

Програмне забезпечення ІТ-компанії та фахова підготовка студентів напрямку “Комп’ютерні науки” на факультетах електроніки та прикладної математики ЛНУ ім. Івана Франка
Злобін Г.Г., Скоропад О., Рикалюк Р.

ЛНУ ім. Івана Франка, Львів, вул. Тарнавського 107, zlobingg@gmail.com; ЕРАМ-Львів, sko@ukr.net; ЛНУ ім. Івана Франка, Львів, вул. Університетська 1, r.rykaluk@gmail.com

Software development company is a factory for the production of IT products. Like any factory, this company has complex structure and big set of processes with a lot of the necessary tools. All these tools are special applications for software development. This article describes a simple software landscape of modern IT company and explains major functionality of components. Particular attention is paid to free software.

Сучасна ІТ компанія – це фабрика з випуску програмних продуктів. Як і кожна фабрика, така компанія має свою структуру та налагоджений складний виробничий процес з великою кількістю необхідних інструментів. В ролі інструментів виступають програмні продукти призначені для розробки програмного забезпечення на всіх його етапах, а оскільки таких етапів є багато, то і перелік продуктів є дуже широким.

Спробуємо класифікувати програмне забезпечення для розробки:

1. Інструменти програмної архітектури, документування та аналітики призначені для візуалізації ідеї програмного проекту та візуалізації блок-схеми та основних компонентів майбутнього ПЗ;
2. Інструменти розробника (редактори, бази знань та інструменти відслідковування помилок /bug-track tool/);
3. Засоби контролю коду (code review);
4. Система контролю версій;
5. Центральний репозитарій програмного коду;
6. Програмні засоби безперервної інтеграції;
7. Програмні засоби автоматичного та ручного тестування;
8. Засоби доставки продукту до користувача, а також засоби комунікації з клієнтом.

Окремої уваги вимагає системне програмне забезпечення для підтримки роботи цієї інфраструктури, яке проте не входить в цей огляд:

1. Засоби віртуалізації;
2. Програмне забезпечення автоматизованого конфігурування;
3. Засоби розгортання програмних середовищ (orchestration tool) та інструменти управління контейнерами;
4. Засоби моніторингу;
5. Програмне забезпечення для резервного копіювання.

Це основний, але далеко не повний перелік типів продуктів, який розширюється настільки швидко, наскільки відбувається все вужча спеціалізація в індустрії програмного забезпечення.

Варто зупинитись на природі перерахованих продуктів. Якщо раніше і розробники програмного забезпечення, і його користувачі віддавали перевагу дорогим брендовим продуктам відомих компаній за принципом дорожче – краще, то зараз ситуація на IT-ринку кардинально змінилась: перевага надається безоплатно поширюваному, умовно-платному, а найбільше - відкритому програмному забезпеченню. Така переорієнтація ринку відбулась з наступних причин:

1. Код відкритого ПЗ вільний для модифікації чим забезпечується максимальна гнучкість проєктів;
2. Відкрите програмне забезпечення розвивається, по суті під наглядом спільноти, тому суттєво знижуються ризики пов'язані з неякісним кодом;
3. Повністю ліквідуються загрози, пов'язані з припиненням розробником супроводу програмного продукту, оскільки супровід відбувається консолідованими зусиллями самих користувачів;
4. Вартість проєктів з використанням відкритого ПЗ на порядок нижча за вартість таких же проєктів з використанням комерційного ПЗ.

Всі ці фактори приводять до того, що не лише користувачі, а й дедалі більше розробників починають переходити на відкрите ПЗ. Великі компанії з розробки ПЗ ефективно використовують практику тестування нових версій продуктів в якості відкритого ПЗ, а лише після цього ліцензують код найбільш стабільних версій. Яскравим прикладом такого методу є тандем CentOS – RedHat.

