

ВІДКРИТИЙ МІЖНАРОДНИЙ УНІВЕРСИТЕТ РОЗВИТКУ ЛЮДИНИ „УКРАЇНА”  
Факультет комп’ютерних технологій  
Кафедра Інформаційних технологій та програмування

**Викладач : Кисельов Г. Д .**

**Конспект лекцій**  
**з дисципліни ”Основи програмної інженерії”**  
**для студентів напряму підготовки 6.0804**  
**“комп’ютерні науки”**  
**спеціальності – 8.080400**  
**– програмне забезпечення автоматизованих систем**

**Київ 2010р.**

## Вступ в програмну інженерію

Наприкінці 90-х р. минулого століття знання і досвід, що були накопичені в індустрії програмного забезпечення за попередні 30-35 років, а також більш ніж 15-літніх спроб застосування різних моделей розробки, усе це, нарешті, оформилося в те, що прийнято називати дисципліною програмної інженерії – *Software Engineering*. Якоюсь мірою, таке формування дисципліни на основі широко розповсюдженого практичного досвіду нагадує ті процеси, що відбувалися в управлінні проектами. Виникали і розвивалися професійні асоціації, спеціалізовані інститути, комітети зі стандартизації й інші утворення, що, зрештою, прийшли до спільної думки про необхідність зведення професійних знань по відповідним областях і стандартизації відповідних програм навчання.

### Програмна інженерія як дисципліна

У 1958 р. всесвітньо відомий статистик Джон Тьюкей (John Tukey) уперше ввів термін software - програмне забезпечення. У 1972 р. IEEE (Computer Society of the Institute for Electrical and Electronic Engineers, IEEE Computer Society – IEEE-CS (Комп'ютерне Суспільство) чи просто IEEE. <http://www.ieee.org/>) випустив перший номер Transactions on Software Engineering – Праці з програмної інженерії. Перший цілісний погляд на цю область професійної діяльності з'явився 1979 р., коли Комп'ютерне Суспільство IEEE підготувало стандарт IEEE Std 730 щодо якості програмного забезпечення. Після 7 років напружених робіт, у 1986 р. IEEE випустило IEEE Std 1002 "Taxonomy of Software Engineering Standards".

У 1990 р. почалося планування всеосяжних міжнародних стандартів, в основу яких лягли концепції і погляди стандарту IEEE Std 1074 і результатів роботи утвореної в 1987 р. спільної комісії ISO/IEC JTC 1 (International Organization for Standardization; IEC – International Electrotechnical Commission; JTC 1 – Joint Technical Committee 1, Information technology).

У 1995 р. група цієї комісії SC7 "Software Engineering" випустила першу версію міжнародного стандарту ISO/IEC 12207 "Software Lifecycle Processes". Цей стандарт став першим досвідом створення єдиного загального погляду на програмну інженерію. Відповідний національний стандарт Росії – ГОСТ Р ИСО/МЭК 12207-99 [ГОСТ 12207,1999] містить повний автентичний переклад тексту міжнародного стандарту ISO/IEC 12207-95 (1995 р.).

У свою чергу, IEEE і ACM (Association of Computer Machinery), почавши спільні роботи ще в 1993 р. з кодексу етики і професійної практики в даній галузі (ACM/IEEE-CS Code of Ethics and Professional Practice), до 2004 р. сформулювали два ключових описи того, що сьогодні ми і називаємо основами програмної інженерії – Software Engineering:

1. Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE 2004 Version - Керівництво до Зводу Знань з Програмної Інженерії, надалі просто "SWEBOK" [SWEBOK, 2004];
2. Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering - Навчальний План для Викладання Програмної Інженерії у Вищах (дана назва представлено у вільному перекладі) [SE, 2004].

Обидва стандарти стали результатом консенсусу провідних представників індустрії і визнаних авторитетів в галузі програмної інженерії - за аналогією з тим, як був створений PMI PMBOK. Так ми прийшли до сьогоденного стану Software Engineering як дисципліни.

### **SWEBOK: Керівництво до зводу знань з програмної інженерії**

З 1993 р. IEEE і ACM координують свої роботи в рамках спеціального спільного комітету - Software Engineering Coordinating Committee (SWECC - <http://www.computer.org/tab/swecc>). Проект SWEBOK був ініційований цим комітетом у 1998 р. Оцінений можливий обсяг змісту SWEBOK і інші фактори привели до того, що було рекомендовано проводити роботи з реалізації проекту не тільки силами добровольців з рядів експертів індустрії і представників найбільших споживачів і виробників програмного забезпечення, але і на основі принципу "повної зайнятості". Базовий комплекс робіт, у відповідності зі спеціальним контрактом, був переданий у Software Engineering Management Research Laboratory Університету Квебек у Монреалі (Universite du Quebec a Montreal).

