

УДК 378.147:004.4

Ліщина Н.М., Ліщина В.О., Повстяна Ю.С.

Луцький національний технічний університет

## ПІДХОДИ ДО ПІДГОТОВКИ ФАХІВЦІВ З РОЗРОБКИ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

**Ліщина Н.М., Ліщина В.О., Повстяна Ю.С.** Підходи до підготовки фахівців з розробки та тестування програмного забезпечення у вищих навчальних закладах. У статті розглянуто основні підходи до підготовки бакалаврів програмної інженерії. Проаналізовано рекомендації для розробки навчальних планів підготовки бакалаврів з розробки та тестування програмного забезпечення. Дається визначення базових понять, об'єктів, процесів і змісту дисциплін. Описано основу інженерії – базові елементи процесу виготовлення програмного продукту.

**Ключові слова:** професійна підготовка, програмна інженерія, навчальний план, інженер програміст, освітні знання з програмної інженерії.

**Ліщина Н.М., Ліщина В.О., Повстяна Ю.С.** Подходы к подготовке специалистов по разработке и тестированию программного обеспечения в высших учебных заведениях. В статье рассмотрены основные подходы к подготовке бакалавров программной инженерии. Проанализированы рекомендации по разработке учебных планов подготовки бакалавров по разработке и тестированию программного обеспечения. Дается определение базовых понятий, объектов, процессов и содержания дисциплин. Описаны основы инженерии – базовые элементы процесса изготовления программного продукта.

**Ключевые слова:** профессиональная подготовка, программная инженерия, учебный план, инженер программист, образовательные знания по программной инженерии.

**Nataliia Lishchyna, Valeriy Lishchyna, Yuliya Povstyana.** Approaches to training specialists in the development and testing of software in higher education. The article reviews the main approaches to the education of bachelors of Software Engineering. Recommendations for the development of curricula for bachelors to develop and test software are analyzed. The base entity, objects, process and concepts of discipline are defined. Foundation engineering –bazovi elements of the manufacturing process software are described.

**Keywords:** professional training, software engineering, curriculum, software engineer, software engineering education knowledge (SEEK).

**Поставка проблеми.** В результаті багаторічної праці всесвітнього програмуючого загалу накопичилась велика кількість знань та досвід побудови різноманітних комп'ютерних програм. Вони знайшли відображення у конкретних програмних продуктах широкого застосування та у сукупності теоретичних і прикладних методів і засобів, принципів і правил, а також цілісних процесів виробництва комп'ютерних систем за участю колективів програмістів і інженерів. В рамках багатогранної діяльності теоретиків та практиків у галузі програмування сформувалися формальні методи верифікації і тестування програм, математичні моделі надійності, методи оцінювання показників якості програмних продуктів тощо.

**Аналіз останніх публікацій та досліджень.** Дослідженнями в галузі ПЗ в період кінця ХХ ст. – початку ХХІ ст. займаються як окремі вчені (З. Дудар, Е. Лавріщева, В. Ліпав, С. Орлік, М. Вроу, Т. Лейтбридж, В. Мейер, І. Соммервілль, Н. ван Вліет), так і авторські колективи: Харківський національний університет радіоелектроніки, Association for Computing Machinery, Institute of Electrical and Electronics Engineers. Результатом цих наукових досліджень є низка підручників та наукових статей, які відображають різні теоретичні, методичні та професійні аспекти.

**Формування мети дослідження.** В статті розглянемо основні підходи до підготовки бакалаврів з програмної інженерії. Проаналізуємо рекомендації для розробки навчальних планів підготовки бакалаврів з розробки та тестування програмного забезпечення. Розглянемо основу інженерії – базові елементи процесу виготовлення програмного продукту.

**Виклад основного матеріалу.** Сьогодні знання й досвід, накопичені в індустрії ПЗ за попередні десятиріччя, оформилися в окрему дисципліну ПЗ. Завдяки розпочатій ще в 1993 р. спільній роботі Асоціації з обчислювальної техніки (Association of Computer Machinery, ACM) та Інституту інженерів з електротехніки й електроніки (Institute for Electrical and Electronic Engineers, IEEE) в 2004 р. були розроблені дві фундаментальні програми з програмної інженерії (Software Engineering):

1. Guide to the Software Engineering Body of Knowledge (SWEBOOK), IEEE 2004 Version – Керівництво із отримання знань з програмної інженерії.

2. Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2004) – Рекомендації з викладання програмної інженерії в університетах.

Керівництво SWEBOOK містить лише базові визначення й описи змістових ліній (ЗЛ), однак не охоплює всіх питань створення ПЗ. Опис ЗЛ в SWEBOOK побудовано за ієрархічним

принципом. При цьому деталізація ЗЛ виконана тільки на тому рівні розуміння природи відповідних тем, який дає можливість самостійного знаходження джерел компетенції, довідкових даних матеріалів. SWEBOOK описує 10 ЗЛ:

- Software requirements – програмні вимоги.
- Software design – дизайн (архітектура).
- Software construction – конструювання програмного забезпечення.
- Software testing – тестування.
- Software maintenance – експлуатація (підтримка) ПЗ.
- Software configuration management – конфігураційне керування.
- Software engineering management – управління в програмній інженерії.
- Software engineering process – процеси програмної інженерії.
- Software engineering tools and methods – інструменти й методи.
- Software quality – якість програмного забезпечення.

Метою SWEBOOK являється саме визначення й систематизація тих аспектів діяльності, які є підґрунтям професійної підготовки інженера-програміста.

У рекомендаціях SE2004 зазначено, що ПІ є одночасно обчислювальною й інженерною дисципліною. У SE2004 наводяться зразки навчальних планів із спеціальності ПІ; подано принципи викладання курсу ПІ, перелік знань, умінь та навичок (Software Engineering Education Knowledge, SEEK), якими має оволодіти студент при вивченні цього курсу:

- уміти індивідуально й у групі працювати над створенням програм; володіти знаннями й навичками, достатніми для практичної роботи;
- знаходити оптимальні проектні рішення в умовах обмежень (техніко- технологічних, часових, фінансових, людських тощо);
- проектувати в декількох предметних галузях; освоювати нові моделі, методи й технології;
- розуміти й застосовувати теорії, моделі та методи ПІ; мати навички міжособистісного спілкування, колективної роботи й лідерства [1].

Навчальний матеріал SEEK поділений на десять ЗЛ, кожна з яких має кілька модулів, розбитих за темами:

Основи ІТ (інформатика; технології та інструментальні засоби створення програм).

Основи математики та інженерії (дискретна математика, граматики, числові помилки, статистика, виміри; емпіричні методи та методики експериментів; пошук рішення, взаємоприйнятне для учасників проекту, розміщення пріоритетів, аналіз компромісів).

Професійна практика (групова динаміка, психологічні принципи; навички спілкування; етика, професійне поведіння, юридичні питання, професійні суспільства, стандарти).

Моделювання та аналіз ПЗ (принципи та мови моделювання, синтаксис семантика; поведінкове, структурне та доменне моделювання; аналіз правильності, коректності і якості; інженерія вимог).

Проектування програмного забезпечення (принципи, компроміси, завдання, шаблони, архітектура, стратегії, проектування людино-машинного інтерфейсу, детальне проектування, інструментарій, оцінка).

Верифікація та атестація ПЗ (мети, параметри й показники, огляди, контрольний аналіз; тестування; аналіз людино-машинного інтерфейсу, аналіз проблем і підготовка відповідних звітів).

Еволюція програмного забезпечення (процеси, моделі, закони еволюції, планування еволюції, розуміння програм і зворотна інженерія, аналіз впливів, перенос, рефакторинг).

Процеси розробки програмного забезпечення (моделі циклу життя, стандарти, окремі процеси; інфраструктура процесів, моделювання процесів, аналіз якості процесів і контроль).

Якість програмного забезпечення (зацікавленість суспільства в якості, атрибути якості, витрати й наслідки поганої якості; процеси й стандарти якості; керування якістю процесів і продуктів).

Управління програмними проектами (моделі управління, планування, організація, контроль; управління випуском і конфігурацією).