Проведемо короткий огляд основних спеціальних програмних продуктів, що широко використовуються компаніями з розробки ПЗ в промисловій експлуатації:

1. Засоби контролю коду – це програмне забезпечення, яке призначене для первинної перевірки програмного коду. Інша важлива функція цього ПЗ полягає в передачі знань від однієї людини, що пише код до інших спеціалістів, які в разі потреби зможуть продовжити написання програми. Вказане ПЗ часто вбудовується в системи контролю версій. Популярні пакети: Gerrit, Varkep, Phabricator;

2. Системи контролю версій та репозитарій – це централізоване сховище коду з організованою структуризацією версій ПЗ. Є головною базою всіх напрацьованих продуктів, дає можливість повернення в будь-який момент до будь-якої зафіксованої версії ПЗ, а також дає змогу створити незалежну гілку написання програмного продукту з довільної версії поточного продукту. Найвідоміші продукти: svn, git, pexus;

3. Системи безперервної інтеграції (CI - continuous integration) з'явилися відносно недавно у зв'язку з переходом на нові технології розробки ПЗ – Agile та Scrum. Це ПЗ дає змогу повністю автоматизувати найбільш рутинні операції розробника, якими є : завантаження залежностей та бібліотек, компіляція, блок-тестування, публікація результатів. Системи CI практично завжди працюють в тандемі з системою контролю версій та системами тестування, що дає змогу, наприклад, здійснювати автоматичне повернення до попередньої версії проєктованого ПЗ за наявності помилок в його коді. Приклади пакетів: jenkins, metacity (комерційне ПЗ), РНРСІ;

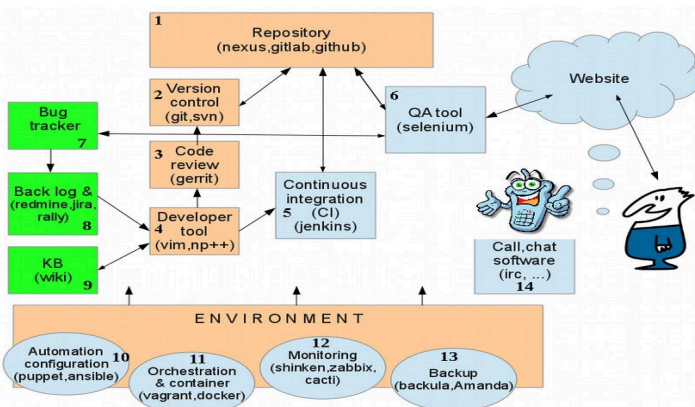


Рис. 1 1 – репозиторії (nexus, github, gitlab тощо), 2 – система контролю версій (git, svn тощо), 3 – система аналізу коду (gerrit), 4 – спеціалізовані текстові редактори із підсвіткою синтаксису (vim, np++ тощо), 5 – система безперервної інтеграції (jenkins, hudson, metacity, РНРСІ тощо), 6 – засоби тестування (Selenium IDE, Sikuli, WebDriver, PHPUnit, Амос тощо), 7 – система відслідковування помилок (BUGS, Bugzilla, eTraxis, GNATS тощо), 8 – система управління проектами і задачами (readmine, jira, rally, GitLab тощо), 9 – вікі-система для створення єдиної бази знань організації (Wiki Media, Confluence), 10 – система автоматичного конфігурування операційних систем і програм (puppet, ansible тощо), 11 – система управління віртуальними машинами (docker, vagrant тощо), 12 – система моніторингу служб і станів (Nagios, monit, shinken, zabbix, cacti тощо), 13 – системи управління резервним копіюванням (backula, amanda, ar+gzip, pgdump, mysqldump, windows backup тощо), 14 – засоби Інтернет-спілкування розробників програм (irc, Jabber/GoogleTalk, HipChat, Skype тощо)

4. Засоби тестування – широка лінійка програмних продуктів, що працює в галузі забезпечення надійності програмного коду (QA – quality assurance).

