

Лекція 1

Програмна інженерія як предмет вивчення

Анотація: Поняття програмної інженерії. SWEBOOK. Основні визначення: інформатика, системотехніка, бізнес-реінжиніринг. Програмне забезпечення: визначення, властивості.

Програмна інженерія

Чим програмування відрізняється від програмної інженерії? Тим, що перше є деякою абстрактною діяльністю і може відбуватися в багатьох різних контекстах. Можна програмувати для задоволення, для того, щоб навчитися (наприклад, на практичних заняттях, на семінарах в університеті), можна програмувати в межах наукових розробок. А можна займатися промисловим програмуванням. Як правило, це відбувається в команді, і абсолютно точно – для замовника, який платить за роботу гроші. При цьому необхідно точно розуміти, що потрібне замовникові, виконати роботу в певні терміни і результат повинен бути потрібної якості – такий, який задовольнить замовника і за який він заплатить. Щоб задовольнити цим додатковим вимогам, програмування набуває різних додаткових видів діяльності: розробкою вимог, плануванням, тестуванням, конфігураційним управлінням, проектним менеджментом, створенням різної документації (проектною, призначеною для користувача і ін.).

Розробці програмного коду передуює аналіз і проектування (перше означає створення функціональної моделі майбутньої системи без урахування реалізації, для усвідомлення програмістами вимог і очікувань замовника; друге означає попередній *макет*, ескіз, план системи на папері). Трудовитрати на *аналіз* і проектування, а також форма представлення їх результатів сильно варіюються від видів проектів і вподобань розробників і замовників.

Потрібні також спеціальні зусилля для організації процесу розробки. У загальному вигляді це ітеративно-інкрементальна модель, за якою необхідна функціональність створюється порціями, яку менеджери і замовник можуть оцінити, і тим самим, є можливість управління перебігом розробки. Проте, ця загальна модель має безліч модифікацій і варіантів.

Розробку системи також необхідно виконувати з урахуванням зручностей її подальшого супроводу, повторного використання і інтеграції з іншими системами. Це означає, що система розбивається на компоненти, які є зручними в розробці, придатними для повторного використання і інтеграції та мають необхідні характеристики,

наприклад, за швидкодією. Для цих *компонент ретельно* опрацьовуються інтерфейси. Сама ж система документується на багатьох рівнях, створюються правила оформлення програмного коду – тобто залишаються численні семантичні сліди, що допомагають створити, зберегти і підтримувати єдину, струнку архітектуру, одноманітний стиль, порядок.

Всі ці та інші додаткові види діяльності, що виконуються в процесі промислового програмування і є необхідними для успішного виконання замовлень, і називатимемо **програмною інженерією** (software engineering). Так ми позначаємо, по-перше, деяку практичну *діяльність*, а по-друге, спеціальну **галузь знання**, або іншими словами, наукову дисципліну. Адже для полегшення виконання кожного окремого проекту, для можливості використовувати різноманітний позитивний *досвід*, що був досягнутий іншими командами і розробниками, цей самий *досвід* піддається осмисленню, узагальненню і належному оформленню. Так з'являються різні методи і практики (*best practices*), саме: тестування, проектування, робота над вимогами, архітектурні шаблони та ін. Також з'являються стандарти і методології, що стосуються всього процесу в цілому (наприклад, *MSF, RUP, CMMI, Scrum*). Ось ці-то узагальнення і входять в програмну інженерію, як у галузь знання.

Ще можна визначити, що **Програмна інженерія (Software Engineering)** є галуззю інформатики, яка вивчає питання побудови комп'ютерних програм, відображає закономірності розвитку програмування, узагальнює досвід програмування у вигляді комплексу знань і правил регламентації інженерної діяльності розробників ПЗ.

У цьому визначенні виділимо два основних аспекти.

Програмну інженерію можна розглядати як інженерну дисципліну, в якій інженери застосовують теоретичні ідеї, методи і засоби при розробці ПЗ, створюють продукти відповідно до стандартів, що регламентують процеси їх проектування і розробки.

Програмна інженерія описує методи управління програмним проектом, якістю та ризиками. Застосування таких методів дозволяє досягти високої якості програмних продуктів. Ця інженерна дисципліна надає всю необхідну інформацію і стандарти для вибору найбільш підходящого методу і процесів життєвого циклу ПЗ для реалізації конкретного проекту.

