

ПЕРСПЕКТИВНІ ДИСЦИПЛІНИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

У межах фундаментальних проєктів Інституту програмних систем НАН України в результаті багаторічних досліджень різних проблем програмної інженерії (ПІ) запропоновано нові дисципліни в складі ПІ, обґрунтовано їхній науковий зміст і завдання щодо створення відповідних навчальних курсів для студентів вищої школи та визначено їхню роль в індустріальному виробництві програмних продуктів (ПП).

Сьогодні ці розробки винятково актуальні, оскільки останніми роками в Україні науково-технічний рівень розроблення програмних продуктів значно знизився через відсутність національних програм розвитку виробничих напрямів ПІ та спеціальних структур, які б створювали національні програмні продукти для задоволення потреб різних галузей економіки й відповідно поповнення держбюджету. Відстаємо ми і з організацією навчального процесу. Якщо в провідних країнах світу вже понад 20 років практикують навчання предмета програмної інженерії, то в нас перші кроки в цьому напрямі почали робити тільки минулого року після відповідної Постанови Кабінету Міністрів України від 12 грудня 2006 р. У нас не вистачає новітніх наукових розробок у галузі ПІ, вітчизняних підручників та посібників, а також викладачів для підготовки спеціалістів із програмної інженерії, які зможуть у майбутньому підвищити рівень індустрії ПІ.

Пропонуємо читачам ознайомитися з основними положеннями результатів наукових досліджень НАН України з проблематики програмної інженерії.

Програмну інженерію застосовують близько сорока років. За цей час вона ввібрала в себе принципи математики, інформатики, програмування та комп'ютерних наук, сформувалася як комплекс засобів і методів для розвитку програмування і виробництва програмних продуктів на інженерній основі з необхідною продуктивністю та якістю. Міжнародна комп'ютерна спільнота спеціалістів зробила для розвитку ПІ досить багато, зокрема:

— побудовано ядро знань SWEBOOK (Software Engineering body of Knowledge, www.swebok.com, 2001р.) з десяти розділів, які стосуються принципів та методів розроблення й організації керування програмним забезпеченням (ПЗ);

— уведено в обіг наукові журнали з програмної інженерії — IEEE Trans. on SE, ACM Trans. on SE and Methodology, а також монографії і підручники з ПІ тощо;

— сформовано міжнародну програму навчання ПІ — Curricula-2004 (www.computing.org/education/cc2004, intuit.ru), у якій зафіксовані розділи SWEBOOK, які мають на меті навчити студентів розробляти ПП з інформатики в різних міжнародних університетах (в Україні її впроваджено з 2007р.);

— створено інститут SEI при Carnegie Mellon University (1984р.), кафедри ПІ в багатьох університетах (США і Європи) для розвитку теорії і практики ПІ;

— проведено низку міжнародних наукових конференцій із різних напрямів ІІ та опубліковано відповідні матеріали;

— запропоновано нові парадигми програмування (об'єктно-орієнтоване, компонентне, сервісне, агентне тощо);

— створено технологічні й інженерні лінії для розроблення цільових об'єктів — програмних систем, доменів, родин систем, програмних проектів тощо;

— оновлено (через реінженерію, реверсну інженерію) багато застарілих програм, накопичених на обчислювальних машинах попередніх поколінь, з метою їх повторного використання на платформах нових комп'ютерів;

— побудовано автоматизовані інструментальні засоби та середовища (Microsoft Visual Studio, MSF, Rational Rose, COM, CORBA тощо) підтримки розроблення і виробництва різних цільових об'єктів з окремих програмних ресурсів (компонентів, reusing, сервісів тощо).

Накопичений великий обсяг знань систематизовано й частково відображено в розділах ядра знань SWEBOOK, присвячених проектуванню і керуванню програмними продуктами (ІІІ), що ввійшли і в програму навчання Curricula-2004.

Однак ці розділи недостатньо повно відображають зміст ІІ з погляду індустрії, рівня науково-технічних напрацювань у ІІ (наприклад, відсутні теорія експертного аналізу проектних рішень, мови опису специфіки доменів, захисту даних, документування ІІІ тощо), а також теорій розроблення, керування та економіки. Крім того, вони зорієнтовані на реалізацію ІІЗ і не охоплюють нові цільові об'єкти програмної інженерії та середовища їх створення.

