою слід збирати дані залежить від типу і контексту рішення. Нерідко траплялося, що ухвалення рішення відкладалося тільки через те, що попередредньо вимоги, пред'явлені до точності і обсягу даних, потрібних для прийняття рішення, були недостатні.

### **Статичне моделювання для прийняття рішень**

Вище вже зазначалося, що модель служить основним засобом для того, щоб впоратися зі складністю і надати допомогу в управлінні дорогим процесом розробки, побудови та випробування складних систем. У цьому зв'язку модель визначається як «фізичне, математичне або інше логічне представлення суті, явища або процесу в системі». Ми користуємося моделями, щоб отримати уявлення про системи або про якісь їхні частини і таким чином вивчити їх

поведінку за певних умов. На основі спостережень за поведінкою моделі в деякому діапазоні умов ми можемо приймати обґрунтоване рішення про розробку, виробництво і розгортання системи. Більше того, ми можемо представляти у вигляді моделей процеси - як технічні, так і ділові, - щоб зрозуміти потенційні наслідки реалізації цих процесів в різному оточенні і за різних умов. І в цьому випадку знання, отримані в результаті спостережень за поведінкою моделі, дозволяють приймати більш осмислені рішення.

Статичне моделювання дає нам тільки уявлення про систему, її оточення, а також про ділові та технічні процеси, в оточенні яких працює система. Результат моделювання може служити лише для приблизної оцінки поведінки системи. Тому статичне моделювання - лише одне з чотирьох основних засобів підтримки прийняття рішень поряд з імітаційним моделюванням, аналізом і експериментом. У багатьох випадках якоїсь однієї техніки недостатньо для зниження невизначеності до рівня, необхідного для прийняття гарного рішення.

### **Типи моделей**

Можна вважати, що статична модель системи - це спрощене представлення (абстракція) реальності, призначене для імітації зовнішнього вигляду або поведінки системи або її елемента. Не існує стандартної універсальної класифікації моделей. Ми будемо користуватися класифікацією, запропонованою Бланшаром (Blanchard) і Фабрицьким (Fabrycky):

- *схематичні моделі* - це діаграми або схеми, на яких представлений якийсь елемент системи або процес. Прикладом може служити схема організаційної структури або діаграма потоків даних. Ця категорія відома також під назвою «deskriptivні моделі»;
- *математичні моделі*, які використовують математичну нотацію для представлення співвідношення або функції. Прикладами можуть служити закони є

Ньютона, статистичні розподіли і диференціальні рівняння, що описують динаміку системи;

*фізичні моделі*, які безпосередньо відображають деякі або більшість характеристик досліджуваної системи або її елемента. Це можуть бути масштабні моделі транспортних засобів, наприклад літака або судна, або макети в натуральну величину, наприклад передня частина автомобіля, що піддається випробуванню зіткненням з перешкодою. Іноді фізична модель може бути частиною реальної системи, як у попередньому прикладі або в разі літакового шасі, що піддається динамічному випробуванню. Але моделлю є і глобус, на якому зображені земні континенти і океани, і шаростержнева модель молекули. Дослідні зразки також вважаються фізичними моделями.

Три перераховані вище категорії розташовані в порядку збільшення степені реалістичності та зменшення абстрактності, починаючи з контекстної діаграми системи і закінчуючи досвідченими зразками виробу. Бланшар і Фабрицький визначають також категорію «аналогових моделей», які зазвичай є фізичними, але не геометричними еквівалентами. Але для наших цілей їх можна помістити в категорію фізичних моделей.

## **Схематичні моделі**

*Схематичні моделі* - важливий засіб обміну інформацією в системній інженерії, як, втім, і у всіх інженерних дисциплінах. Вони служать для представлення зв'язків у формі діаграм з використанням загальноприйнятої символіки. На технічних кресленнях або малюнках зображується модель проєктованого компонента, а на електричних схемах - модель електронного виробу.

Схематичні моделі - незамінний інструмент комунікації, тому що їх можна легко і швидко намалювати і при необхідності змінити. Але при цьому вони найабстрактніші з усіх, оскільки дають досить обмежене уявлення про систему або про один з її елементів. Тому існує ризик помилкової інтерпретації, який може бути зменшений шляхом чіткого описання сенсу нестандартної і неочевидної термінології. Нижче описується кілька типів схематичних моделей.

**Ескізи.** Хоча ескізи рідко застосовуються в системній інженерії, все ж це один з видів графічної моделі, на якій ілюструються деякі особливості модельованого об'єкта. По-перше, це спрощене - часто до межі-зображення предмета. По-друге, ескіз підкреслює і акцентує окремі риси, звичайно в перебільшеній формі, щоб донести до глядача конкретну ідею. На рис. 2.2, «Ідеальний проєкт ракети з точки зору різних спеціалістів», наочно - і краще, ніж можна було б передати одними словами, - представлена необхідність в системній інженерії. В опис концепції функціонування системи цілком можна було б включити ескіз сценарію використання.

**Архітектурні моделі.** В якості знайомого прикладу використання моделювання при проєктуванні складного виробу можна назвати модель архітектури будівлі, що будується. Якщо замовник хоче, щоб будівля відповідала його вимогам, то зазвичай наймає архітектора, який переводить побажання замовника на мову планів і завдань, в яких точно зазначено, що і, значною мірою, як повинен зробити будівельник. У цьому прикладі архітектор грає роль «системного інженера будівлі», що відповідає за те, щоб дотримати баланс між утилітарними і естетичними бажаннями власника будівлі і обмеженнями на цінову доступність, терміни будівництва і місцеві будівельні норми і правила.

Спочатку архітектор розмовляє з замовником, уточнюючи його побажання відно форми та розмірів будинку, після чого робить кілька начерків. Це графічні моделі, на яких відображені насамперед зовнішній вигляд і прив'язка до місцевості. Одночасно архітектор готує ряд варіантів поверхових планів, щоб замовнику було простіше визначитися із загальною площею і приблизним розміщенням внутрішніх приміщень. Якщо замовник виявить бажання більш детально ознайомитися з тим, як буде виглядати будинок, то архітектор може побудувати масштабну модель з дерева або картону. Це вже можна назвати фізичною моделлю, за формою подібною передбачуваній будівлі. Для будинків зі складною лінією даху або взагалі незвичайної форми така модель може виявитися вдалим капіталовкладенням.

Розглянуті вище моделі служать для обміну проєктною інформацією між замовником і архітектором у формі (графічної), найбільш зрозумілої замовнику.

Але у фактичному будівництві будівлі, як і будь-якої складної системи, бере участь безліч фахівців: теслі, водопровідники, електрики, муляри та ін. Усім їм потрібна більш точна і детальна інформація, яку вони можуть зрозуміти і втілити в конструкцію з відповідних будівельних матеріалів. Ця інформація міститься в кресленнях і завданнях, наприклад: схема електропроводки, повітропроводи системи кондиціонування, сантехнічна арматура і т. п. Креслення - це моделі, зображені в масштабі з зазначенням розмірів, в яких використовуються стандартні галузеві символи для електричних, сантехнічних та інших приладів. Цей різновид моделей, як і рисунки будівлі, представляють фізичні характеристики, але в більш абстрактному вигляді, так як компоненти описуються не картинками, а символами. Ці моделі призначені для передачі детальної проектної інформації будівельникам.

**Структурні схеми системи.** Звичайно, системи набагато складніші звичайних конструкцій. Як правило, вони виконують певні дії у відповідь на зміни в оточенні. Тому для опису та передачі інформації про їх структуру та поведінку необхідні моделі різних типів.

Одна з найпростіших моделей - «структурна схема». Ієрархічні структурні схеми зображуються у вигляді дерева, в якому структура гілок відповідає зв'язкам між компонентами на сусідніх рівнях системи. На верхньому рівні знаходиться єдиний блок, який представляє систему в цілому; на другому рівні - блоки, відповідаючі підсистемам; на третьому кожна підсистема розбивається на компоненти і т. д. Блоки на кожному рівні з'єднані лініями з батьківським блоком. На рис.9.2 показана узагальнена структурна схема системи, що складається з трьох підсистем і восьми компонентів.