Вказане ПЗ призначене для контролю якості робочих версій коду і дає змогу відсіяти у разі хорошої організації процесу до 90% помилок в продуктах. Галузь QA інтенсивно розвивається, що призводить до появи великої кількості нового ПЗ для тестування. На сьогодні одних лише типів тестів нараховується близько десятка і для кожного з них існує своя лінійка тестового ПЗ. Найвідоміші пакети: Selenium IDE, Sikuli, WebDriver, PHPUnit, Атос.

5. Інструменти для документування коду, що полегшують подальшу підтримку та передачу ПЗ іншим командам. Ці інструменти це не звичайні коментарі між рядками, а структуровані коментарі, що пізніше слугують для генерування Web- та Pdf-документів із зручними для читання блок-схемами та списками класів та методів (DoxyGen, Swagger, Dexus);

6. Також окрім самих інструментів важливими є правила та методи роботи в команді. До методів можна віднести правила комунікації з використанням електронної пошти, Skype тощо;

7. Методології ведення проєктів – Agile, Scrum, WaterFlow та їх модифікації.

Як вже було зазначено, одною з найбільших переваг відкритого програмного забезпечення є широка доступність документації та навчальних матеріалів, що дає змогу ефективно та максимально швидко готувати персонал для роботи з ним. Процес підготовки проходить у вигляді так званих техконференцій у максимально вільному форматі, де встановлюється лише загальна тема та перелік властивостей програмного забезпечення, що обов'язкові до розгляду. Такий вид навчання нагадує обмін знаннями в професійних спільнотах і дуже добре зарекомендував себе.

На рис. 1 зображена блок-схема базового ПЗ, що широко використовується розробниками програмного забезпечення.

Для використання базового ПЗ, зображеного на рис. 1, у професійній діяльності бажано забезпечити його вивчення під час навчання у ВЗО або на підготовчих курсах в ІТ-компанії. Результати вступних співбесід на посаду системного інженера (Junior System Engineer) зазвичай показують високий рівень фундаментальної (теоретичної) підготовки випускників Львівського національного університету імені Івана Франка і водночас дуже слабкі знання промислових методів та інструментів розробки ПЗ. Спробуємо проаналізувати причини такого стану на прикладі двох факультетів Львівського національного університету імені Івана Франка - факультету електроніки та факультету прикладної математики та інформатики. З 2013 р. У навчальних лабораторіях електроніки факультету встановлено лише ліцензійне програмне забезпечення - у чотирьох лабораторіях ОС Linux та прикладне програмне забезпечення для неї, у трьох ОС Linux та ОС Microsoft Windows 7 за програмою Dream Spark. Причиною такого становища став провал платної факультетської підписки на Dream Spark через помилки

в апікаційних формах, зроблених представником факультету електроніки, та пізніше оформлення безоплатної кафедральної підписки на Dream Spark на кафедрі радіофізики та комп'ютерних технологій. У навчальних лабораторіях факультету прикладної математики та інформатики встановлено ОС Microsoft Windows 7 за програмою Dream Spark та програмне забезпечення для неї. В деяких лабораторіях встановлено також ОС Linux для забезпечення навчальних курсів, які торкаються безпосередньо цієї операційної системи.

На факультеті електроніки ЛНУ ім. Івана Франка читаються лекційні курси, які пов'язані з розробкою та використанням програмного забезпечення [1]:

- I. Алгоритмізація та програмування, перший семестр;
- II. Комп'ютерна графіка, перший семестр;
- III. Об'єктно-орієнтоване програмування, другий семестр;
- IV. Кросплатформне програмування, третій семестр;
- V. Технології комп'ютерного проектування, четвертий семестр;
- VI. Організація баз даних і знань, четвертий семестр;
- VII. Операційні системи, четвертий семестр;
- VIII. Web-технології та Web-дизайн;
- IX. Технології створення програмних продуктів, шостий семестр;
- X. Проектування інформаційних систем, шостий семестр;
- XI. Управління IT-проектами, восьмий семестр.