Програмна інженерія як інженерна дисципліна (або інженерія) – це сукупність прийомів виконання діяльності, пов'язаної з виготовленням програмного продукту для різних видів цільових об'єктів із застосуванням методів, засобів і інструментів наукової складової програмної інженерії. Основу інженерії складають наступні базові елементи процесу виготовлення програмного продукту:

– ядро знань SWEBOOK, як набір теоретичних концепцій і формальних визначень стосовно методів і засобів розроблення та керування програмними проектами, які можуть застосовуватися у інженерії програмування;

– базовий процес ПІ, як стрижень процесній діяльності в організації-розробнику ППІ;

– стандарти, як набір регламентованих правил конструювання проміжних артефактів у процесах ЖЦІ;

– інфраструктура – умови середовища та методичне забезпечення базового процесу ПІ і підтримка дій його виконавців, що займаються виробленням програмного продукту;

– менеджмент проекту (РМВОК) – ядро знань з керування промисловими проектами, як набір стандартних процесів, а також принципів і методів планування і контролювання роботами в проекті.

З інженерної точки зору в програмній інженерії вирішуються задачі виготовлення ППІ, подані як технологічні процеси формування вимог, проектування і супроводу продукту, а також перевірки операцій базового процесу на правильність виконання різних функціональних задач проекту та вкладання робіт за проектом у заданий замовником строк.

Програмну інженерію можна розглядати з двох пов'язаних точок зору:

– як інженерну діяльність, у якій інженери різних категорій виконують роботи в рамках проекту, використовуючи відповідні теоретичні методи і засоби ПІ, що рекомендовані у ядрі знань SWEBOOK, а також стандарти процесів проектування цільових об'єктів за обраними методами;

– як систему керування проектом, якістю і ризиками з залученням правил і положень стандартів ЖЦІ, якості та менеджменту проекту.

Інженерна діяльність обов'язково планується та ґрунтується на розподілі робіт у проекті між різними категоріями виконавців. Менеджер проекту – це головна діюча особа проекту, відповідальна за проектування і контроль виконання робіт спеціальними службами інфраструктури проекту в організації, зокрема служби верифікації, тестування, якості тощо. Продукт колективного виготовлення передається замовнику для супроводу. В ньому можуть бути знайдені різні помилки і недоліки, які усувають розробники.

Ця діяльність у програмній інженерії практично вже відпрацьована і за своєю сутністю близька до інженерної діяльності у промисловості, де інженерія – це спосіб застосування наукових результатів у виготовленні технічних виробів на основі технологічних правил і процедур, методик виміру, оцінки і верифікації в цілях задоволення і отримання користі від виготовленого продукту або товару [1].

Ядро знань SWEBOOK – стислий опис концептуальних основ програмної інженерії. Структурно поділяється на 10 розділів (knowledge areas), які умовно можна розкласти за двома категоріями: проектування продукту і інженерна діяльність. Перша категорія – це методи і засоби розробки (формування вимог, проектування, конструювання, тестування, супровід), друга категорія – методи керування проектом, конфігурацією і якістю та базовим процесом організації-розробника.

Методи ядра знань програмної інженерії менеджер проекту зіставляє з відповідними стандартними процесами ЖЦІ, виконання яких забезпечує послідовне розроблення програмного продукту. Наповнення базового процесу програмної інженерії методами з ядра знань SWEBOOK, а також задачами і діями стандартного ЖЦІ, обумовлює його пристосування до потреб конкретної організації-розробника щодо певної регламентованої послідовності розробки і супроводу програмного продукту. Все це створює технологічний базис інженерії виготовлення конкретного продукту (або низки однотипних продуктів) в організації. На початкових стадіях розробки виконуються процеси визначення вимог до продукту, проектні рішення і каркас (абстрактна архітектура) майбутнього продукту. На основі вимог і каркасу розробляються або вибираються готові прості об'єкти для „наповнення” каркасу змістом для подальшого його доведення до стану готового продукту.

Базовий процес (БП) – це метарівень для забезпечення «процесного продукування» продукту. Він містить основні поняття стосовно оснастки, організаційної структури колективу розроблювачів та методології оцінки, виміру, керування змінами і удосконалювання самого процесу. В цілому базовий процес містить множину логічно пов'язаних з ним видів інженерної діяльності організації-розробника та набір засобів і інструментів щодо виготовлення програмного продукту [2].