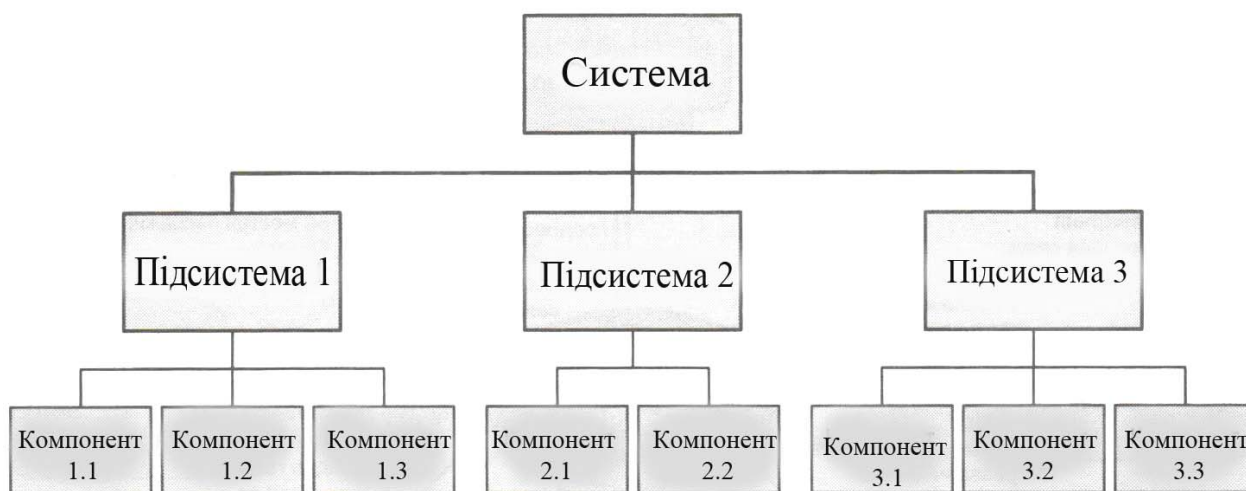


Рис. 9.2. Традиційна ієрархічна система.

Структурну схему можна вважати дуже абстрактною моделлю, зосередженою виключно на структурних модулях, з яких складена система, і фізичних зв'язках між ними. Прості прямокутні блоки - це чиста символіка, тут навіть не робиться спроб показати фізичну форму елементів системи. Зате така схема дуже чітко передає важливий тип зв'язків між елементами системи, а також її принципову організацію. Для представлення більш складних взаємодій

між підсистемами і компонентами існують більш детальні діаграми і описи. Взаємодії між блоками можна уявити написами на з'єднувальних лініях.

**Контекстні діаграми системи.** Ще одна корисна для системного проектування модель - це контекстна діаграма, на якій представлені всі зовнішні об'єкти, з якими може взаємодіяти система прямо або побічно. Ми вже бачили контекстну діаграму на рис. 3.2. На ній система зображається в центрі, без будь-яких деталей внутрішньої структури, в оточенні всіх впливаючих на неї систем, середовищ і впливів. Мета контекстної діаграми системи - привернути увагу до зовнішніх чинників і подій, які слід розглянути при формуванні повного набору вимог до системи і обмежень. При цьому необхідно візуалізувати не тільки умови експлуатації, але також стадії на шляху до експлуатації: установка, комплексування, перевірка придатності до експлуатації і т. п.

### **Математичні моделі**

Математичні моделі служать для опису функціонування і залежностей мовою математики. Вони найбільш корисні у випадках, коли деякі елементи системи можна проаналізувати окремо, а найважливіші особливості їхньої поведінки - представити за допомогою добре вивчених математичних конструкцій. Якщо модельований процес містить випадкові величини, то краще застосувати імітаційне моделювання. Важлива перевага математичних моделей полягає в тому, що вони широко відомі і добре вивчені. Взагалі, похибками неминучою апроксимації можна знехтувати (але забувати про дане нехтування не варто), і результатам математичного моделювання можна довіряти. Математичні моделі здатні представити детерміновані (невипадкові) функції або процеси в різних формах. Типовими прикладами є рівняння, графіки і електронні таблиці.

Наближені обчислення - це приклад використання математичних моделей, які є абстрактними уявленнями обраних функціональних характеристик досліджуваного елемента системи. Такі моделі дозволяють виявити ті змінні, які визначають основні особливості поведінки системи, і знехтувати ефектами вищого порядку, які без потреби лише ускладнили б математику. Таким чином, вони спрощують розуміння ключових особливостей функціонування елемента системи.

Як і в будь-якій моделі, результати наближених обчислень слід інтерпретувати, не забуваючи про обмеження, пов'язані з знехтуванням змінними, які можуть виявитися значущими. Якщо результат перевірки на правдоподібність істотно відрізняється від результату обчислень що перевіряється, то, перш ніж піддавати сумніву початково отриманий результат, слід досліджувати наближення та інші допущення.

Виробляючи навички використання наближених обчислень, системний інженер повинен оцінити, наскільки глибоко слід вивчати фундаментальні технічні дисципліни в кожному конкретному випадку. Один з можливих підходів - задовольнитися опитуванням проектувальників, які виробляли аналіз спочатку. Інший - попросити фахівця виконати незалежну перевірку. Третій - скористатися власними знаннями, доповнивши їх зверненням до підручника або довідником, і виконати наближені обчислення самостійно.

Який підхід вибрати, звичайно, залежить від ситуації. Але краще, коли в деяких критично важливих технічних дисциплінах системний інженер володіє достатніми знаннями фундаментальних основ, щоб без побоювання виносити незалежні судження. Вироблення таких навичок - частина особливої ролі системного інженера в об'єднанні фахівців у різних дисциплінах, оцінці ризиків системи та визначенні тих областей, в яких потрібен аналіз, розробка або експеримент.

**Елементарні співвідношення.** У будь-якому розділі техніки і фізики існує ряд елементарних співвідношень, з якими повинен бути знайомий системний інженер. Закони Ньютона застосовні до всіх систем, що належать до засобів пересування. Якщо мова йде про напружені елементи конструкцій, то корисно знати формули, що зв'язують властивості міцності і пружності балок, циліндрів та інших простих конструкцій. У разі електронних компонентів системний інженер повинен розуміти елементарні властивості електричних схем. У більшості технічних дисциплін існують евристичні правила, зазвичай засновані на елементарних математичних формулах.

**Статистичний розподіл.** Будь-який інженер знайомий з гаусовим (нормальним) розподілом, що характеризує випадковий шум та інші природні явища. Є й інші цікаві функції розподілу, наприклад розподіл Релея, корисний при аналізі зашумлених сигналів РЛС, розподіл Пуассона, експоненціальний і біноміальний розподіл. Всі вони описуються простими математичними формулами.

**Графіки.** Якщо модель являє емпіричні співвідношення, для яких не існує явних математичних рівнянь, то вона звичайно зображується у вигляді графіка. На рис. 2.1 у розділі 2 наведено графік, який ілюструє типовий зв'язок між функціональними можливостями і витратами на їх розробку. Такі моделі використовуються в основному для представлення якісних концепцій, хоча дані, зображені у вигляді графіка, можуть показувати і кількісні зв'язки. Стовпчасті діаграми, які використовуються, наприклад, для показу зміни обсягів виробництва по місяцях або вартості альтернативної продукції, теж є моделями, які служать для отримання інформації у формі, більш зручною в порівнянні зі списком чисел на аркуші паперу.

### **Фізичні моделі**

Фізичні моделі безпосередньо відображають деякі або навіть більшість фізичних характеристик досліджуваної системи або її елемента. У цьому сенсі фізичне моделювання - найменш абстрактний і, значить, самий зрозумілий спосіб. Однак фізична модель за визначенням є спрощенням модельованого об'єкта чи явища. Це може бути лише частина виробу, або масштабна копія, або дослідний зразок. У таких моделях багато застосувань протягом циклу розробки, деякі з них ілюструються на прикладах нижче.

**Масштабні моделі.** Це (звичайно) зменшені копії будівлі, транспортного засобу чи іншої системи, які часто застосовуються, щоб можна було скласти уявлення про зовнішній вигляд виробу. Прикладом використання масштабних моделей в техніці може служити випробування моделі літального апарату в аеродинамічній трубі або підводного апарату в гідродинамічній трубі або в дослідницькому басейні.



**Макети.** Це варіант транспортного засобу, частини будівлі або іншої конструкції, виконаний у натуральну величину, для використання на пізніх стадіях розробки систем, що містить пристосування для операторів та іншого персоналу. Макети дають реалістичне уявлення про людино-машинному інтерфейс, що дозволяє провести валідацію проектних рішень і, можливо, внести зміни перед тим, як приступати до детального проектування інтерфейсів.

**Дослідні зразки.** У попередніх розділах обговорювалися створення та випробування дослідних зразків на етапах розробки, конструювання і виробництва залежно від даної системи. Вони також є фізичними моделями, щоправда, володіють більшістю властивостей готової системи. Проте, строго кажучи, це моделі.

### **Програмно-апаратне моделювання**

Програмно-апаратне моделювання є різновидом фізичного моделювання, при якому апаратні засоби системи використовуються в поєднанні з імітаційним моделюванням на комп'ютері. Прикладом може служити система самонаведення ракети. Для реалістичних експериментів з вивчення динаміки наведення така система обладнується матеріалами, що поглинають НВЧ-випромінювання, рухливими джерелами випромінювання та реальною головою самонаведення. У результаті виходить програмно-апаратна модель з апаратними засобами в контурі управління (hardware-in-the-loop simulation), що реалістично представляє складне оточення.

Інший приклад програмно-апаратної моделі дає керований комп'ютером руховий стенд, що застосовується для випробування інерційних компонентів і платформ в ході розробки. На стенд подаються сигнали, що змушують піддавати компоненти переміщенню і вібраціям, характерним для руху проектованої платформи; при цьому він забезпечений контрольно-вимірювальною апаратурою для вимірювання похибки результуючих показників приладів.