Для розв'язання важливих проблем індустрії програмної інженерії, відображення стану науково-технічних досягнень у ІІ та навчання студентів у складі ІІІ пропонуємо сформулювати нові дисципліни ІІІ, а саме — наукові, інженерні, управлінські, економічні, виробничі, освітянські тощо. Ці дисци-

пліни побудовані на сучасних підходах і науковому підґрунті відповідних фундаментальних наук. У цілому вони забезпечать різні напрями виробництва програмних засобів і складних об'єктів (доменів, родин систем, проектів тощо), що містять ІІЗ. На їх основі поліпшиться рівень підготовки спеціалістів (аналітиків, менеджерів, верифікаторів тощо), здатних забезпечити весь цикл виробничих робіт відповідно до специфічних потреб певного сегмента ринку програмної продукції.

Пропонуємо стисле тлумачення кожної з цих майбутніх дисциплін ІІІ, враховуючи їхнє місце в індустрії ІІІ і навчанні.

НАУКОВА ДИСЦИПЛІНА

Основою наукової дисципліни становлять класичні науки (теорія алгоритмів, множин, доказу, математична логіка тощо), а також теорія програмування та відповідні загальні мовні засоби проектування на певному рівні абстракції моделей і архітектур цільових програмних об'єктів. Як наука, вона містить основні поняття і об'єкти, теоретичні підходи, методи програмування та керування виготовленням ІІІ [1–9].

Основними поняттями ІІ є: дані та їхні структури (прості і складні), функції й композиції, базові та цільові об'єкти. Прості об'єкти розробляють на основі їх формального опису, а цільові об'єкти — із застосуванням інженерних методів керування виробництвом.

Теорія програмування — сукупність методів, мов і засобів формального проектування архітектури та специфікації складників цільових об'єктів, методів їх доказу (верифікації і тестування), систематичних і формальних методів програмування цих об'єктів та керування ресурсами (персоналом, матеріальними й фінансовими) і характеристиками, а також методів оцінювання результатів розроблення кінцевого продукту для досягнення показників якості ІІІ.

Таким чином, наукова дисципліна є теоретичним фундаментом ПІ і навчати її необхідно не тільки для підвищення рівня кваліфікації майбутніх фахівців із програмної інженерії, але й для підтримки та розвитку нових можливостей і засобів програмування, які вдосконалять відповідні напрями індустрії ПІ. Однією з важливих наукових проблем для індустріального виробництва ПІ є інтеграція (композиція, синтез) складників майбутнього продукту, базована на сумісності їхніх інтерфейсів. Її необхідно розв'язувати, спираючись на фундаментальну теорію синтезу, як одне з відгалужень наукової дисципліни ПІ.

Головним напрямом навчання у вищих навчальних закладах (ВНЗ) повинна стати саме наукова дисципліна, яку необхідно, на наш погляд, представити загальним теоретичним курсом, курсами систематичного програмування (об'єктно-орієнтованого, компонентного, агентного тощо), окремими класичними курсами та додатковими курсами з програми Curricula-2004, наприклад, інженерія якості [4].

ІНЖЕНЕРНА ДИСЦИПЛІНА

Інженерна дисципліна — це сукупність інженерних прийомів, засобів і стандартів, орієнтованих на виготовлення цільових об'єктів ПІ із застосуванням наукової дисципліни ПІ [3, 4]. Виходячи з нашого досвіду, базовими компонентами цієї дисципліни вважатимемо такі:

— ядро знань SWEBOOK як набір методів і засобів розроблення ПІ і керування проектами;

— базовий процес ПІ як стрижень процесної діяльності в організації-розробника ПІ;

— стандарти як набір регламентованих правил конструювання проміжних артефактів у процесах ЖЦ;

— інфраструктура — умови середовища та методичне забезпечення базового процесу ПІ й підтримки дій його виконавців, які виробляють ПІ;

— загальні засоби й інструментальні середовища підтримки вироблення ПІ.

Технологія інженерного виробництва ПІ базується на повторному використанні компонентів (ПВК), готових засобах, ресурсах й інструментах їх побудови. Такими технологіями є інженерія ПВК (Reuse Engineering), інженерія застосувань (Application Engineering), доменів (Domain Engineering), родин систем (Family Engineering) [3, 4, 6].