Необхідність в програмній інженерії, як у спеціальній галузі знань, була усвідомлена світовою спільнотою в кінці 60-х років минулого століття, більш ніж на 20 років пізніше за народження самого

програмування, якщо вважати таким фактом появу відомого звіту фон Неймана "*First Draft of a Report on the EDVAC*" («Перший проект звіту на EDVAC»), що був надрукований ним в 1945 році. Народженням програмної інженерії є 1968 рік – конференція *NATO Software Engineering*, м. Гарміш (ФРН), яка цілком була присвячена розгляду цих питань.

У конференції взяли участь понад п'ятдесят фахівців з одинадцяти різних країн, які професійно займалися програмним забезпеченням: як користувачі і розробники, так і викладачі з університетів. Обговорення охоплювали усі аспекти програмного забезпечення, включаючи:

- відношення програмного забезпечення до апаратних засобів та комп'ютерів;
- проектування програмного забезпечення;
- розробку або впровадження програмного забезпечення;
- поширення програмного забезпечення;
- послуги з підтримки програмного забезпечення.

Включаючи безліч прямих цитат та обмін думками, звіт відображає жваві суперечки протягом початкової дискусії. Незважаючи на те, що велика частина обговорень мала суто технічний характер, звіт також містить розділи, які стосуються обговорення, що можуть зацікавити набагато ширшу аудиторію. Це стосується таких предметів, як:

- проблеми досягнення достатньої надійності в системах даних, які дедалі більше інтегруються в центральну діяльність сучасного суспільства
- труднощі дотримання графіків та специфікацій щодо великих програмних проектів
- освіта інженерів програмного забезпечення (або систем обробки даних)
- надзвичайно суперечливе питання, чи варто встановлювати ціну на програмне забезпечення окремо від обладнання.

У сферу програмної інженерії потрапляють всі питання і теми, які пов'язані з організацією і поліпшенням процесу розробки **ПЗ**, управлінням колективом розробників, розробкою і впровадженням програмних засобів підтримки життєвого *циклу розробки ПЗ*.

Новий програмний проект розробляється 1-2 роки, а еволюціонує 6-7 років. На супровід проекту витрачається 61% проти 39% коштів на його розробку. Ефективність розробників в залежності від кваліфікації коливається в співвідношенні 1:10, а значить, потрібно підвищувати рівень знань розробників ПЗ. На сьогодні ядро стабільних знань з програмної інженерії становить 75% від тих знань, якими користуються

в практичній діяльності. У зв'язку з цим проведена систематизація накопичених знань в програмуванні та ряді інших областей інформатики.

Міжнародним комітетом при американському об'єднанні комп'ютерних фахівців ACM (Association for Computing Machinery) і інституті інженерів з електроніки та електротехніки IEEE Computer Society було створено ядро знань SWEBOOK. У цьому ядрі були систематизовані різноманітні знання в області програмування, планування і управління, сформульовано поняття програмної інженерії та десяти областей, які відповідають процесам проектування ПЗ і методам їх підтримки.

Як інженерна дисципліна, вона охоплює всі аспекти створення ПЗ, починаючи від формування вимог до створення, супроводу і зняття з експлуатації ПЗ, а також включає інженерні методи оцінки трудовитрат, вартості, продуктивності і якості. Тобто мова йде саме про інженерну діяльність в програмуванні, оскільки її сутність близька до визначення інженерної діяльності в тлумачному словнику:

- **інженерія** - це спосіб застосування наукових результатів, що дозволяє отримувати користь від властивостей матеріалів і джерел енергії;
- **інженерія** - діяльність по створенню машин для надання корисних для споживача послуг і виробів.

Інженери з програмної інженерії - це фахівці, які виконують практичні роботи з реалізації програм із застосуванням теорії, методів і засобів комп'ютерної науки. Комп'ютерна наука охоплює теорію і методи побудови обчислювальних і програмних систем, тоді як програмна інженерія розглядає питання практичної побудови ПЗ. Знання комп'ютерної науки необхідні фахівцям в області програмного забезпечення так само, як знання фізики - інженерам-електронікам. Якщо для вирішення конкретних завдань програмування не існує відповідних методів або теорії, інженери застосовують свої знання, які накопичені ними в процесі розробок конкретного ПЗ, а також використовують досвід застосування відповідних інструментальних програмних засобів.