### **Технічне моделювання**

Для проектування компонентів і субкомпонентів існують інженерні інструменти, що доповнюють математичні моделі, які були описані в попередніх розділах. Використовуються вони в основному фахівцями-проектувальниками, але й системні інженери повинні бути знайомі з їх можливостями і обмеженнями, щоб розуміти, для чого вони застосовуються.

У наш час електричні схеми не проектуються шляхом копіювання елементів макетних плат. Для отримання, перевірки та модифікації необхідних функціональних блоків можна використовувати емулятори, повторюючи весь цикл до отримання бажаних результатів. Існують інструментальні засоби, здатні автоматично скласти документацію та виробити готову плату «в залізі».

Аналогічно розрахунок таких складних конструкцій, як будівлі і мости, можна виконати за допомогою інструментів моделювання. Цей варіант імітаційного моделювання дозволяє врахувати безліч складних взаємодій між механічними елементами, що входять до складу конструкції, - завдання, яке на практиці важко вирішити за допомогою аналізу і випробувань.

Описані вище методики буквально зробили революцію в багатьох аспектах проектування, розробки, випробування та виробництва обладнання. Системні інженери, що працюють в цих областях, просто зобов'язані особисто оцінити застосування і можливості технічного моделювання, щоб ефективно очолювати інженерно-технічну роботу.

### **Моделювання оточення**

Моделювання оточення використовується переважно на етапі технічних випробувань та атестації системи. Це різновид фізичного моделювання, коли моделюється не сама система, а елементи навколишнього середовища. Здебільшого такі моделі є динамічними, детермінованими і дискретними.

Ці моделі призначені для вивчення умов експлуатації (зазвичай небезпечних), яких важко або занадто дорого організувати для валідації проектних рішень, прийнятих при розробці системи або її елементів, а також тих, що необхідні для підтримки та забезпечення роботи системи. Нижче наведено кілька прикладів.

**Механічні випробування на міцність.** Система або її елементи, які повинні працювати в несприятливих умовах, наприклад ракети, авіаційні комплекси, космічні апарати і т. Д., Необхідно піддавати навантаженням, що імітують такі умови. Зазвичай для цього застосовуються вібраційні стенди, віброперетворювачі і випробування на удар.

**Випробування на зіткнення з перешкодою.** Прагнучи відповідати стандартам безпеки, автовиробники піддають свої вироби випробуванням на зіткнення з перешкодою (краш-тестам), в процесі яких кузов автомобіля приноситься в жертву заради отримання даних про те, якою мірою його конструкція знижує нанесені пасажирів ушкодження. Робиться це за допомогою манекенів, оснащених різноманітними датчиками, що вимірюють ступінь впливу удару при зіткненні. Як випробування, так і аналіз їх результатів зазвичай комп'ютеризовані.

**Випробування в аеродинамічній трубі.** При розробці літальних апаратів неоціненну користь надає аеродинамічна труба. Хоча сучасні комп'ютерні програми здатні моделювати сили, що діють на тіло в потоці рідини чи газу, складність поведінки, особливо поблизу звукового бар'єру, і взаємодії між різними поверхнями тіла часто робить необхідним випробування на стендах, де для дослідження аеродинамічних моделей тіла або його складових частин створюються контрольовані умови в повітряному потоці.

Як зазначалося вище при обговоренні масштабних моделей, схожі імітаційні моделі застосовуються при розробці корпусів і рульових пристроїв надводних і підводних судів - з використанням гідродинамічних труб і дослідному басейні.

### **Моделювання віртуальної реальності**

Завдяки потужності сучасних комп'ютерів з'явилася практична можливість створювати тривимірне візуальне оточення для спостерігача, який може в реальному масштабі часу реагувати, виходячи зі свого фактичного або змодельованого місця розташування і напрямку погляду. Досягається це шляхом збереження всіх координат оточення в базі даних, перерахунку виду оточення з поточної позиції спостерігача під поточним кутом зору і

проектування цього виду на екран або інший індикаторний пристрій, зазвичай монтується в шоломі спостерігача. Нижче описуються деякі приклади моделювання віртуальної реальності.

**Просторові моделі.** Просторова модель віртуальної реальності часто буває корисна, коли важливо візуалізувати внутрішній об'єм замкнутого простору, а також входи і виходи з нього. Існують комп'ютерні програми, що дозволяють швидко проектувати такі простори і їх внутрішній стан. Функція віртуальної реальності дає спостерігачеві можливість «прогулюватися» по цьому простору в будь-якому напрямку. Моделі подібного роду можуть бути корисні для попереднього проектування будівель, споруд, центрів управління, складів, частин судна і навіть планування заводських цехів. У них може бути додаткова функція - друк двовимірних або тривимірних зображень з написами і розмірами.

Для побудови просторової віртуальної моделі необхідно ввести в комп'ютер детальний опис тривимірного простору і предметів, що знаходяться в ньому. У модель вводяться також дані про становище спостерігача, одержувані від шолома або якого-небудь вказівного пристрою - джойстика, миші і т.п. Віртуальне зображення обчислюється в реальному масштабі часу і проектується або на індикатор в шоломі спостерігача, або на екран дисплея.

## **КОРОТКІ ВИСНОВКИ**

### **Прийняття рішень**

Прийняття рішень - процес, що складається з декількох кроків. Ступінь формалізації кожного кроку залежить від типу і складності рішення. Ми описуємо інфраструктуру для дослідження трьох типів рішень: структурованих, слабо-структурованих і неструктурованих. Ця класифікація не є дискретною «як можна було б подумати» а представляє континуум, що охоплює як типові / стандартні / вивчені структуровані рішення, так і нетипові / інтуїтивні / суб'єктивні неструктуровані рішення.

Процес прийняття рішень давно описаний і вивчений і зазнав мало змін. Він складається з чотирьох етапів: підготовка і дослідження, проектування та оцінка моделі, вибір з кількох альтернатив і реалізація.

### **Моделювання протягом розробки системи**

Моделювання направляє прийняття рішень в умовах складності та невизначеності; модель дозволяє прояснити поведінку і ключові зв'язі. Зокрема, імітаційні моделі - це засіб моделювання динамічної поведінки. Інші засоби, в тому числі різні методи аналізу компромісів, моделюють процес вибору з декількох альтернатив.

### **Моделювання для прийняття рішень**

Моделі можна віднести до трьох категорій.

1. У схематичних моделях для представлення елементів системи або процесів використовуються діаграми. Прикладом можуть служити архітектурні ескізи креслення, такі як поверхові плани. Організація системи

представляється у вигляді блок-схем. Часто вони мають вигляд дерева, що представляє ієрархічну структуру, або містять прості прямокутні блоки для представлення фізичних та інших елементів.

На контекстних діаграмах системи показані всі взаємодіючі з нею зовнішні об'єкти; сама система представлена у вигляді «чорного ящика» (без деталізації внутрішнього устрою). На контекстній діаграмі описуються взаємодії системи з її оточенням.

На схемах функціональних потоків моделюються функціональні взаємодії; функціональні елементи представлені прямокутниками, а взаємодії та потоки інформації, матеріалів або енергії між елементами - стрілками. Назви елементів починаються дієсловом, що позначає дію. В якості прикладів схеми функціональних потоків і її розширень можна привести моделі життєвого циклу системи, IDEF0-діаграми і діаграми F2D2.

Схеми функціональних потоків для процесу аналогічні - вони утворюють ієрархічний опис складного процесу. На них також відображаються зв'язки між компонентами складного процесу і вимогами і специфікаціями. Прикладами схематичних моделей можуть служити діаграми, визначені в мовах UML і Systems Modeling Language (SysML) (див. Розділ 8).

2. У математичних моделях для представлення зв'язків використовується математична нотація. Вони являють собою важливу підмогу для розробки системи і можуть бути корисні як для проєктувальників, так і для системних інженерів. За допомогою математичних моделей також виконується перевірка результатів складного аналізу та імітаційного моделювання на правдоподібність.

3. Фізичні моделі - це фізичне уявлення системи або її елементів. Вони активно використовуються при проєктуванні і випробуванні системи. До них відносяться експериментальні моделі, макети і дослідні зразки.

### **Імітаційне моделювання**

На імітаційних моделях вивчається динамічна поведінка системи та її елементів, вони застосовуються на кожному етапі розробки системи. Керівництво імітаційним моделюванням покладається на системних інженерів.

Комп'ютерні «військові ігри» - приклад імітаційного моделювання бойових операцій, в якому дві команди грають за противні сторони. Вони використовуються для оцінки бойової ефективності різних тактичних і системних варіантів.

При імітаційному моделюванні ефективності системи розглядаються альтернативні архітектури, такі моделі застосовуються на етапі розробки концепції для порівняльної оцінки. Проєктування моделей ефективності - сама по собі складна інженерна задача. При розробці складних моделей необхідно дотримуватися балансу між похибкою і вартістю, тому що такі моделі цілком можуть виявитися самостійними системами. Для ефективного та своєчасного отримання результатів необхідно контролювати кордони робіт.