Інженерія ПВК сформувалася як систематична й цілеспрямована діяльність з пошуку і підбору компонентів, реалізованих і представлених у репозитаріях або бібліотеках ПВК [2, 3, 9]. Проектуючи такі компоненти, спочатку створюють загальну структуру — каркас продукту, описують його, пропонують обрані готові компоненти й ПВК. Як ПВК можуть бути використані прикладні системи (у бізнесі, економіці тощо) і загальносистемні програмні засоби (транслятори, редактори тестів, системи генерації, інтеграції тощо).

Інженерія застосувань та *інженерія доменів* ґрунтуються на багаторазовому використанні різнотипних ПВК. Головне завдання — це побудова прикладних систем і родин систем з урахуванням завдань домену та загальних і змінюваних характеристик представників родини. Технологія автоматизації домену набуває рис конвеєрного виробництва з ПВК, в основі якого лежить попередній опис моделі домену в DSL (Domain Specific Language) та специфікація членів родини [3]. Виробництво доменів базується на принципах керування планами робіт за графіками, контролі результатів робіт й оцінюванні рівня застосування готових ресурсів при реалізації специфіки домену і програмуванні його завдань.

Базові компоненти виробництва можна вдосконалювати (наприклад, відповідно до моделі СММ) і пристосовувати до типу цільових об'єктів, конкретних умов середовища і колективу виконавців.

Значення інженерної дисципліни у виробництві головне. Без інженерії неможливий жоден промисловий продукт. Тут потрібно ретельно дослідити всі напрацювання (як наукові, так й інженерні) у галузі комп'ютерних наук і на їх підґрунті створити фундамент інженерної дисципліни, яка включатиме опис стандартних принципів інженерії і базових компонентів, а також сучасних мов специфікації доменів, членів родин та процесів виробництва ПП за інженерними технологіями.

ДИСЦИПЛІНА КЕРУВАННЯ

Базисом цієї дисципліни є класична теорія керування складними системами, сучасний менеджмент проекту та відповідний стандарт IEEE Std.1490 — настанова до ядра знань РМВОК (Project Management Body of Knowledge). Теорію керування, а саме теорію організаційного керування, розробив академік В.М. Глушков. Вона перевірена практикою побудови технологічних процесів у металургійній, суднобудівельній і хімічній промисловостях, а також упроваджена в масове виробництво (зокрема, в АСУ «Львів»).

Теорія керування складними системами поширена й за кордоном, особливо в частині теорії планування виробництвом. Так, на фірмі «Дироп» з метою планування і складання планів-графіків великих комплексів робіт для модернізації її заводів розроблено метод CRM (Critical Path Method), базисом якого є графічне представлення робіт і різних видів операцій із зазначенням часу їхнього виконання. Інший метод мережевого планування PERT (Program Evaluation and Review Technique) випробувано під час реалізації проекту розроблення ракетної системи «Polaris», що поєднувала близько 3800 підрядників із кількістю операцій більш ніж 60 тис. Застосування цього методу було настільки успішним, що проект завершили на два роки раніше від запланованого терміну. Кожен із цих методів виник у надрах промислового виробництва, адаптований до середовища програмування і став базовим в індустрії програмних продуктів.

Теорія керування і планування відображена в стандарті РМВОК. У ньому визначені процеси ЖЦ-проекту і головні сфери знань, згруповані за завданнями: ініціація, планування, використання, моніторинг і керування, завершення. Головна сфера знань цього ядра — інтеграція — визначає концепцію керування організаційною діяльністю колективу виконавців проекту, базовану на методах прийняття рішень щодо ресурсів, загальних завдань, служб контролю правильності проекту та вкладання в задану замовником вартість [3, 4, 6].

Ці базові напрацьовані теорії керування та планування, стандартні положення РМВОК, серії стандартів ISO-9001 з якості та відповідне методичне забезпечення повинні стати основою дисципліни керування в ПП. Курс, присвячений вивченню цих питань, готуватиме у ВНЗ майбутніх висококваліфікованих менеджерів проектів та інших фахівців у галузі організаційного керування випуском ПП.