Інженери, як правило, працюють за умов до укладених контрактів і виконують завдання проекту з урахуванням цих умов і обмежень на терміни, час, вартість і ін. **На відміну від науки, мета якої - отримання знань, для інженерії знання – це спосіб отримання певного зиску.**

В програмної інженерії працюють:

- **програмісти**, які займаються безпосередньо розробкою ПЗ;

- **менеджери**, які планують і керують проектом, відстежують терміни і витрати;
- **архітектори** – це спеціалісти, які на основі бізнес-вимог формують стиль розбиття системи на складові підсистеми та інтерфейси програмної взаємодії складових програмного продукту;
- **проектувальники** – це спеціалісти, які виконують розробку проекту реалізації розробки програмного проекту за складовими у часі та планують всі етапи розробки проекту;
- **інженери служби ведення бібліотек і репозитаріїв компонентів**;
- **технологи** – це спеціалісти, які визначають інженерні методи і стандарти, створюють для проекту модель ЖЦ, що задовольняє його цілям і завданням;
- **тестувальники** (контролери), які перевіряють правильність виконання процесу проектування шляхом тестування і на основі зібраних даних проводять вимірювання різних характеристик якості, включаючи оцінку надійності ПЗ;
- **верифікатори** – це спеціалісти, які перевіряють правильність реалізації функцій в проекті;
- **валідатори** – це ті, хто перевіряє ПЗ на відповідність заданим вимогам.

Розробку програмних систем можна вважати інженерної діяльністю, але вона має деякі відмінності від традиційної інженерії:

- гілки інженерії мають високу ступінь спеціалізації, а в програмній інженерії спеціалізація торкнулася лише окремих областей (наприклад, операційні системи, транслятори, редактори і т.п.);
- програмування об'єктів ґрунтується на стандартах, за допомогою яких відображаються типові вимоги замовників, тобто типізація об'єктів і артефактів в сфері програмування;
- технічні рішення класифіковані і каталогізовані, а в програмній інженерії кожна нова розробка - це нова проблема, для реалізації якої встановлюють аналогію з раніше розробленими системами. Одним з інженерних рішень каталогізації в програмуванні є патерн проектування.

Для перетворення програмної інженерії в спеціальність світова комп'ютерна громадськість створила професійні комітети, які регламентують аспекти процесу програмування: ядро знань SWEBOOK, етичний кодекс програміста [1.13], навчальні курси (Curricula -2001, 2004) з підготовки фахівців в області програмної інженерії, навчання спеціальності і сертифікація фахівців.

Таким чином, виникнення програмної інженерії як дисципліни розробки ПЗ визначено такими важливими факторами:

- значним обсягом накопичених знань в області створення ПЗ;
- появою нових методів аналізу, моделювання та проектування ПЗ;
- вдосконаленням методів виявлення помилок в ПЗ;
- ефективною організацією колективів розробників ПЗ і оцінки їх трудової діяльності;
- використанням готових програмних компонентів, високотехнологічних засобів і інструментів розробки ПЗ;
- необхідністю еволюційного розвитку компонентів і систем, а також їх адаптацією до нових умов, що змінюються операційному середовищі та комп'ютерних мереж.

Програмна інженерія робить акцент на підвищенні якості та продуктивності програмного забезпечення за рахунок застосування нових і вдосконалених методів проектування ПЗ; готових компонентів і методів їх генерації; методів еволюції, верифікації та тестування ПЗ; інструментальних засобів; методів управління проектами, оцінки якості та вартості.

Програмна інженерія використовує досягнення інформатики, тісно пов'язана з системотехнікою, часто передує бізнес-реінжинірингу. Трохи докладніше про цей контекст програмної інженерії.

Інформатика (*computer science*) – це комплекс теоретичних наук, які засновані на математиці і присвячені формальним основам обчислювальності. Сюди відносять математичну логіку, теорію графіків, методи побудови компіляторів, математичні формальні методи, що використовуються у верифікації і модельному тестуванні і так далі. Важко точно відокремити програмну інженерію від інформатики, але в цілому спрямованість цих дисциплін різна. *Програмна інженерія* націлена на вирішення проблем виробництва, *інформатика* – на розробку формальних, математизованих підходів до програмування.