Фізичні або засновані на фізичних законах моделі використовуються при проєктуванні транспортних засобів з високими робочими характеристиками та

інших динамічних систем. Вони можуть значно скоротити час розробки і знизити витрати.

До складу програмно-апаратних моделей входять апаратні компоненти в поєднанні з керованими комп'ютером механізмами. Це фізичні моделі динамічних умов експлуатації.

При імітаційному моделюванні умов навколишнього середовища система та її елементи поміщаються в несприятливі умови. Модель створює штучне оточення для перевірки відповідності системи вимогам до функціонування.

Нарешті, комп'ютерні інженерні інструменти суттєво спрощують проектування електричних схем, розрахунок конструкцій на міцність і вирішення інших інженерних завдань.

### **Аналіз компромісів**

Процеси пошуку компромісу свідомо чи підсвідомо присутні в будь-якому рішенні, яке ми приймаємо (в особистому житті або на роботі). Важливим аспектом дослідження компромісів виступає моделювання альтернатив. Кінцевим результатом дослідження є вибір «кращого» з двох або більше варіантів. Для прийняття особливо важливих рішень (типових для системної інженерії) потрібен офіційний аналіз компромісів.

Аналіз компромісів, офіційний чи неофіційний, включає наступні кроки:

1. Визначити мету.
2. Відібрати відповідні альтернативи.
3. Визначити критерії вибору у формі показників ефективності.
4. Призначити критеріям вибору вагові коефіцієнти, що відображають їх відносну важливість для вирішення.
5. Вибрати або розробити спосіб оцінки по кожному критерію.
6. Обчислити або отримати шляхом вимірювання порівняльні оцінки за кожним критерієм; для кожної альтернативи проаналізувати поєднання отриманих оцінок.
7. Проаналізувати отримані результати і їх стійкість.

При необхідності піддати результати ревізії і відкинути альтернативи, що не задовольняють якійсь суттєвій вимозі. Наприклад, можна відмовитися від недостатньо виборчих показників ефективності, які приблизно однакові для всіх альтернатив. Точність значень обмежуйте точністю самого грубого вимірювання, досліджуйте повний «профіль» оцінок кожного кандидата.

Дослідження та аналіз компромісів - це лише підмога для прийняття рішень, а не вірна формула успіху. Чисельні результати створюють безпідставну ілюзію точності й переконливості. Нарешті, якщо формальний переможець не перевищує альтернативних варіантів за всіма статтями, то необхідний подальший аналіз.

### **Методи оцінювання**

Оскільки системна інженерія стикається зі складними рішеннями про події з невизначеним результатом, їй на допомогу пропонується набір корисних інструментів і прийомів. Ми розглянули п'ять таких інструментів.

1. Багатовимірною теорією корисності (МАИТ) використовується функція корисності для перетворення критерію вибору в безрозмірне значення корисності. Значення корисності потім можна об'єднати і отримати підсумкову оцінку для кожної альтернативи.
2. Метод аналізу ієрархій - це математично обґрунтована методика »в якій критерії та альтернативи порівнюються попарно для отримання як вагових коефіцієнтів, так і порівняльних оцінок альтернатив.
3. Дерева рішень - це графи для подання можливих варіантів вибору. Кожному варіанту можна зіставити значення і показник невизначеності (у вигляді ймовірності), а потім визначити очікувані результати для альтернативних шляхів вирішення.
4. Аналіз витрати-ефективності зазвичай застосовується для статичного та імітаційного моделювання з метою розрахунку ефективності кожної альтернативи на одиницю витрат.
5. У методі структурування функції якості (QFD) визначається матриця (будинок якості), в якій відображені зв'язки між потребами замовника, вимогами до системи, компонентами системи і важливістю компонента для проекту в цілому. Матриця може бути досліджена для отримання кількісних оцінок альтернатив.

## Лекція 10. ПРОГРЕСИВНА РОЗРОБКА

Етап ескізного проектування - це та частина життєвого циклу розробки системи, на якій шляхом аналізу, моделювання, розробки і створення дослідних зразків вирішується велика частина проблем, обумовлених невизначеностями, властивими обраній концепції системи. Основна мета етапу ескізного проектування - зниження потенційних ризиків розробки нової складної системи до рівня, на якому функціональні проекти всіх раніше неперевіраних підсистем і компонентів пройшли валідацію. Як наслідок, ризик виявлення серйозних проблем повинен бути достатньою мірою низький, що дозволяє з упевненістю приступити до повномасштабної інженерно-технічної розробки. Головні завдання на даному етапі - виробити (якщо необхідно) ґрунтовний і стабільний технічний підхід до проектування системи, провести його валідацію і продемонструвати особам, що дають дозвіл на повномасштабну розробку.

**Аналіз функціонування та проектування.** Типові дії включають:

- аналіз прив'язки функцій до компонентів і субкомпоненту, виявлення аналогічних функціональних елементів в інших системах;
- виконання аналізу та імітаційного моделювання для вирішення проблем, що залишилися, пов'язаних з функціонуванням.

**Розробка прототипу.** Типові дії включають:

- виявлення проблем фізичної реалізації, пов'язаних з використанням неперевіреної технології, і визначення рівня аналізу, розробки і випробувань, необхідного для зниження ризиків до прийнятної величини;
- проектування критично важливого програмного забезпечення;
- проектування, розробку і побудову дослідних зразків критичних компонентів і підсистем;
- усунення недоліків, виявлених в ході випробування та атестації.

**Стендові випробування.** Типові дії включають:

- розробку планів випробувань і критеріїв оцінки критичних елементів, а також розробку, закупівлю і резервування спеціального випробувального устаткування і оснащення;
- проведення випробувань критичних компонентів, оцінку результатів і повідомлення про потребу виправлення виявлених недоліків або надмірно завищених вимог; мета полягає в тому, щоб отримати зрілий, що пройшов валідацію проект системи.

**Аналіз вимог**

Як було зазначено вище на етапі ескізного проектування насамперед переслідуються дві мети:

1. Повторна перевірка правильності функціональних вимог до системи, розроблених на етапі вибору концепції або пізніше.
2. Виявлення в обраній концепції системи тих компонентів, які ще недостатньо відпрацьовані для повноцінного технічного застосування

(тобто не пройшли перевірку в наявних системах) і тому повинні бути до роблені на етапі ескізного проектування.

### **Функціональні вимоги до системи**

При виборі кращої концепції системи функції системи були прив'язані до основних підсистем, які були далі розбиті на функціональні елементи. Ці концепції функціонального проекту потім втілювалися в документ, що містить вимоги до системи, який є входом для етапу ескізного проектування.

При аналізі даних вимог слід враховувати обставини проведення етапу вибору концепції. Якщо, як часто буває, на нього було відведено всього кілька місяців і виділено обмежене фінансування, особливо якщо це відбувалося в умовах конкурентної боротьби, то результати слід розглядати як попередні, допускають модифікацію і тому підлягають особливо ретельному аналізу. До прийнятих раніше проектних рішень слід ставитися з часткою скептицизму до тих пір, поки вони не вивчені і не визнані обґрунтованими. Це не означає, що обрані технічні підходи обов'язково слід переглянути, просто не слід приймати їх, не розібравшись, як вони були отримані.

### **Простежування вимог**

Щоб по-справжньому розібратися в значимості і чутливості функціональних вимог до системи, потрібно простежити, як вони були виведені з Вимог до показників функціонування. Розуміння взаємозв'язку між функціональними вимогами і показниками функціонування необхідно, щоб приймати проектні рішення, придатні для фізичного втілення функцій за допомогою апаратних і програмних засобів.

Слід повернутися до сценарію підтримки життєвого циклу системи і заново визначити, які функції необхідні з урахуванням різних обставин, в яких може опинитися система як до початку експлуатації, так і в процесі використання. Крім того, потрібно розглянути вимоги до сумісності, надійності, ремонтпридатності і готовності, екологічності, а також до показників функціонування. На цьому етапі до складу вимог до підсистем і компонентів включаються вимоги, що ставляться до людино-машинного інтерфейсу і безпеки.

Як вже зазначалося, деякі вимоги часто сформовані не повністю, тоді як інші не піддаються вимірюванню. Наприклад, цінова доступність і потенціал зростання системи нерідко відсутні в явному вигляді. Вимоги до інтерфейсу користувача часто виражені в якісній формі, так що виміряти їх неможливо. Зв'язок кожного із зазначених питань з функціональним проектом повинна бути зрозуміла і документована.

### **Зв'язок з вимогами призначення**

Якщо не відразу ясно, як задовольнити якісь вимоги до системи, то необхідно глибше розібратися, наскільки вони обґрунтовані, простеживши їх виникнення ще на крок далі, тобто до взаємозв'язку з можливістю використання системи за призначенням, інакше кажучи, до вимог призначення. Часто цей зв'язок втрачається на ранніх етапах опису системи і повинен бути відновлений, щоб системний інженер міг зі знанням справи підходити до вирішення проблем, що неминуче виникають в процесі розробки.