Серед компаній, що підтримали цей унікальний проект були Boeing, MITRE, Raytheon, SAP. У результаті проекту, здійсненого за фінансової підтримки цих і інших компаній і організацій, а також з урахуванням його значимості для індустрії, SWEBOK Advisory Committee (SWAC) прийняв рішення зробити SWEBOK загальнодоступним – <http://www.swebok.org.Y/> перспективі, якщо вдасться забезпечити відповідний рівень фінансування, SWAC вважає за необхідне закінчену версію SWEBOK 2008 зробити також вільно доступною на Web-сайті проекту. Сьогоднішня "публічність" (загальнодоступність) результатів проекту стала можлива, у першу чергу, саме завдяки підтримці SWEBOK Industrial Advisory Board (IAB) – структури, що поєднує представників компаній, що підтримали проект.

Проект SWEBOK планувався у вигляді трьох фаз: Strawman ("солом'яна людина"), Stoneman ("кам'яна людина") і Ironman ("залізна людина"). До 2004 р. була випущена версія Посібника зі Зводу Знань 3-ї фази - Ironman, тобто максимально наближена до остаточного варіанту і схвалена IEEE у лютому 2005 р. до публікації в якості Trial-версії. Основна мета поточної "пробної" версії SWEBOK – поліпшити представлення, цілісність і корисність матеріалу керівництва на основі збору й аналізу відгуків на дану версію для того, щоб випустити фінальну редакцію документа в 2008 р. З ряду обґрунтованих причин, "SWEBOK є досить консервативним" [SWEBOK, 2004, с. В-2].

Після 6 років безпосередніх робіт над документом, SWEBOK включає "лише" 10 галузей знань (knowledge areas, KA). При цьому, що справедливо і для PMBOK, додавання нових галузей знань у SWEBOK досить прозоре. Усе, що для цього

потрібне, зрілість (чи, принаймні, явний і швидкий процес досягнення зрілості) і загальноприйнятності відповідної галузі знань, якщо це не призведе до серйозного

ускладнення SWEBOOK (\*концепція “загальноприйнятності” - generally accepted – визначена в IEEE Std 1490--1998, Adoption of PMI Standard — A Guide to the Project Management Body of Knowledge).

Важливо розуміти, що програмна інженерія є дисципліною, що розвивається. Більш того, дана дисципліна не стосується питань конкретизації застосування тих чи інших мов програмування, архітектурних рішень або, тим більше, рекомендацій, що стосуються більш-менш розповсюджених чи тих, що розвиваються, з тим чи іншим ступенем активності/помітності технологій (наприклад, web-служб).

Керівництво до зводу знань, а таким є SWEBOOK, включає базове визначення й опис галузей знань (наприклад, конфігураційне управління – configuration management) і, безумовно, є недостатнім для охоплення всіх питань, що відносяться до питань створення програмного забезпечення, але, у той же час потрібним для їхнього розуміння.

Необхідно відзначити, що однією з найважливіших цілей SWEBOOK є саме визначення тих аспектів діяльності, що складають суть професії інженера-програміста.

## **Структура і зміст SWEBOOK**

Опис галузей знань у SWEBOOK побудовано за ієрархічним принципом, як результат структурної декомпозиції. Така ієрархічна побудова звичайно нараховує два-три рівня деталізації, прийнятих для ідентифікації тих чи інших загальновизнаних аспектів програмної інженерії. При цьому, структура декомпозиції галузей знань деталізована тільки до того рівня, що потрібен для розуміння природи відповідних тем і можливості пошуку джерел компетенції й інших довідкових даних і матеріалів. У принципі, вважається, що як такий “звід знань” з програмної інженерії представлений не в обговорюваному керівництві (SWEBOOK), а в першоджерелах (як зазначених у ньому, так і представлених за його рамками) [SWEBOOK, 2004, с.1-2 ].

SWEBOOK описує 10 галузей знань:

1. Software requirements – програмні вимоги
2. Software design – дизайн (архітектура)
3. Software construction – конструювання програмного забезпечення
4. Software testing - тестування
5. Software maintenance – експлуатація (підтримка) програмного забезпечення
6. Software configuration management – конфігураційне управління
7. Software engineering management – управління в програмній інженерії
8. Software engineering process – процеси програмної інженерії
9. Software engineering tools and methods – інструменти і методи
10. Software quality – якість програмного забезпечення.