Ці базові напрацьовані теорії керування та планування, стандартні положення РМВОК, серії стандартів ISO-9001 з якості та відповідне методичне забезпечення повинні стати основою дисципліни керування в ПП. Курс, присвячений вивченню цих питань, готуватиме у ВНЗ майбутніх висококваліфікованих менеджерів проектів та інших фахівців у галузі організаційного керування випуском ПП.

ЕКОНОМІЧНА ДИСЦИПЛІНА

Економіка ПП є самостійною дисципліною предмета ПП зі своєю теорією і практикою оцінювання вартісних, часових і експертних показників стосовно складання контрактів на створення ПП, прийняття проектних рішень, подання вимог, розроблення архітектури тощо, визначення ризиків проектування за заданими ресурсами, проведення розрахунків за роботи виконавців та отриману якість ПП. Ця дисципліна найбільш розвинута з погляду методів економічних розрахунків у ПП, а саме — наявних методологій прогнозування розміру ПП (FPA— Function Points Analyses, Feature Points, Mark-N Function Points, 3D Function Points тощо), оцінювання витрат на розроблення ПП за допомогою родини моделей COCOMO або інших математичних моделей (Angel, Slim, Seer тощо) [4].

При формуванні цієї дисципліни необхідно використати фундаментальні економічні методи, пов'язані з принципами розподілу робіт у складних системах, методи розрахунків вартості окремих частин систем залежно від їхнього розміру і системи в цілому, чинні стандарти щодо оцінювання ПП тощо. Систематизований і науково обґрунтований курс економічної дисципліни ПП закрийє ту прогалину, яка існує через відсутність відповідних посібників і підручників для навчання фахівців, зайнятих у виробництві ПП.

ВИРОБНИЧА ДИСЦИПЛІНА

Головним питанням індустрії як такої є не тільки випуск відповідної продукції, але й отримання прибутку від неї. У галузі ПП великі прибутки від випуску програмної продукції отримують такі світові фірми, як Microsoft, IBM, Intel та інші, а також індійські фірми, які спеціалізуються на оновленні застарілих програмних систем. Виробництво ПП базується на технологічних процесах виготовлення продукту із застосуванням відповідних теорій проектування та інструментальних середовищ побудови ПП масового використання.

Першими спробами на шляху індустріального виробництва є технологічна підготовка розроблення ПП (ТПР) [10], лінії продуктів (Product line) (інституту SEI, США [11]) для задоволення ринкових потреб користувачів на деякі види програмної продукції, а також різновиди нових інженерних технологій виробництва ПП, а саме — інженерії застосувань, доменів, родин, діючі засоби підтримки їх виробництва (ОС, загальносистемні засоби, нові мови, інтегровані середовища розробки тощо).

ТПР застосовували, розробляючи АІС «Юпітер». Виробництво ПП на лінії продуктів здійснюють із готових програм, інформаційних ресурсів, ПВК, засобів та інструментів за технологічною лінією, у яку включають необхідні методи розроблення,

тестування й оцінювання отриманого результату. Технологія конструювання на лінії виконана за каркасом ПП із застосуванням низки підібраних ПВК в інструментальному середовищі, що містить необхідні засоби й інструменти виготовлення ПП і механізми відстеження ходу побудови продукту за встановленими планами.

За останні роки в Україні такі лінії та інструментальні середовища для виготовлення різних видів ПП здебільшого не розробляють. Поширеним явищем став аутсорсинг готових систем та інструментів, що становить більше як 35% від загального обсягу робіт із програмування. Виникають труднощі під час їх супроводження, оскільки ці продукти не завжди отримані за ліцензією і зазвичай не мають відповідної документації щодо принципів побудови і застосування.

Сьогодні в індустрії ПП не розв'язано проблеми оцінювання складності об'єктів і процесів виготовлення ПП, замало робіт, у яких було б запропоновано шляхи подолання складності, особливо за умов інтеграції великих об'єктів із різних простих і готових ресурсів.

Ця дисципліна, як предмет навчання, повинна включати класичні методи і технології виробництва різних продуктів, опис специфічних особливостей об'єктів проектування з урахуванням готових ресурсів, їхніх специфікацій, стандартів виробництва та документування готового продукту.

НАВЧАННЯ ПРЕДМЕТА ПІ

Усі розглянуті вище базові теорії і дисципліни ПП будуть предметом навчання інформатики студентів ВНЗ у майбутньому.