### **Зв'язок з попередніми системами**

Якщо новій системі передувала система зі схожими функціями, як зазвичай і буває, то важливо чітко розуміти, в чому полягають подібність і відмінність, а також як і чому нові вимоги відрізняються від старих. Зокрема, потрібно зрозуміти, які недоліки попередньої системи і як передбачається усунути їх у новій.

Корисність такого порівняння, звичайно, залежить від доступності ключових осіб та архівних записів про розробку попередньої системи. Але, як мінімум, порівняння повинно вселити додаткову впевненість в обраному підході або навести на думку про альтернативи, які варто вивчити. У тих випадках, коли можна звернутися до ключових учасників розробки, їх рекомендації з приводу потенційних проблем і витягнутих уроків можуть виявитися безцінними.

### **Виявлення компонентів, які потребують розробки**

Вище було відзначено, що основна мета етапу ескізного проектування - переконатися, що всі компоненти системи готові до повномасштабної інженерно-технічної розробки. Це означає, що проекти компонентів не викликають сумнівів і можуть бути реалізовані без суттєвого ризику того, що при їх реалізації виявляться функціональні або фізичні дефекти, які зажадають іншого підходу до задоволення вимог.

З даного твердження випливає, що всі компоненти системи повинні бути достатньо зрілими, тобто відпрацьовані до такої міри, коли всі з проектні проблеми, будь-якого значення, вже вирішені. Процес, в ході якого підвищується рівень зрілості, називається «розробкою», і тому основний зміст етапу ескізного проектування полягає в розробці тих компонентів системи, які раніше не були доведені до необхідного рівня зрілості, що гарантує успішне функціонування. Це, в свою чергу, означає, що всі компоненти, визнані недостатньо зрілими для використання в повномасштабній інженерно-технічній розробці, мають бути доопрацьовані, а їх проект - підданий валідації. Компоненти, які визнані настільки зрілими, що не потребують доопрацювання, все одно повинні пройти валідацію шляхом аналізу або випробування до того, як будуть використані в інженерно-технічній розробці.

**Оцінка зрілості компонента.** Визначити, чи достатньо відпрацьований компонент для безпосереднього використання в повномасштабній інженерно-технічній розробці, можна, тільки порівнявши його з аналогічними компонентами, які були успішно розроблені і виготовлені раніше. Якщо перевірених аналогів не існує, то порівняння найчастіше можна розбити на дві частини, функціональну і фізичну, задавши наступні питання.

1. Чи існують компоненти з близькими функціональними можливостями і показниками функціонування, які були перевірені і успішно пройшли випробування? Якщо є помітні відмінності, то чи вірно, що ці відмінності не виходять за загальновизнані кордони показників для компонентів даного типу?
2. Чи існують компоненти, в конструкції яких використовуються схожі матеріали і компоновка? Чи вірно, що закладені в проект напруга, допуски,

безпека і характеристики протягом терміну служби лежать в загальноновизнаних межах, прийнятих для схожих існуючих компонентів?

**Аналіз ризику.** Виявивши елементи системи, що вимагають додаткової розробки, потрібно далі визначити її характер і об'єм. Саме тут особливо важливі досвід і знання системного інженера, оскільки ці рішення передбачають пошук точного балансу між зростанням витрат у міру підвищення ретельності розробки з одного боку і зростанням ризику через недостатню увагу до деталей - з іншого. Нижче містяться посилання на методологію оцінки ризику при розробці системи, а більш докладно це питання розглядається в окремому розділі в кінці розділу.

**Планування розробки.** Зі сказаного вище зрозуміло, що планування етапу ескізного проектування повинне бути засноване на оцінці зрілості кожного компонента пропонованого проекту системи з метою: 1) визначити конкретний характер неперевіраних особливостей проектних рішень і 2) вирішити, якого роду аналіз, розробка та випробування потрібні для дозволу питань, що залишилися. Для більшості нових систем невизначеності концентруються в обмеженому числі критичних областей, тому зусилля з розробки можуть бути зосереджені лише на цих недостатньо зрілих компонентах.

**Бюджет зниження ризику.** Результати проведеного аналізу ризиків та рішення про те, які заходи потрібно вжити для їх зниження, слід включити в детальний план розробки та керуватися ним при проведенні робіт з аналізу, розробці та перевірці відповідності на етапі ескізного проектування. Важливою стороною цієї діяльності є ретельне продумування того, як розподілити ресурси і сили між окремими компонентами і підсистемами, які планується розробити. Чи дозволить запропонований розподіл досягти належного балансу між потенційним виграшем і капіталовкладеннями? Чи достатньо виділених ресурсів для отримання необхідних даних? Якщо, як це часто буває, наявних ресурсів не вистачає для вирішення всіх запланованих завдань, то звичайно краще замінити якісь найбільш сумнівні компоненти більш традиційними, ніж зазнати невдачі через те, що нові компоненти не пройдуть валідацію при їх використанні і складі системи. Таким чином, план розробки та зниження ризиків повинен включати бюджет пом'якшення ризиків з окремими статтями для кожної значущої задачі розробки.

### **Використання імітаційних моделей**

Хоча багато розглянутих вище проблем можуть бути вирішені тільки шляхом створення прототипів фактичних апаратних або програмних елементів, є також ряд таких, які вдається ефективно досліджувати на імітаційній моделі. Наведемо кілька прикладів.

- *Динамічні елементи.* Якщо не вважати динамічних ефектів з дуже високою частотою, то динаміку системи в більшості випадків можна з великою точністю вивчити на моделі. Так, динаміку літака або ракети з шістьма ступенями свободи можна вивчити дуже детально.

- *Людино-машинні інтерфейси.* Користувальницький інтерфейс - це елемент управління в більшості складних систем. Щоб його правильно спроектувати, необхідна активна участь потенційних користувачів. Залучити їх до такої участі найпростіше, створивши прототип інтерфейсу на ранніх етапах розробки і покращуючи його в міру накопичення досвіду.
- *Сценарії функціонування.* Діючі системи зазвичай виявляються в різноманітних обставинах, що роблять на систему різний вплив. Для вивчення таких впливів корисна імітаційна модель з можливістю задавати вихідні умови, причому проводити моделювання потрібно задовго до створення дослідних зразків системи і випробування в реальних умовах.

## **КОРОТКІ ВИСНОВКИ**

### **Зниження ризиків програми**

Мета етапу ескізного проектування полягає в тому, щоб усунути більшу частину невизначеностей (ризиків) в ході аналізу та розробки і провести валідацію прийнятого підходу до проектування системи в якості основи для подальшої повномасштабної інженерно-технічної розробки. Результатом ескізного проектування є проектна документація і пройшла валідаційну модель (макет, зразок) системи.

Ескізне проектування особливо критично для систем, в яких широко застосовуються передові технології або неперевірені концепції, на розробку яких можуть піти роки.

На етапі ескізного проектування виконуються наступні дії:

- аналіз вимог - співвіднесення функціональних вимог з потребами;
- аналіз функціонування та проектування - виявлення проблем, пов'язаних з функціонуванням;
- розробка прототипу - побудова та випробування прототипів критичних компонентів;
- випробування та атестація - валідація зрілості критичних компонентів.

### **Аналіз вимог**

Аналіз функціональних вимог до системи необхідний для того, щоб співвіднести їх з вихідними вимогами призначення, особливо з тими, задовольнити які нелегко. Виявляються також відмінності від попередньої системи.

### **Аналіз функціонування та проектування**

До числа компонентів, які можуть вимагати доопрацювання, відносяться такі, в яких:

- реалізується нова функція;
- здійснюється нова реалізація існуючої функції;

- використовується нова технологія виробництва стосовно відомого типу компонентів;
- розширюються функціональні можливості раніше перевіреного компонента;
- є складні функції, інтерфейси та взаємодії.

### **Розробка дослідного зразка як методика пом'якшення ризику**

Ризики програми, що вимагають додаткового вивчення, можуть бути пов'язані з різними обставинами:

- незвично жорсткі вимоги до показників функціонування;
- нові матеріали і процеси;
- екстремальні умови зовнішнього середовища;
- складні інтерфейси між компонентами;
- нові програмні елементи.

### **Стендові випробування**

Перевірка придатності до експлуатації ставить метою усунути ризики, для її виконання потрібно складання генерального плану випробувань (TEMP). Крім того, необхідно розробити випробувальне обладнання, провести оціночні випробування, а також проаналізувати і оцінити їх результати. За результатами випробувань проводиться усунення виявлених недоліків проекту. Однак створення спеціального випробувального устаткування і споруд часто вимагає значних капіталовкладень. Тому необхідно експериментальне дослідження проекту інтерфейсів на ранніх етапах.

При розробці систем широко використовуються моделі підсистем і компонентів. Імітаційне моделювання дуже важливо на всіх стадіях розробки, а особливо при аналізі динамічних систем і програмного забезпечення, для розробки таких моделей потрібні спеціальні зусилля штату аналітиків і операторів.

Засоби розробки - це установки, що імітують умови навколишнього середовища; вони використовуються для стендових випробувань та атестації компонентів. Створення таких установок потребує значних капіталовкладень, а їх експлуатація - постійного персоналу.