На додаток до них SWEBOOK також включає огляд суміжних дисциплін, зв'язок з якими представлений як фундаментальний, важливий та обґрунтований для програмної інженерії:

1. Computer engineering
2. Computer science
3. Management
4. Mathematics
5. Project management
6. Quality management
7. Software ergonomics
8. Systems engineering

Варто відзначити, що прийняті розмежування між галузями знань, їх компонентами (subareas) і іншими елементами досить довільні. При цьому, на відміну від PMBOK, галузі знань SWEBOOK не включають “входи” і “виходи”. Деякою мірою така декомпозиція пов'язана з тим, що SWEBOOK не асоційовано з тією чи іншою моделлю (наприклад, життєвого циклу) чи методом. Хоча на перший погляд перші п'ять галузей знань у SWEBOOK представлені в традиційній послідовній (каскадній - waterfall) моделі, це не більш ніж слідування прийнятій послідовності висвітлення відповідних тем. Інші галузі і структура декомпозиції галузей представлені за алфавітом.

На рис. 1 представлені перші п'ять галузей знань.

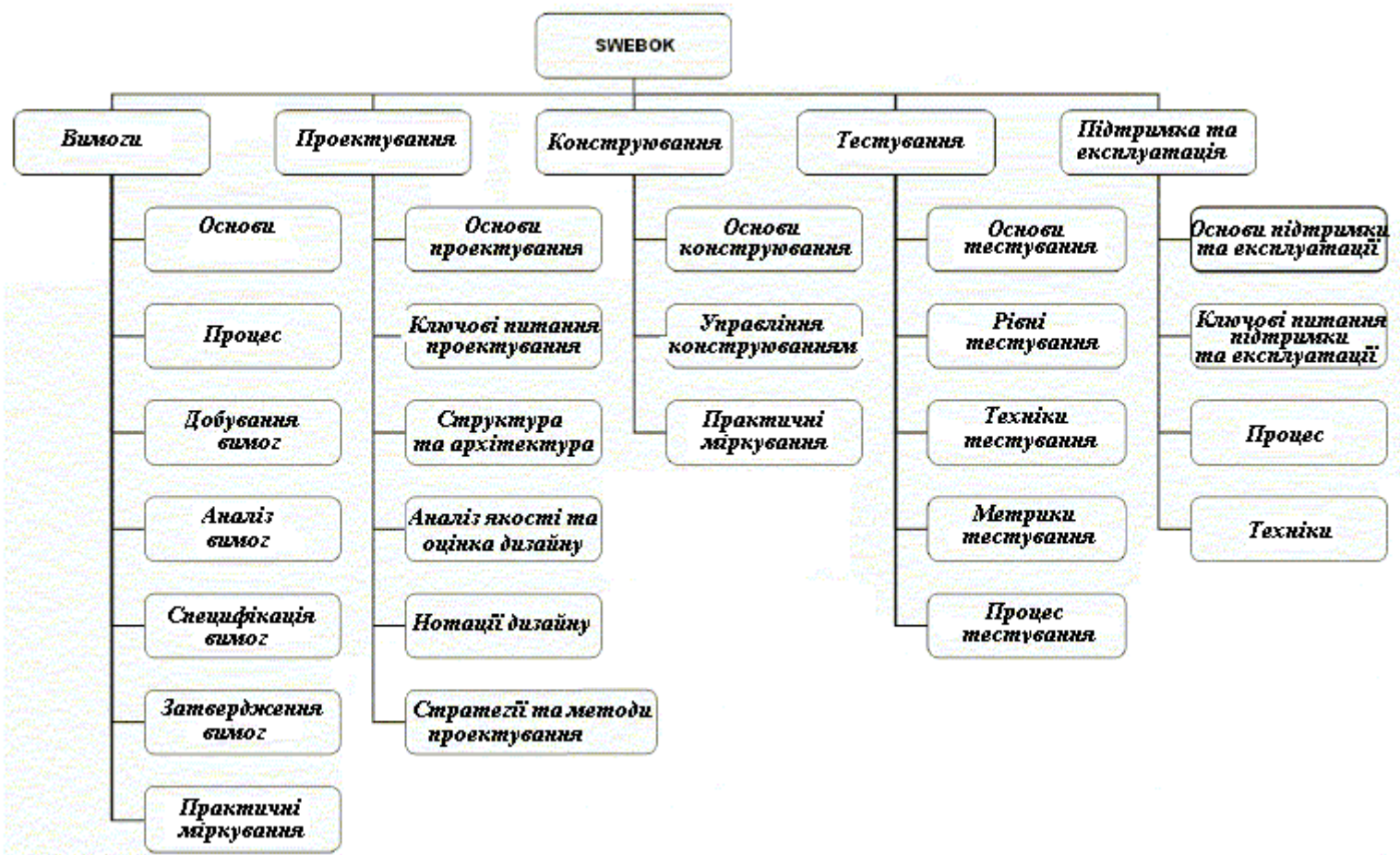


Рис. 1. Перші п'ять галузей знань [SWEBOOK, 2004, с.1-8, рис.2 ]