

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ	7
ГЛАВА 1. ПРО ПРЕДМЕТ ВИВЧЕННЯ	7
1.1. ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ПРОГРАМНА ІНЖЕНЕРІЯ)	7
1.2. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	10
КОНТРОЛЬНІ ЗАПИТАННЯ	12
ГЛАВА 2. ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
2.1. ПРОЦЕС	13
2.2. ВДОСКОНАЛЕННЯ ПРОЦЕСУ	15
2.3. КЛАСИЧНІ МОДЕЛІ ПРОЦЕСУ	17
КОНТРОЛЬНІ ЗАПИТАННЯ ДО ГЛАВИ 2	24
ГЛАВА 3. РОБОЧИЙ ПРОДУКТ, ДИСЦИПЛІНА ЗОБОВ'ЯЗАНЬ, ПРОЕКТ ТА УПРАВЛІННЯ НИМ	26
3.1. РОБОЧИЙ ПРОДУКТ	26
3.2. ДИСЦИПЛІНА ЗОБОВ'ЯЗАНЬ	28
3.3. ПРОЕКТ	30
КОНТРОЛЬНІ ЗАПИТАННЯ	35
ГЛАВА 4. АРХІТЕКТУРА ПЗ	36
4.1. ОБГОВОРЕННЯ	36
4.2. ВИЗНАЧЕННЯ	37
4.3. МНОЖИННІСТЬ ТОЧОК ЗОРУ	39
4.4. МОВА UML	42
4.5. ВИДИ ДІАГРАМ	44
4.6. ПРИКЛАДИ	45
КОНТРОЛЬНІ ЗАПИТАННЯ	48
ГЛАВА 5. УПРАВЛІННЯ ВИМОГАМИ	49
5.1. ПРОБЛЕМА	49
5.2. ВИДИ І ВЛАСТИВОСТІ ВИМОГ	51
5.3. ВАРІАНТИ ФОРМАЛІЗАЦІЇ ВИМОГ	54
5.4. ДЕЯКІ ПОМИЛКИ ПІДЧАС ДОКУМЕНТУВАННЯ ВИМОГ	56
5.5. ЦИКЛ РОБОТИ З ВИМОГАМИ	56
КОНТРОЛЬНІ ЗАПИТАННЯ	57
ГЛАВА 6. КОНФІГУРАЦІЙНЕ УПРАВЛІННЯ	58
6.1. ПРОБЛЕМАТИКА	58
6.2. ОДИНИЦІ КОНФІГУРАЦІЙНОГО УПРАВЛІННЯ	60
6.3. УПРАВЛІННЯ ВЕРСІЯМИ	62
6.4. УПРАВЛІННЯ ЗБІРКАМИ	63
6.5. ПОНЯТТЯ BASELINE	65
КОНТРОЛЬНІ ЗАПИТАННЯ	66
ГЛАВА 7. УПРАВЛІННЯ ЯКІСТЮ ТА ТЕСТУВАННЯ	67
7.1. СТАНДАРТИЗАЦІЯ В СУЧАСНОМУ БІЗНЕСІ І ПРОМИСЛОВОСТІ	67
7.2. СТАНДАРТИЗАЦІЯ ЯКОСТІ	70
7.3. МЕТОДИ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПЗ	70
7.4. ТЕСТУВАННЯ	72
7.5. ВИДИ ТЕСТУВАННЯ	76
7.6. РОБОТА З ПОМИЛКАМИ	78
КОНТРОЛЬНІ ЗАПИТАННЯ	80
ГЛАВА 8. ДІАГРАМНА ТЕХНІКА В РОБОТІ ІЗ ЗНАННЯМИ	81
8.1. МЕТОД «ВАРІАНТИ ВИКОРИСТАННЯ»	81
8.2. ІТЕРАТИВНИЙ ЦИКЛ АВТОР/РЕЦЕНЗЕНТ	87
КОНТРОЛЬНІ ЗАПИТАННЯ	94