### **Зниження ризику**

Оцінка ризику - базовий інструмент системної інженерії; вона виробляється на

всьому протязі розробки, але особливо на етапі ескізного проектування. Включає виявлення джерел ризику, визначення ймовірності та критичності ризику.

**ПРОГРАМНА СИСТЕМНА ІНЖЕНЕРІЯ. ТЕХНІЧНЕ ПРОЕКТУВАННЯ.**

Термін «інженерія програмних систем» (software systems engineering) запропонував доктор Уїнстон Ройс (Winston Royce), винахідник каскадної діаграми, яка на зорі розвитку програмної інженерії дозволила представити природний зв'язок між обома дисциплінами. Однак цей термін не був прийнятий зростаючим спільнотою, так що за галуззю затвердилася назва «програмна інженерія» (software engineering).

У першому десятилітті XXI століття обидва спільноти визнали той факт, що ці дві дисципліни мають багато спільного. І «старий» термін воскресили, щоб підкреслити застосування принципів та методик системної інженерії до розробки програмного забезпечення. Зрозуміло, потік ідей поширюється в обох напрямках, даючи початок появі нових концепцій в системній інженерії, прикладом може служити об'єктно-орієнтована системна інженерія (object-oriented systems engineering - OOSE). Сьогодні неможливо заперечувати зростаючу роль програмного забезпечення в створенні сучасних складних систем.

Однак терміни «програмна інженерія» » і «інженерія програмних систем» - не синоніми. Перший відноситься до розробки і постачання програмних продуктів, автономних або вбудованих, а другий - до використання певних принципів стосовно до програмної інженерії, як галузі знання.

Розробка програмного забезпечення (друга половина XX століття) співпала за часом з розвитком технології цифрових обчислень, на яку, в свою чергу, вплинув прогрес в напівпровідникових технологіях. Програмне забезпечення - це керуючий і обробляє елемент в системах збору та обробки даних. Це саме той засіб, за допомогою якого цифровому комп'ютеру повідомляється, як чинити з джерелом даних, щоб перетворити дані в корисну інформацію або дію. На зорі розвитку комп'ютерів програми використовувалися для розрахунку артилерійських таблиць на дуже примітивних машинах, результати цих розрахунків використовувалися при проведенні бойових дій Другої світової війни. У наші дні ПО застосовується для управління різними комп'ютерами (від одиночних систем на кристалі до неймовірно потужних суперкомп'ютерів), які виконують нескінченно різноманітні завдання. Така гнучкість і потенційна міць робить ПО незамінною складовою частиною сучасних систем - як простих, так і складних.

За останні кілька десятиліть було багато спроб запропонувати класифікацію ПЗ, але ми вважаємо, що майже всі категорії можна віднести до одного з трьох загальних типів.

**Системне ПЗ.** Ця категорія програмного забезпечення надає служби (послуги) іншим програмам і не призначена для автономного використання. Класичний приклад - операційна система. Операційна система комп'ютера або сервера надає різні служби даних, файлів, зв'язку та інтерфейсів (і це лише мала частина) іншим програмам, що працюють на тому ж комп'ютері.

**Вбудоване ПЗ.** Ця категорія програмного забезпечення надає конкретні служби (послуги), функції та компоненти у складі більшої системи. Даний тип

ПО найбільш близький системної інженерії, тому що відповідно до базовим принципом функціональні можливості прив'язуються до конкретних підсистемах, в тому числі реалізованим програмно. Приклади легко зустріти в таких системах, як супутники, системи оборонного призначення, системи національної безпеки та енергетичні системи.

**Прикладне ПЗ.** Ця категорія програмного забезпечення надає служби (послуги) для задоволення конкретної потреби і розглядається як «автономне». Зазвичай додатки взаємодіють з системним і вбудованим ПО, будучи споживачами їхніх послуг. В якості прикладів можна назвати популярні офісні пакети програм - текстові процесори, електронні таблиці і програми для підготовки презентацій.

Як описано в попередніх розділах, будь-який проект розробки проходить ряд етапів від ідеї до завершення. Концепція моделі життєвого циклу - це цінний управлінський інструмент для планування робіт, укомплектування персоналом, організації робіт та управління ресурсами, складання календарних графіків та інших допоміжних дій, необхідних для успішного виконання проекту. Вона важлива також для визначення контрольних точок і точок прийняття рішень, які дозволяють стежити за дотриманням термінів і бюджету.

**Основні етапи розробки.** Ми бачили, що метод системної інженерії включає чотири кроки.

1. Аналіз вимог.
2. Функціональне опис.
3. Опис фізичної реалізації.
4. Валідація проектних рішень.

Процес розробки ПЗ також можна звести до чотирьох основних кроків:

1. Аналіз.
2. Проектування, яке включає архітектурне проектування, проектування процедур і т. д.
3. Кодування та автономне тестування, звані також реалізацією.
4. Інтеграційний і комплексне тестування.

Хоча ці кроки не є повним аналогом кроків методу системної інженерії, цілі відповідних кроків досить близькі.

Слід зазначити, що, як і у випадку методу системної інженерії, в різних варіантах опису процесу розробки ПЗ застосовується різна термінологія в плані найменування кроків або етапів, а іноді деякі основні кроки розбиваються на декілька. Наприклад, проектування може бути розділене на попереднє та детальне; автономне тестування іноді об'єднують з кодуванням, а буває, що його виділяють в окремий крок. Системне тестування іноді називають інтеграцією і тестуванням. Необхідно пам'ятати, що розбиття на кроки - це модель процесу, і тому можливі різні варіації і інтерпретації.

Як і у випадку з моделями життєвого циклу системи, в різних моделях процесу розробки ПЗ зустрічаються одні й ті ж базові функції, а відмінності стосуються переважно способу виконання кроків, послідовності дій і іноді форми їх подання. У загальному і цілому моделі розробки ПЗ можна віднести до однієї з чотирьох категорій:

**1. Лінійні моделі.** Як і в формальних моделях життєвого циклу розробки системи, лінійна модель розробки ПЗ складається з послідовності кроків, як правило, зі зворотним зв'язком, результатом яких є програмний продукт. Лінійні моделі розробки добре працюють, коли вимоги повністю зрозумілі і стабільні, план-графік не надто напружений, ресурсів достатньо, а методики добре задокументовані.

**2. Інкрементні моделі.** У інкрементній моделі основні кроки такі ж, як і в лінійній, але процес ітеративно повторюється. При цьому ступінь деталізації кожного кроку на різних ітераціях може відрізнятись. За такої моделі розробки існує кілька моментів часу, коли функціональні можливості реалізовані частково. Подібна модель добре пристосована до ситуації, коли вимоги стабільні і бажано отримати частину функціональних можливостей ще до того, як створення системи буде повністю завершено.

**3. Еволюційні моделі.** Еволюційні моделі схожі на інкрементні, але призначені для випадків, коли остаточні характеристики і властивості продукту невідомі на початку процесу розробки. Такі моделі дозволяють отримати обмежені функціональні можливості на основі рішень не призначених для виробництва (наприклад, прототипів), але придатних для експериментів, демонстрації та ознайомлення. Для еволюційних моделей вкрай важлива наявність зворотного зв'язку, оскільки в результаті трьох вищезазначених процедур система «еволюціонує», наближаючись до задоволення потреб користувачів.

**4. Гнучкі моделі.** Гнучкі моделі розробки відходять від класичної схеми з чотирма кроками. Якщо в лінійній, інкрементній та еволюційній моделі ці кроки виконуються в різній послідовності і по-різному повторюються, то в гнучкій вони якимось чином комбінуються, і межі між ними розмиваються. Гнучкі методики гарні для ситуацій, коли ні структура, ні опис заздалегідь невідомі, а зміни відбуваються протягом усього процесу.

**Схема функціональних потоків.** Дані моделі активно використовуються у FFBD моделях, які бувають декількох видів. FFBD подібна до схеми функціональних блоків, але замість того, щоб, зображати функціональні інтерфейси як на FBD, на схемі FFBD з'єднання (вони зображаються у вигляді стрілок), являють потік управління. Оскільки FFBD включає встановлення послідовності (що не дозволяють зробити ні схеми функціональних блоків (FBD), ні мова IDEF0), логічні точки розриву зображуються сумованими вентилями (summing gate). Такі конструкції дають можливість зобразити концепції, орієнтовані на процеси. За допомогою FFBD можна змоделювати майже будь-який процес. На рис. 11.11 наведено приклад FFBD.

Як і у випадку будь-якої функціональної діаграми, кожен функцію в ієрархії можна розкласти на підфункції і для кожного рівня розробити відповідну діаграму. Функціональні діаграми - стандартний для структурного аналізу спосіб відображення поведінки і функціональних можливостей системи.