Інфраструктура – це набір технічних, технологічних, програмних (методичних) та людських ресурсів організації-розробника, необхідних для виконання підпроцесів базового процесу

програмної інженерії, орієнтованого на виконання договору з замовником програмного проекту. До технічних ресурсів відносяться: комп'ютери, пристрої (принтери, сканери тощо), сервери і т.п. До програмних – загальносистемне ПЗ середовища розробки, напрацювання колективу, оформлені у вигляді повторно використовуваних компонентів та інформаційне забезпечення. Технологічні та методичні ресурси складають – методики, процедури, правила, рекомендації стандартів щодо процесу і керування персоналом, включаючи комплект документів, що встановлює регламент виконання і регулювання процесів ЖЦ, пристосованих для вирішення конкретних задач проекту. Людські ресурси – це групи розробників та служб керування проектом, планами, якістю, ризиком, конфігурацією та перевірки правильності виконання проекту розробниками.

Засоби, проміжні результати розроблення за процесами ЖЦ, а також методики керування різними ресурсами, виконання БП і застосування методів програмування, зберігаються у базі знань проекту.

Після виконання проекту і отримання досвіду побудови конкретного продукту, базовий процес і його окремі елементи, можуть удосконалюватися (через доопрацювання або зміни прийомів, залучення доробка, змінювання, додавання нових засобів) відповідно до вимог стандарту ДСТУ ISO/IEC 15504-7 (Оцінювання процесів ЖЦ ПЗ. Наставови з удосконалення процесу) з метою підвищення рівня можливостей і оцінки потужності процесу.

Готовність всіх видів забезпечення організації-розробника продуктів, досконалість виконуваних процесів набула якість створеного в ній продукту надають підстави для оцінки зрілості організації або сертифікації процесів виробництва ПЗ. Для оцінювання зрілості може застосовуватися модель зрілості CMM (Capability Maturity Models), запропонована Інститутом програмної інженерії SEI США, або інша модель, наприклад, Bootstrap, Trillium тощо. Модель CMM встановлює рівні зрілості організації стосовно створення програмних продуктів. Рівень зрілості визначається наявністю в організації базового процесу, всіх необхідних видів ресурсів (у тому числі і фінансових), відповідних стандартів і методик, а також професіональних здібностей (зрілості) членів колективу організації, здатних виготовляти програмні продукти в заданий строк і встановленої вартості. Модель CMM пропонує п'ять рівнів зрілості, від першого – найнижчого, до п'ятого – найвищого. П'ятий рівень зрілості процесів організації свідчить про здатність команди розробників створювати якісний програмний продукт [4].

Стандарти ПП – встановлюють технологічно відпрацьований набір процесів зі строго визначеним і регламентованим порядком проведення різних видів робіт у програмної інженерії, зв'язаних з розробленням програмного продукту і оцінюванням його якості, ризику тощо. Стандарти у галузі програмної інженерії регламентують різні напрямки діяльності щодо програмування програмних продуктів. Вони стандартизують термінологію і поняття, життєвий цикл, якість, вимірювання, оцінювання продуктів і процесів. Найбільш важливими серед них є стандарт ISO/IEC 12207 „Процеси життєвого циклу програмного забезпечення” (та його дещо застарілий вітчизняний еквівалент ДСТУ 3918-99), серія стандартів ДСТУ ISO/IEC 14598 “Оцінювання програмного продукту”, стандарт ДСТУ ISO 15939 “Процес вимірювання”, серія стандартів ДСТУ ISO/IEC 15504 “Оцінювання процесів ЖЦ ПЗ”, базові стандарти з якості - ДСТУ ISO 9001 «Системи управління якістю. Вимоги», ДСТУ 2844–94, ДСТУ 2850–94, що регламентують різні аспекти забезпечення якості ПП. Серед стандартів, що безпосередньо пов'язані з якістю ПЗ, слід також назвати проект нової серії стандартів ДСТУ ISO/IEC TR 9126 “Програмна інженерія. Якість продукту”.

У цих стандартах узагальнені знання спеціалістів з технології проектування і інженерних методів керування розробкою, починаючи від встановлення вимог, і закінчуючи оцінюванням якості продукту і можливою його подальшою сертифікацією. Процеси ЖЦ в стандарті ISO/IEC 12207 подають загальні положення, задачі й регламентовані дії по проектуванню, а також рекомендації щодо застосування цих процесів для розроблення і контролю проміжних результатів. У стандарті містяться також організаційні процеси – планування, керування і супроводу. Процес планування призначений для складання планів, графіків робіт щодо виконання проекту і розподілу робіт між різними категоріями фахівців, а також для контролю планів і виконаних робіт. Процес керування проектом визначає задачі та дії з керування роботами у проекті, виконуваними фахівцями, які володіють теорією керування, а також стеження за плановими строками, що надані замовником проекту. Процес супроводу – включає дії щодо покращення готового продукту, виявлення й усунення знайдених в ньому недоліків і внесення нових або видалення деяких функцій у продукт.