РОЗДІЛ 2. ПРОГРАМНІ ЗАСОБИ ПІДТРИМКИ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	95
ГЛАВА 9. MICROSOFT SOLUTIONS FRAMEWORK(MSF).....	95
9.1. Історія і поточний статус	95
9.2. Основні принципи MSF	97
9.3. Модель команди.....	98
9.4. Інші особливості.....	103
Контрольні запитання	104
ГЛАВА 10. CAPABILITY MATURITY MODEL INTEGRATION (СММІ) (ІНТЕГРАЦІЯ ЗАВЕРШЕНИХ ПРАЦЕЗДАТНИХ МОДЕЛЕЙ).....	106
10.1. Що таке СММІ?.....	106
10.2. Рівні зрілості процесів за СММІ	107
10.3. Галузі вдосконалення.....	108
Контрольні запитання	109
ГЛАВА 11. "ГНУЧКІ" (AGILE) МЕТОДИ РОЗРОБКИ.....	110
11.1. Загальні положення.....	110
11.2. EXTREME PROGRAMMING	110
11.3. SCRUM.....	112
Контрольні запитання	115
ГЛАВА 12. ОГЛЯД ТЕХНОЛОГІЇ MICROSOFT VISUAL STUDIO TEAM SYSTEM (VSTS).....	116
12.1. ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ПРОГРАМНА ІНЖЕНЕРІЯ)	116
12.2. СКЛАД ПРОДУКТУ	118
12.3. ПРАВИЛА ІНСТАЛЯЦІЇ.....	123
12.4. ПАКЕТ TEAM EXPLORER.....	124
Контрольні запитання	126
ГЛАВА 13. VSTS: УПРАВЛІННЯ ЕЛЕМЕНТАМИ РОБИТ (WORK ITEMS).....	127
13.1. Визначення, властивості, життєвий цикл	127
13.2. ЗАСОБИ ВИКОРИСТАННЯ ЕЛЕМЕНТІВ РОБОТИ.....	133
Контрольні запитання	143
ГЛАВА 14. VSTS: КОНФІГУРАЦІЙНЕ УПРАВЛІННЯ.....	144
14.1. Загальні положення.....	144
14.2. СИСТЕМА КОНТРОЛЮ ВЕРСІЙ	144
14.3. АВТОМАТИЧНІ ЗБІРКИ	158
Контрольні запитання	170
ГЛАВА 15. VSTS: ТЕСТУВАННЯ.....	171
15.1. Загальні положення.....	171
15.2. СИСТЕМА ВІДСЛІДКУВАННЯ ПОМИЛОК	171
15.3. Модульні тести	177
15.4. ПАКЕТИ ТЕСТІВ	179
15.5. АВТОМАТИЧНЕ ТЕСТУВАННЯ Web-ДОДАТКІВ.....	182
Контрольні запитання	191
ГЛАВА 16. VSTS: ПІДТРИМКА РІЗНИХ МОДЕЛЕЙ ПРОЦЕСУ	192
16.1. Підтримка шаблонів процесу.....	192
16.2. ОГЛЯД ІСНУЮЧИХ ШАБЛОНІВ	195
16.3. MSF FOR AGILE SOFTWARE DEVELOPMENT.....	196
16.4. SCRUM.....	199
Контрольні запитання	201
ЛІТЕРАТУРА.....	202

ВСТУП

Програмна інженерія - наука про принципи і методології, що використовуються при розробці та супроводі програмних систем. Вона передбачає систематизований, упорядкований і конкретний підхід до розробки, експлуатації і супроводу програмного забезпечення, використання принципів інженерії по відношенню до процесу розробки програмного забезпечення.

Мета викладання дисципліни “Інженерія програмного забезпечення” полягає в поданні програмної інженерії у вигляді цілісного викладу, концентруючись на концепції процесу, на різних типах методологій розробки програмного забезпечення (CMMI, MSF, Scrum), окремих видах діяльності процесу – розробці архітектури, конфігураційному управлінні, роботі з вимогами, тестуванні. Осторонь залишені питання програмування, оскільки у межах загального курсу їх неможливо ефективно розглянути. Як програмні засоби, що підтримують цілісний процес розробки ПЗ, розглядається технологія компанії Microsoft - Visual Studio Team System (VSTS) з акцентом на Team Foundation Server (TFS). Показується, як викладений вище теоретичний матеріал можна реалізувати на практиці, з підтримкою програмних засобів розробки.

Посібник складається з довох розділів. Теоретичний матеріал, наданий в першому розділі, умовно може бути поділений на три частини. Перша частина присвячена питанням процесу розробки програмного забезпечення. Друга частина розглядає питання управління процесом розробки програмного забезпечення. Третя частина присвячена питанням використання діаграмної техніки в роботі із знаннями.

Мета другого розділу даного курсу - ознайомити студентів з програмними засобами, що підтримують цілісний процес розробки ПЗ. Детально розглядається технологія компанії Microsoft - Visual Studio Team System (VSTS) з акцентом на Team Foundation Server (TFS). Показується,

як викладений теоретичний матеріал можна реалізувати на практиці, з підтримкою програмних засобів розробки.

Матеріал другого розділу умовно може бути поділений на дві частини. Перша частина присвячена ІТ-рішенням для управління розробкою програмного забезпечення (ПЗ) з використанням різних методів розробки. Друга частина присвячена огляду технології Microsoft Visual Studio Team System (VSTS) для розробки ПЗ.

Теоретичний матеріал, що подається в посібнику, розрахований на студентів, які вперше вивчають цю дисципліну.

В кінці кожної глави посібника наведені контрольні запитання, які дозволять студентам перевірити ступінь засвоєння викладеного матеріалу.

В результаті вивчення дисципліни студенти навчаться створювати прикладне програмне забезпечення систем підтримки прийняття рішень, інформаційних систем, інтелектуальних систем, мультимедійних систем, різноманітних програмних продуктів для бізнесу, управляти вимогами, вмістом, термінами, вартістю і якістю, людськими ресурсами, ризиками при розробці програмного забезпечення, тестувати програмне забезпечення на модульному, інтеграційному і системному рівнях, розробляти проектну робочу технічну документацію.

Розділ 1. Основи програмної інженерії

Глава 1. Про предмет вивчення

Інженерія програмного забезпечення. Основні визначення: інформатика, системотехніка, бізнес-реінжиніринг. Програмне забезпечення: визначення, властивості.

1.1. Інженерія програмного забезпечення (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 гроші за виконану роботу. 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 передують аналіз (тобто створення функціональної моделі майбутньої системи без урахування реалізації, дозволяє усвідомити вимоги і очікування замовника) і проектування (тобто попередній макет, ескіз, план системи на папері). Трудовитрати на аналіз і проектування, а також форма представлення їх результатів сильно варіюються від видів проектів і переваг 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 швидкістю. 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 знань, або 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). Саме ці узагальнення складають програмну інженерію як галузь знань.

Необхідність в програмній інженерії, як у спеціальній галузі знань, була усвідомлена світовою спільнотою в кінці 60-х років минулого сто-