**Діаграма потоків даних.** Ця діаграма складається в основному з «бульбашок» (кіл або еліпсів), які представляють собою функціональні елементи, з'єднані лініями, які забезпечені примітками з описами даних,

переданих між цими елементами. Сховища даних представлені двома паралельними лініями, а зовнішні об'єкти - прямокутниками. Зазвичай для представлення системи потрібні DFD декількох рівнів, починаючи з контекстної діаграми, на якій є всього один «бульбашка» - система, оточена зовнішніми об'єктами, зображуваними у вигляді прямокутників. На наступних рівнях декомпозиції для кожного з «бульбашок» з попереднього рівня показані відповідні йому потоки даних. З точки зору системного інженера, DFD аналогічна схемі функціональних блоків, але без потоку управління.

Програмне забезпечення може бути представлено у вигляді елементів, званих комп'ютерними програмами, а кожна програма складається з набору команд.

**Складові частини програми.** Можна вважати, що комп'ютерна програма включає складові частини декількох типів. Нижче ці частини перераховані в порядку убування розміру, і для кожної зазначено загальноприйнята назва.

1. «Модуль», або «пакет», - це велика структурна одиниця програми, що виконує одну або кілька операцій. До складу програми середнього і великого розміру зазвичай входить від кількох до десятків і сотень модулів.

2. В об'єктно-орієнтованих програмах класом називається структурний елемент, що складається з «атрибутів» (елементи даних) та асоційованих з ними «методів» або «служб» (функції). Об'єкт - це екземпляр класу.

3. Функція - набір команд для виконання операцій над даними та управління потоком обробки. «Утилітою», або «бібліотечної функцією», називається часто вживана функція (наприклад, тригонометрична), надана операційною системою.

4. «Управляюча конструкція» визначає порядок виконання команд. Наведемо чотири приклади керуючих конструкцій:

а) Послідовність: впорядкована послідовність команд.

б) Галуження за умовою: if (умова) then (операція 1) else (операція 2).

в) Цикл: do while (умова) або do until (умова).

г) багатокількі розгалуження: case (ключ 1): (операція 1) ... (ключ і): (операція п).

5. «Команда» - «декларативне» або «який виконувався» припис комп'ютера, складене з ключових слів, символів та імен даних і функцій.

6. Ключове слово, символ або ім'я елемента даних небудь функції.

Нарешті, «структура даних» - це визначення набору взаємопов'язаних елементів даних, наприклад «запис», «масив» або «зв'язаний список».

**Мова проектування програм.** Для подання проектів ПЗ, отриманих в результаті використання традиційної методології структурного аналізу і проектування, корисний мову проектування програм (Program Design Language - PDL), який іноді називають «структурованим англійським». Він складається з високорівневих команд, пов'язаних керуючими конструкціями, як у справжній програмі, але замість ключових слів і речень мови програмування використовуються звичайні текстові фрази. Лістинг на PDL зрозумілий будь-



якому програмному інженеру і більш-менш просто транслюється у вихідний код на мові програмування.

**Подання ТОВ.** Ми бачили, що в процесі OOD породжуються діаграми і описові матеріали, в тому числі визначення об'єктів, що становлять складові частини проміжної програми. За допомогою інструменту, що підтримує UML, проектну інформацію можна автоматично перетворити в архітектуру комп'ютерної програми.

### Мови програмування

Вибір мови програмування - одне з найважливіших рішень при проектуванні ПЗ. Тут дуже важливий тип системи - вбудована, програмно насичена або обчислювально-орієнтована, призначена вона для комерційного або військового застосування, інтерактивна вона чи реального часу. Які б не були переваги проектувальників ПО, природа додатки повинна стояти на першому місці. Від мови можуть залежати придатність програмного продукту для супроводу, переносимість, зручне читання і безліч інших характеристик.

Якщо не вважати окремих спеціалізованих застосувань, комп'ютерні програми пишуться на мовах високого рівня, в яких одна команда зазвичай транслюється в кілька елементарних машинних операцій. У табл. 11.1 наведено кілька прикладів застарілих і сучасних мов програмування, їх структурні складові, основні сфери застосування та загальний опис.

Таблиця 11.1. Поширені мови програмування

Мова	Структурні складові	Основні сфери застосування	Опис
Ada95	<ul style="list-style-type: none"> <li>• Об'єкти</li> <li>• Функції</li> <li>• Завдання</li> <li>• Пакети</li> </ul>	<ul style="list-style-type: none"> <li>• Військові системи</li> <li>• Системи реального часу</li> </ul>	Проектувався спеціально для застосування у вбудованих військових системах, де в чому замінив C ++
C	<ul style="list-style-type: none"> <li>• Функції</li> </ul>	<ul style="list-style-type: none"> <li>• Операційні системи з обладнанням</li> <li>• Додатки реального часу</li> <li>• Програми загального призначення</li> </ul>	Потужна мова загального призначення, що володіє дуже високою гнучкістю
C ++	<ul style="list-style-type: none"> <li>• Об'єкти</li> <li>• Функції</li> </ul>	<ul style="list-style-type: none"> <li>• Імітаційне моделювання</li> <li>• Додатки реального часу</li> <li>• Інтерфейси з обладнанням</li> <li>• Програми загального призначення</li> </ul>	Потужна мова загального призначення, в якій організовані об'єктно-орієнтовані конструкції

COBOL	<ul style="list-style-type: none"> <li>• Підпрограми</li> </ul>	<ul style="list-style-type: none"> <li>• Ділові та фінансові додатки</li> </ul>	Словесна мова, в якійсь мірі самодокументована. Колись була основною мовою програмування ділових додатків
FORTRAN	<ul style="list-style-type: none"> <li>• Підпрограми</li> <li>• Функції</li> </ul>	<ul style="list-style-type: none"> <li>• Наукові додатки</li> <li>• Аналіз даних</li> <li>• Імітаційне моделювання</li> <li>• Програми загального призначення</li> </ul>	Давно створена мова загального призначення, застосовується головним чином в програмах, призначених для обчислень
Java	<ul style="list-style-type: none"> <li>• Об'єкти</li> <li>• Функції</li> </ul>	<ul style="list-style-type: none"> <li>• Внутрішні додатки</li> <li>• Програми загального призначення</li> </ul>	Похідна від C ++ інтерпретована платформо незалежна мова
Visual Basic	<ul style="list-style-type: none"> <li>• Об'єкти</li> <li>• Підпрограми</li> </ul>	<ul style="list-style-type: none"> <li>• Графічні додатки</li> <li>• Користувальницькі інтерфейси</li> </ul>	Мова, що дозволяє графічно маніпулювати програмними об'єктами
Мова асемблера	<ul style="list-style-type: none"> <li>• Підпрограми</li> <li>• Макрокоманди</li> </ul>	<ul style="list-style-type: none"> <li>• Управління устаткуванням</li> <li>• Драйвери</li> </ul>	Мова для операцій самогонизького рівня; забезпечує повний контроль над машиною

**Мови четвертого покоління і спеціалізовані мови.** Мови четвертого покоління (4 GL) зазвичай є власністю правовласників і надають високорівневі методи для вирішення задач в деякій предметній області. Найчастіше такі мови поставляються разом з системою баз даних і так чи інакше пов'язані зі структурованою мовою запитів (SQL). Ключова риса 4 GL полягає в тому, щоб максимально наблизити середовище мови програмування до природної мови предметної області та надати інтерактивні інструменти для опису рішення. Наприклад, розглянемо процедуру інтерактивного створення форми для введення даних на робочій станції. Програміст вводить назви полів, описує допустимі значення та інші обмеження, після чого «екран» стає частиною програми. Мови четвертого покоління можуть прискорити розробку додатків певного виду, але, взагалі кажучи, не стерпні, тобто працюють тільки з продуктами одного постачальника.

Існує безліч областей, для яких були розроблені спеціалізовані дуже ефективні мови. У таких мовах зазвичай застосовуються лексика і конструкції з області, для якої вони призначені. Завдання спеціалізованої мови - імітувати, наскільки можливо, предметну область і зменшити час розробки, підвищивши в той же час надійність. У багатьох випадках продуктивність приноситься в жертву простоті використання і розробки. Приступаючи до розробки замовного ПЗ, системний інженер повинен вивчити, які мови є для даної предметної області і наскільки вони корисні.

Етап технічного проектування - це частина процесу розробки нової системи, коли відбувається проектування всіх її деталей, так щоб вони поєднувалися один з одним, утворюючи єдине ціле, що працює у відповідності

з вимогами призначення. Це напружена і добре організована робота, спрямована на створення компонентів, які характеризуються надійністю, ремонтпридатністю і безпекою при всіх умовах, в яких може опинитися система, а також придатністю до виробництва у встановлені строки та в межах певних витрат. Загальний підхід до проектування, що дозволяє досягти вищезазначених цілей, повинен був бути визначений на попередніх етапах, а на етапі технічного проектування опрацьовуються деталі зовнішніх і внутрішніх інтерфейсів, і проект вперше повністю реалізується у вигляді апаратних і програмних елементів системи.

Метод системної інженерії на етапі технічного проектування включає основні дії в рамках кожного з чотирьох кроків методу системної інженерії.

**1. Аналіз вимог.** Типові дії включають:

- аналіз повноти і несуперечності технічних вимог до системи;
- виявлення вимог до зовнішніх і внутрішніх взаємодій і інтерфейсам.