Ядро знань SWEBOOK і стандарти ЖЦ мають зв'язок. Процесам ЖЦ зіставляються необхідні методи ядра і тим самим визначається базовий процес проекту, що доповнюється методиками і обмеженнями щодо вироблення продукту. Діючі фундаментальні моделі ЖЦ (водоспадна, спіральна тощо), які широко використовуються на практиці, пропонують вкладений в них стиль проектування і реалізації деяких видів продуктів.

Менеджмент проекту – це керування розробленням проекту з використанням теорії керування та процесів ядра знань РМВОК (Project Management body of knowledge). В настанові з використання РМВОК подано положення і правила керування часовим виробничим циклом побудови унікального продукту в рамках проекту. РМВОК є стандартом, що розроблений американським Інститутом управління проектами ([www.pmi.org](http://www.pmi.org)), з початку без урахування рівня комп'ютеризації промисловості (1987р.), а потім і з його врахуванням (2000 р.). Слід зазначити, що на теперішній час настанови до РМВОК та SWEBOOK введені в статус стандартів, а саме: ISO/IEC TR 19759 ("Guide to the Software Engineering Body of Knowledge (SWEBOOK)) та IEEE Std.1490 "IEEE Guide adoption of PMI Standard. A Guide to the Project Management Body of Knowledge) [3].

Ядро знань РМВОК містить опис лексики, структури процесів і областей знань, відображаючи сучасну практику керування проектами в різних областях промисловості. В ньому визначені процеси ЖЦ проекту і головні області знань, згруповані за задачами: ініціація, планування, використання, моніторинг і керування, завершення. Крім того, область знань – інтеграція визначає прийняття рішень про використання ресурсів в кожному момент виконання проекту і керування загальними задачами проекту.

Область знань керування вмістом проекту включає процеси, які необхідні для виконання робіт за проектом, а також для його планування з розбивкою робіт на більш прості для спрощення процесу керування. Область керування якістю містить процеси і операції досягнення цілей проекту щодо якості, правила і процедури для поліпшення процесу досягнення цілей і забезпечення якості відповідно заданим вимогам. Область керування людськими ресурсами організації і розподілу робіт між виконавцями відповідно до їхньої кваліфікації містить процедури регламентування виконання робіт з розроблення програмного продукту [4].

**Висновки.** Стандарт РМВОК і стандарт ISO/IEC 12207, мають багато спільного, особливо стосовно організаційних процесів з керування проектом і інженерією доменів. На даний час розроблені механізми формалізованого опису об'єктів і компонентів, їхніх інтерфейсів, а також методи їхнього розміщення у репозитаріях компонентів і інтерфейсів проектів. Ядро знань SWEBOOK і РМВОК пов'язані подібними моделями ЖЦ, методами й інструментами керування процесами виконання проекту. Грані SWEBOOK, СТАНДАРТИ, РМВОК відображають виробничий фундамент, котрий застосовується при проектуванні програмних систем з використанням наукових і інженерних досягнень у програмній інженерії. Вони забезпечують технологічність та досягнення якості розроблення програмних продуктів різного призначення.

Наука й інженерія, SWEBOOK, СТАНДАРТИ, РМВОК як головні елементи програмної інженерії, зв'язані між собою процесами ЖЦ, теорією, методами проектування і керування розробкою проекту. Вони застосовуються при виробництві, як основні положення технології проектування програмних продуктів у певному середовищі розроблення.

1. Брой М. Теория и практика программной инженерии [Электронный ресурс] / Манфред Брой // Открытые системы. – 2011. – № 9. – Режим доступа : <http://www.osp.ru/os/2011/09/13011564/>
2. Лаврищева Е.М. Методы и средства инженерии программного обеспечения : учеб. пособ. / Е. М. Лаврищева, В. А. Петрухин. – М. : МФТИ, 2006. – 304 с.
3. Липаев В.В. Программная инженерия. Методологические основы: учеб. пособ. / В. В. Липаев. – М. : ТЕИС, 2006. – 608 с.
4. Лаврищева К.М. Программная инженерия / К.М. Лаврищева. – К., 2008. – 319 с.