Системотехніка (*system engineering*) об'єднує різні інженерні дисципліни із розробки всіляких штучних систем – енергоустановок, телекомунікаційних систем, вбудованих систем реального часу і так далі. Дуже часто ПЗ виявляється складовою подібних систем, що виконують завдання управління відповідним устаткуванням. Такі системи називаються *програмно-апаратними*, і беручи участь в їх створенні, програмісти вимушені глибоко розбиратися в особливостях відповідної апаратури.

Бізнес-реінжиніринг (*business reengineering*) – в широкому сенсі позначає модернізацію бізнесу в певній компанії, впровадження нових

практик, що підтримуються відповідними новими інформаційними системами. При цьому акцент може бути як на внутрішній реорганізації компанії, так і на розробці нового клієнтського сервісу (як правило, ці питання взаємозв'язані). Бізнес-реінжиніринг часто передує розробці і впровадженню інформаційних систем на підприємстві, оскільки потрібно спочатку навести певний порядок в діловодстві, а лише потім закріпити його інформаційною системою.

Зв'язок програмної інженерії (як галузі практичної діяльності) з інформатикою, системотехнікою і бізнес-реінжинірингом показаний на Рис. 1.1.

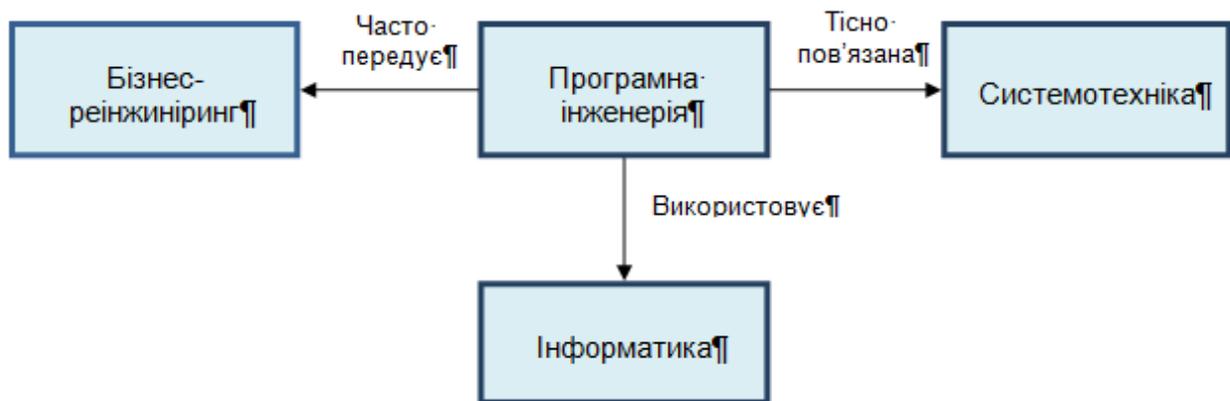


Рис. 1.1. Зв'язок програмної інженерії (як галузі практичної діяльності) з іншими галузями знань

В редакції SWEBOOK 2004 року визначаються десять областей знань в контексті програмної інженерії, які поділено на 2 групи.

Основні області (Рис. 1.2):

- Вимоги до ПЗ (Software requirements).
- Проектування (Software design).
- Конструювання (Software construction).
- Тестування (Software testing).
- Супроводження (Software maintenance).

Організаційні області (Рис. 1.3):

- Управління конфігурацією (Software configuration management).
- Управління проектами (Software engineering management).
- Процеси програмної інженерії (Software engineering process).
- Засоби та інструменти (Software engineering tools and methods).
- Якість ПЗ (Software quality).