Незважаючи на те, що в шести комп'ютерних лабораторіях факультету електроніки встановлені лише ліцензійні ОС (в чотирьох лабораторіях лише Linux, в трьох лабораторіях Linux і Microsoft Windows 7) та ліцензійно чисте ПЗ, рівень використання базового ПЗ є поки що недостатнім. Наразі лише в наступних курсах спостерігається використання базового ПЗ

Таблиця 1.

Лекційні курси, семестр/Інструменти	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Операційні системи, 4	-	-	-	-	-	-	-	-	-	+	-	+	+	-
Технології створення програмних продуктів, 6	+	+	-	-	+	-	-	+	-	-	-	-	-	-
Управління IT-проектами, 8	-	-	-	-	-	-	-	*	-	-	-	-	-	-

Умовні позначення: + використовується у лекційному курсі, - не використовується у лекційному курсі, * лектор використовує Open Proejct тому, що Open Proejct подібний до Microsoft Proejct.

На факультеті прикладної математики та інформатики ЛНУ ім. Івана Франка читаються лекційні курси, які пов'язані з розробкою та використанням програмного забезпечення [2]:

- I. Програмування, перший-третій семестр;
- II. Організація та обробка електронної інформації, другий семестр;
- III. Програмне забезпечення, четвертий семестр;
- IV. Бази даних та інформаційні системи, п'ятий семестр;
- V. Паралельні та розподілені обчислення, п'ятий семестр;
- VI. Платформи корпоративних інформаційних систем, п'ятий семестр;
- VII. Програмування та підтримка веб-застосувань, п'ятий семестр;
- VIII. Бази даних та інформаційні системи, шостий семестр;
- IX. Обробка зображень та мультимедіа, шостий семестр;
- X. Проектування програмних систем, шостий семестр;
- XI. Програмування мовою Java, шостий семестр;
- XII. Операційні системи та системне програмування, сьомий семестр;
- XIII. Теорія програмування, сьомий семестр;
- XIV. Нечітке моделювання в середовищі Matlab, сьомий семестр;
- XV. Логічне та функціональне програмування, сьомий семестр;
- XVI. Операційні системи та системне програмування, восьмий семестр;
- XVII. Проектування систем штучного інтелекту, восьмий семестр;
- XVIII. Розподілені інформаційно-аналітичні системи, восьмий семестр;
- XIX. Об'єктно-орієнтовна система Smalltalk, восьмий семестр;
- XX. Програмування під UNIX-подібними системами, восьмий семестр.

Домінування ОС Microsoft Windows в навчальних лабораторіях факультету прикладної математики та інформатики неявно призводить до домінування закритого прикладного програмного забезпечення часом навіть із порушенням ліцензійності цього програмного забезпечення. Рівень використання базового ПЗ як і на факультеті електроніки є недостатнім, що призводить до того, що студенти обох факультетів не отримують належних знань з промислових методів та інструментів розробки ПЗ. Це є основною причиною організації ІТ-компаніями як у себе, так і у вищих закладах освіти (курси SoftServe у Львівському національному університеті імені Івана Франка, курси Ерам у Львівській політехніці тощо). На жаль, таке становище призводить до втрати інтересу у частини студентів до вивчення дисциплін з навчального плану вищого закладу освіти та переоцінки важливості технологічних знань, які викладаються на підготовчих курсах.

На думку авторів доповіді потрібно:

- I. усунути розрив між навчанням та виробничою практикою в ІТ-компаніях шляхом максимально можливого впровадження базового ПЗ в навчальні курси, які викладаються у вищих закладах освіти;

II. зміцнити зв'язки вищих закладів освіти з виробництвом шляхом стажування викладачів напрямку “Комп'ютерні науки” у провідних ІТ-компаніях на безоплатній основі;

III. відмовитись від стереотипу, який панує в головах багатьох викладачів вищих закладів освіти, про те, що пропрієтарне програмне забезпечення є найкращим та найнадійнішим.

Джерела

1. http://electronics.lnu.edu.ua/base_e/its/index.php?p=2/
2. <http://ami.lnu.edu.ua/academics/bachelor/curriculum-informatics>