Ми розробили перший українськомовний посібник [2], присвячений основам ПП, який сьогодні використовують у декількох університетах. Крім того, розроблено новий підручник російською [5] і українською [7] мовами, який уже відображає деякі з визна-

чених нових дисциплін ПІ, а також методи програмування, верифікації і тестування програм. У ньому подано базові елементи й інструменти інженерії розроблення різних цільових об'єктів ПІ, а також процеси ЖЦ, методи керування колективами виконавців, якістю, тривалістю і вартістю робіт. Описано базове ядро SWEBOOK, РМВОК і стандарти, а також їхнє призначення в інфраструктурі менеджменту проекту. Підручник ставить за мету навчити студентів теорії і практики створення комп'ютерних програм у сучасних інструментальних системах та сприяти розвитку їхніх творчих здібностей щодо створення нових теорій у комп'ютерних науках.

Структура цього підручника відповідає робочій програмі Curricula-2004, а саме – типовій програмі SE-201 з базовими темами ядра SWEBOOK. Він відповідає сучасним вимогам навчання, але потребує ще вдосконалення – необхідно додати опис нових дисциплін ПІ.

* * *

Отже, запропоновані дисципліни ПІ є узагальненням багаторічних досліджень, практики розроблення ПІ і виконання фундаментальних проектів НАН України, наукових публікацій [1-8], пов'язаних з ПІ, а також викладання курсів лекцій із технології програмування та програмної інженерії в Київському національному університеті імені Тараса Шевченка та МФТІ при Інституті кібернетики ім. В.М. Глушкова НАН України. Безсумнівно, ці дисципліни доповнять SWEBOOK новими обґрунтованими напрямками. Їх розгляне відповідний комітет Curricula-2010 як пропозиції від України до робочої програми, як головні важелі навчання та індустріального розвитку ПІ.

1. Андон Ф.И., Лаврищева Е.М. Методы инженерии распределенных компьютерных приложений. — К.: Наук. думка, 1998. — 228 с.
2. Бабенко Л.П., Лаврищева Е.М. Основы программной инженерии. — К.: Знання, 2001. — 269 с.

3. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика. — К.: Наук. думка, 2006. — 450 с.
4. Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль, Т.М. Коротун, Е.М.Лаврищева, В.Ю. Суслов. — 2-е изд. — К.: Академперіодика, 2007. — 672 с.
5. Лаврищева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения: Учебное пособие. — М.: Министерство образования и науки РФ, 2007. — 415 с.
6. Лаврищева Е.М. Становление и развитие модульно-компонентной инженерии программирования в Украине. — К.: Институт кибернетики им. В.М. Глушкова, 2008. — 33 с. (Препринт 2008-1).
7. Лаврищева Е.М. Программная инженерия — научная и инженерная дисциплина // Кибернетика и системный анализ. — 2008. — №3. — С. 19–28.
8. Лаврищева К.М. Програмна інженерія: Підручник.— ВНУ, 2008. — 435 с.
9. Бабенко Л.П. Проблемы повторного использования в программной инженерии // Кибернетика и системный анализ. — 1999. — №2. — С. 155–166.
10. Лаврищева Е.М. Основы технологической разработки прикладных программ СОД. — К., 1987. — 29 с. (Препр. / Ин-т кибернетики им. В. М. Глушкова; (87–5)).
11. Northrop L.M. SEI's Software Product Line Tenets // IEEE Software. — 2002. — Vol. 19. — №4. — P. 32–39.

К. Лаврищева

ПЕРСПЕКТИВНІ ДИСЦИПЛІНИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Резюме

У статті подано класифікацію майбутніх дисциплін програмної інженерії (Software Engineering — SE). Визначено структуру, зміст, базові поняття та призначення кожної з них, а також їхню роль у навчанні спеціальності в індустріальному виробництві програмних продуктів. Запропоновано ввести ці перспективні дисципліни в програму навчання Curricula-2010.

К. Lavrischeva

PROSPECT DISCIPLINES OF SOFTWARE ENGINEERING

Summary

A classification of promising future disciplines of Software Engineering is presented in the article. The structure, essence, basic concepts, and destination of each discipline, and also their role in specialty teaching and software product manufacturing are defined. It is proposed to add these promising disciplines to the program of Curricula-2010.