**2. Аналіз функціонування та проектування (функціональний опис).**

Типові дії включають:

- аналіз взаємодії між компонентами і інтерфейсів між ними, а також виявлення проблем, пов'язаних з проектними рішеннями, комплексуванням і випробуванням;
- детальний аналіз способів взаємодії з користувачами;
- проектування та створення прототипів користувальницьких інтерфейсів.

**3. Проектування компонентів (опис фізичної реалізації).** Типові дії

включають:

- розроблення попередніх проектів всіх апаратних і програмних компонентів і інтерфейсів;
- реалізацію детальних проектів обладнання та програмного коду після експертного рецензування;
- побудова дослідних зразків розроблених компонентів.

**4. Валідація проектних рішень.** Типові дії включають:

- випробування та атестацію спроектованих компонентів в плані функціонування, інтерфейсів, надійності і можливості виробництва;
- виправлення недоліків;
- документування проекту виробу.

## **КОРОТКІ ВИСНОВКИ**

Терміни «програмна інженерія» та «інженерія програмних систем» - не синоніми. Перший відноситься до розробки програмних продуктів, автономних або вбудованих, а другий – до використання визначених принципів до дисципліни програмної інженерії. Ми вважаємо, що програмне забезпечення складається із трьох основних компонентів: 1) команд, які називаються також кодом; 2) структур даних і 3) документації.

**Подолання важкості та абстрактності.**

За останні 20 років роль ПЗ змінилось – в більшості сучасних систем вона є пріоритетною. Тому програмна інженерія стала повноправною частиною розробки систем.

## **Природа розробки програмного забезпечення.**

В програмному забезпеченні можна виділити наступні критерії:

1. Системне ПЗ, поставляючи служби іншим програмам.
2. Вмонтоване ПЗ – служби, функції та компоненти більш масштабної системи.
3. Прикладне ПЗ – автономна система, постачальник служби для задоволення конкретної потреби.

Системи, в яких використовується ПЗ, можна розбити на три категорії:

1. Вмонтовані програмні системи що надають комбінації обладнання та програмного забезпечення. В основному вони складаються з обладнання, а на ПЗ покладають функції управління роботою апаратних компонентів. Прикладами можуть служити транспортні засоби, космічні апарати, робота технічні та воєнні системи.

2. Програмно насичені системи складаються з комп'ютерів та сіток, котрі управляються програмами. ПЗ в них реалізує практично всі функціональні можливості в системі, в тому числі всі складні функції обробки інформації.

В якості прикладів можна навести системи резервування авіаквитків, системи управління фінансовою діяльністю і системи управління запасами.

3. Обчислювально-орієнтовані системи - це великомасштабні комп'ютерні обчислювальні ресурси для вирішення складних обчислювальних завдань. До прикладів можна віднести гідрометеорологічні центри, системи прогнозування вражаючого дії ядерної зброї, вдосконалені системи дешифрування інформації та інші операції, що передбачають великий обсяг обчислень.

### **Моделі життєвих циклів розробки ПЗ**

В загальних рисах життєвий цикл переважно програмних систем схожий на життєвий цикл системної інженерії. Хоча існує чимало моделей життєвого циклу, можна визначити чотири основних типи:

1. Лінійні моделі - послідовність кроків, зазвичай зі зворотним зв'язком.

2. Додаткове моделі - повторення послідовності кроків для поступового нарощування можливостей і функціональності; останнє розширення включає всі можливості.

3. Еволюційні моделі - схожі на інкрементні, але ранні версії призначені для експериментів, аналізу, ознайомлення і демонстрації. Наступні версії великою мірою спираються на досвід, отриманий при роботі з ранніми версіями.

4. Гнучкі моделі - типові кроки розробки ПО комбінуються різними способами для швидкого отримання надійного продукту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Теорія автоматичного керування. Навчальний посібник Авт. Сорока К.О. – Харків, ХНАМГ, 2006 – 187 с.
2. Попович М.Г., Ковальчук О.В. Теорія автоматичного керування. – К.: Либідь, 1997. – 544 с.
3. Фельдбаум А.И. Методы теории автоматического управления. – М.: Наука, 1971- 744с
4. Куропаткин П.В. Теория автоматического управления . – М.: Высш. шк., 1973 – 528 с.
5. Айзерман М.А. Теория автоматического управления. М.: Наука, 1966.
6. Нетушил А.В. Теория автоматического управления. Ч.1. Линейные системы
7. Воронов А.А. (под ред.). Теория автоматического управления. Ч.1 – М.: Выс. шк. 1986. – 367 с.
8. Юревич Е.И. Теория автоматического управления. – Л.: Энергия, 1975. – 416 с.
9. Бесекерский В.А., Попов Е.П. Теория систем автоматического регулирования. – М.: Физматгиз, 1975. – 768 с.
10. Сорока К.О. Методичні вказівки до лабораторних робіт з предмету "Теорія автоматичного керування" – Харків, 2001, 55 с.
11. Сорока К.О., Андрійченко В.П. Методичні вказівки до самостійного вивчення дисципліни та виконання контрольних робіт з курсу "Теорія автоматичного керування". – Харків, ХНАМГ, 2004, – 40 с.
12. Blanchard B., Fabrycky W. System Engineering and Analysis, Fourth Edition. Chapters 6,12,13. Prentice Hall, 2006.
13. Chase W. P. Management of Systems Engineering. Chapter 6. John Wiley & Sons, Inc., 1974.
14. Hitchins DK Systems Engineering: A 21st Century Systems Methodology. Chapters 8, 11, 12. John Wiley & Sons, Inc., 2007.
15. International Council on Systems Engineering. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. INCOSE-TP-2003-002-03.2. Section 4. July, 2010.
16. O'Connor PDT TestEngineering: A Concise Guide to Cost-effective Design, Development and Manufacture. Chapters 6-8,10. JohnWiley&Sons, Inc., 2005.
17. Petroski H. SuccessthroughFailure: TheParadoxofDesign. Princeton University, 2006.
18. Rechtin E. SystemsArchitecting: CreatingandBuildingComplexSystems. Chapter 7. Prentice Hall, 1991.
19. Reynolds MT Testand Evaluationof Complex Systems. John Wiley&Sons, Inc., 1996.
20. Shinnners SM A Guide for Systems Engineeringand Management. Chapter 7. Lexington Books, 1989.
21. Stevens R., Brook R, Jackson K., Arnold S. Systems Engineering Copingwith Complexity. Chapter5. Prentice Hall, 1998.
22. Попович М.Г., Борисюк М.Г., Гаврилюк В.А та ін. Теорія електропривода. – К.: Вища шк., 1993, – 494 с.

23. Baker L., Clemente P., Cohen B., Permenter L., Purves B., Salmon P. Foundational Concepts for Model Driven System Design. INCOSE Model Driven Design Interest Group, INCOSE, July 2000. Balmelli L., Brown D., Cantor M., Mott M. Model-driven systems development. IBM Systems Journal, 2006, 45 (3), pp. 569-585.
24. Blanchard B., Fabrycky W. System Engineering and Analysis, Fourth Edition. Chapter 3. Prentice Hall, 2006.
25. Brooks F. P. The Mythical Man Month - Essays on Software Engineering. Addison-Wesley, 1995. Chase W. P. Management of Systems Engineering. Chapters 3, 4. John Wiley, 1974.
26. Dam S. DOD Architecture Framework: A Guide to Applying System Engineering to Develop Integrated, Executable Architectures. SPEC, 2006.
27. Defense Acquisition University. Risk Management Guide for DoD Acquisition, Sixth Edition. DAU Press, 2006.
28. Department of Defense Web site. DoD Architecture Framework Version 2.02. <http://cio-nii.defense.gov/sites/dodaf20>.
29. Eisner H. Computer-Aided Systems Engineering. Chapter 12. Prentice Hall, 1988.
30. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition. Addison-Wesley, 2004.
31. Hoffmann H. SysML-based systems engineering using a model-driven development approach. Telelogic White Paper, Version 1, January 2008.
32. International Council on Systems Engineering. Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities. Version 3.2, July 2010.
33. Kasser J. A Framework for Understanding Systems Engineering. The Right Requirement, 2007. Maier M., M.Rechtin M. The Art of Systems Architecting. CRC Press, 2009.
34. Pressman R. S. Software Engineering: A Practitioner 's Approach. McGraw Hill, 2001.
35. Systems Engineering Fundamentals. Chapter 7. Defense Acquisition University (DAU Press), 2001

# **ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ**

**з дисципліни**

## **«ОСНОВИ СИСТЕМНОЇ ІНЖЕНЕРІЇ»**

для студентів освітньо-кваліфікаційного рівня «бакалавр»  
напряму підготовки: 6050201 – «Системна інженерія»

Підписано до друку 28.10.2015 р.  
Формат 84x108\32. Папір офсетний Друк на різнографі.  
Умов.-друк.арк. 2,1. Обл.-вид. арк. 6,7. Зам. №15.  
Тираж 50 прим.

Віддруковано у ТзОВ «Гал-друк»  
м.Тернопіль, вул. Бродівська, 44  
тел./факс 520563