Наукоємні області (Рис. 1.4):

- Професійні навички з програмної інженерії (Software Engineering Professional Practices)
- Економіка інженерії програмного забезпечення (Software Engineering Economics)
- Обчислювальні основи (Computing Foundations)
- Математичні основи (Mathematical Foundations)
- Основи інженерії (Engineering Foundations)

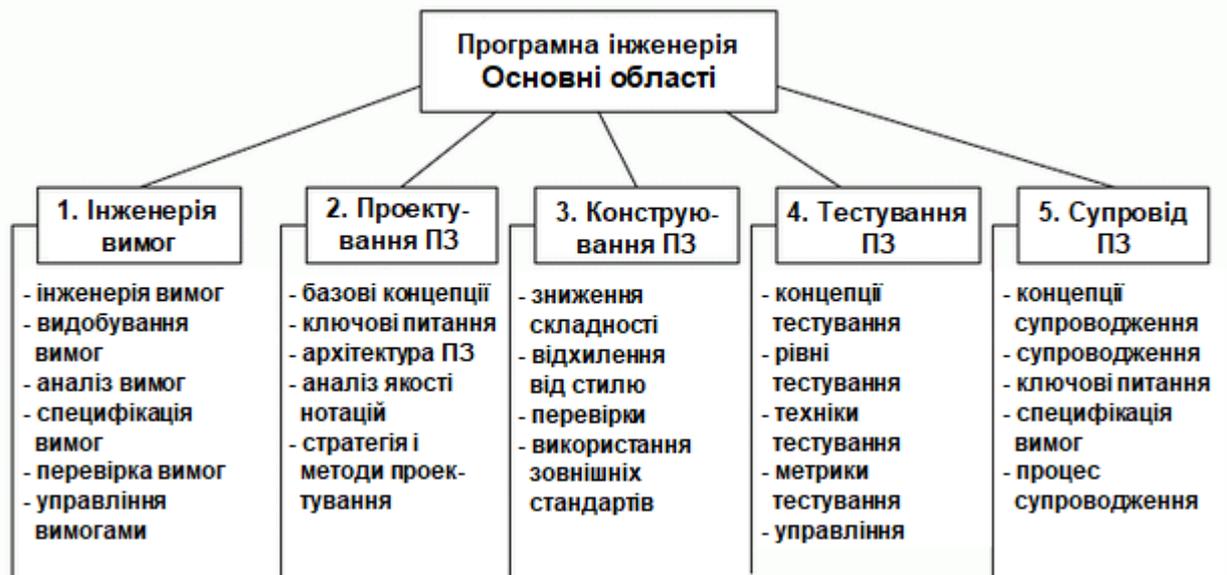


Рис. 1.2. Основні області програмної інженерії



Рис. 1.3. Організаційні області програмної інженерії



Рис. 1.4. Наукоємні області програмної інженерії

У редакції 2014 року кількість областей знань SWEBOK збільшилась до 15 (Табл. 1.1)

Таблица 1.1

Області знань програмної інженерії за SWEBOK

15 галузей знань SWEBOK	10 галузей знань SWEBOK
Вимоги до програмного забезпечення (Software Requirements)	Вимоги до програмного забезпечення (Software requirements).
Розробка програмного забезпечення (Software Design)	Розробка програмного забезпечення (Software design).
Конструювання програмного забезпечення (Software Construction)	Конструювання програмного забезпечення (Software construction).
Тестування програмного забезпечення (Software Testing)	Тестування програмного забезпечення (Software testing).
Супроводження програмного забезпечення (Software Maintenance)	Супроводження програмного забезпечення (Software maintenance)
Управління конфігурацією програмного забезпечення (Software Configuration Management)	Управління конфігурацією програмного забезпечення (Software configuration management).
Управління програмною інженерією (Software Configuration Management)	Управління проектами програмного забезпечення (Software engineering management).
Процеси інженерії програмного забезпечення (Software Engineering Process)	Процеси інженерії програмного забезпечення (Software engineering process).
Моделі та методи програмного забезпечення (Software Engineering Models and Methods)	Засоби та інструменти програмного забезпечення (Software engineering tools and methods).
Якість програмного забезпечення (Software quality)	Якість програмного забезпечення (Software quality).
Професійні навички з програмної інженерії (Software Engineering Professional Practices)	
Економіка інженерії програмного забезпечення (Software Engineering Economics)	

Обчислювальні основи (Computing Foundations)	
Математичні основи (Mathematical Foundations)	
Основи інженерії (Engineering Foundations)	

Також SWEBOOK визначає дисципліни, що відіграють велику роль в програмній інженерії:

Суміжні дисципліни (2014)	Суміжні дисципліни (2004)
Комп'ютерна інженерія (Computer Engineering)	Комп'ютерна інженерія (Computer Engineering)
Комп'ютерні науки (Computer Science)	Комп'ютерні науки (Computer Science)
Загальний менеджмент (General Management)	Менеджмент (Management)
Математика (Mathematics)	Математика (Mathematics)
Управління проектами (Project Management)	Ергономіка ПЗ (Software Ergonomics)
Управління якістю (Quality Management)	Контроль якості (Quality Management)
Системна інженерія (Systems Engineering)	Системне адміністрування (Systems Administrating)

Програмне забезпечення

Визначення. Програмне забезпечення (ПЗ) безліч логічних розпоряджень, що розвиваються в часі, за допомогою яких деякий колектив людей управляє і використовує багатопроцесорну і розподілену систему обчислювальних пристроїв.

Це *визначення*, яке дане Харальдом Мілсом, відомим фахівцем в галузі програмної інженерії з компанії *IBM*, містить в собі таке.

1. Логічні розпорядження – це не лише самі програми, але і різна документація (наприклад, з експлуатації програм) і ширше – певна система відносин між людьми, що використовують ці програми в межах деякого процесу діяльності.
2. Сучасне ПЗ призначене, як правило, для одночасної роботи з багатьма користувачами, які можуть бути значно віддалені один від одного у фізичному просторі. Таким чином, обчислювальне середовище (персональні комп'ютери, сервера і так далі), в якому ПЗ функціонує, виявляється розподіленим.

3. Завдання, які вирішуються сучасним ПЗ, часто вимагають різних обчислювальних ресурсів через різну спеціалізацію цих завдань, із-за великого об'єму виконуваної роботи, а також з міркувань безпеки. Наприклад, з'являється сервер бази даних, сервер додатків і ін. Таким чином, обчислювальне середовище, в якому ПЗ функціонує, виявляється багатопроцесорним.
4. ПЗ розвивається в часі – виправляються помилки, додаються нові функції, випускаються нові версії, міняється його апаратна база.

Властивості. Таким чином, ПЗ є складною динамічною системою, що включає технічні, психологічні і соціальні аспекти. ПЗ помітно відрізняється від інших видів систем, що створюються (створених) людиною, – механічних, соціальних, наукових і ін., і має такі особливості, які виділені Фредеріком Бруксом (*Фредерік Філіпс Брукс — інженер програмного забезпечення та вчений-інформатик, найбільш відомий за управління розробкою ОС System/360 для IBM, та її наступників, а пізніше за опис цього процесу в книжці «Міфічний людино-місяць».*) в його статті "Срібної кулі немає".

1. **Складність** програмних об'єктів, яка істотно залежить від їх розмірів. Як правило, більше ПЗ (більша кількість користувачів, більший об'єм оброблюваних даних, жорсткіші вимоги по швидкодії і ін.) з аналогічною функціональністю – це інше ПЗ. Класична наука будувала прості моделі складних явищ, і це вдавалося, оскільки складність не була характеристичною межею даних явищ. (Порівняння програмування саме з наукою, а не з театром, кінематографом, спортом і іншими галузями людської діяльності, виправдано, оскільки воно походить, головним чином, з математики, а перші його плоди – програми – призначалися для використання при наукових розрахунках. Крім того, більшість програмістів мають природничо-наукову, математичну або технічну освіту. Таким чином, парадигми наукового мислення широко використовуються при програмуванні – явно або неявно.)
2. **Узгодженість** – ПЗ ґрунтується не на об'єктивних посилках (подібно до того, як різні системи в класичній науці ґрунтуються на постулатах і аксіомах), а повинно бути узгоджене з великою кількістю інтерфейсів, з якими згодом воно повинне взаємодіяти. Ці інтерфейси погано піддаються стандартизації, оскільки ґрунтуються на численних людських угодах, що погано формалізуються.

3. **Змінність** – ПЗ легко змінити і, як наслідок, вимоги до нього постійно змінюються в процесі розробки. Це створює багато додаткових труднощів під час його розробки і еволюції.
4. **Нематеріальність** – ПЗ неможливо побачити, воно віртуальне. Тому, наприклад, важко скористатися технологіями, які засновані на попередньому створенні креслень, що успішно використовуються в інших промислових галузях (наприклад, в будівництві, машинобудуванні). Там на кресленнях в схемному вигляді відтворюються геометричні форми створюваних об'єктів. Коли об'єкт створений, ці форми можна побачити.