

МІЖРЕГІОНАЛЬНА  
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

**НАВЧАЛЬНА ПРОГРАМА**  
дисципліни  
**“МЕТОДОЛОГІЯ РОЗРОБКИ**  
**ПРОГРАМНИХ ПРОДУКТІВ**  
**І ВЕЛИКИХ ПРОГРАМНИХ СИСТЕМ”**  
(для спеціалістів)

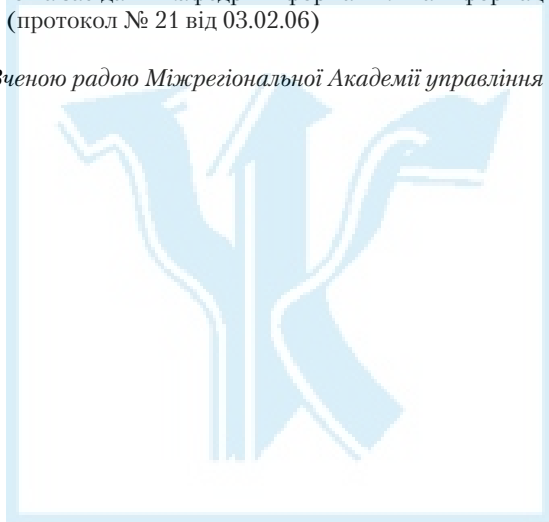
МАУП

Київ 2006

Підготовлено доцентом кафедри інформатики та інформаційних технологій *Ю. В. Бондарчуком*

Затверджено на засіданні кафедри інформатики та інформаційних технологій (протокол № 21 від 03.02.06)

*Схвалено Вченою радою Міжрегіональної Академії управління персоналом*



**Бондарчук Ю. В.** Навчальна програма дисципліни “Методологія розробки програмних продуктів і великих програмних систем” (для спеціалістів). — К.: МАУП, 2006. — 14 с.

Навчальна програма містить пояснювальну записку, тематичний план, зміст дисципліни “Методологія розробки програмних продуктів і великих програмних систем”, питання для самоконтролю, а також список літератури.

© Міжрегіональна Академія  
управління персоналом (МАУП),  
2006

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

Досвід розробки та супроводження великих програмних систем виявив ряд особливостей:

- розробка таких систем не під силу одному програмісту; отже потрібні знання і навички щодо методів групової розробки програм;
- визначення якості програмного забезпечення не збігається з інтуїтивною оцінкою розробника, тобто потрібні певні визнані але формальні методи;
- наявна проблема досягнення необхідної якості програмного забезпечення за обмежений час та обґрунтований бюджет;
- програмні об'єкти системи розробляються з можливістю їх повторного використання в інших проектах, що потребує ретельного спеціального продуманого документування та відслідковування різних версій об'єктів за їх одночасного використання;
- опис програмних систем у різних проекціях є самостійною проблемою і потребує як певної методики, так і спеціальних інструментів і навіть спеціально підготовлених фахівців;
- реальною є ситуація, коли супроводження складної системи виконується не безпосередніми її розробниками; тому всі етапи проекту мають бути настільки прозорими, щоб на будь-якому етапі (зокрема, супроводження) до роботи над/з системою міг би підключатися новий фахівець; це досягається в сучасній програмній інженерії використанням спеціальних метамов та інструментів вираження загальних інформаційних сутностей.

Предметом цієї дисципліни є вивчення сучасних інтегрованих середовищ розробки програмних продуктів, сучасних програмних технологій зв'язування окремо розроблених компонентів, стратегій розробки проектів (на прикладі RUP), моделювання інформаційних систем (на прикладі UML), CASE-засобів розробки (на прикладі Rational Rose), стратегій маркетингу програмних продуктів, методів оцінки витрат та економічної доцільності розробки програмних продуктів.

### **Мета вивчення дисципліни “Методологія розробки програмних продуктів і великих програмних систем”**

1. Чітке розуміння парадигми циклу розробки складних корпоративних і серійних програмних систем, зокрема важливість етапу проектування.
2. Опанування студентами основними сучасними засобами розробки програмних продуктів.

3. Мати ґрунтовну теоретичну базу у сфері сучасної програмної інженерії.
4. Опанування підходами до формування успішних стратегій їх розповсюдження та маркетингу.
5. Розуміти особливості маркетингу програмних продуктів.

### **Завдання дисципліни**

У результаті вивчення дисципліни студенти повинні:

1. Уміти класифікувати програмні системи за рівнем складності, оцінки вартості та трудомісткості.
2. Представляти опис систем в термінах мови моделювання UML.
3. Мати достатньо повне уявлення про особливості групової (командної) роботи над програмними проектами.
4. Набути практичних навичок роботи з будь-якою CASE-системою від проекту до прототипу системи.
5. Уміти оформлювати програмну документацію на всіх етапах циклу розробки та впровадження системи.

### **Місце дисципліни та її значення в навчальному процесі**

Професійна підготовка сучасного фахівця з розробки програмних систем вимагає не просто володіння знаннями та навичками з програмування, роботи з базами даних, опанування мережевими технологіями, а й чіткого розуміння сучасної програмної інженерії. Остання дає можливість в стислі строки розробляти програмні системи, які не під силу одному програмісту.

### **Міждисциплінарні зв'язки**

Цей курс ґрунтується на знаннях, отриманих студентами при вивченні дисципліни “Інформатика та комп'ютерна техніка”, “Програмування та алгоритмічні мови”, “Системи керування базами даних”, “Комп'ютерні мережі та телекомунікації”, “Методи прийняття управлінських рішень”, “Системи технологій” та ін. Вивчення дисципліни має завершувати у циклі дисциплін з інформатики та інформаційних технологій.

### **Передбачені форми контролю знань**

Під час вивчення курсу передбачається практична робота студентів за комп'ютерами, в тому числі в мережах, як під керівництвом викладача, так і самостійно.

Передбачено постійний контроль у процесі вивчення дисципліни (опитування на лекціях, виконання практичних завдань) та періодичний (контроль знань за кожним модулем, іспит за дисципліною).

**ТЕМАТИЧНИЙ ПЛАН**  
**дисципліни**

**“МЕТОДОЛОГІЯ РОЗРОБКИ ПРОГРАМНИХ ПРОДУКТІВ  
І ВЕЛИКИХ ПРОГРАМНИХ СИСТЕМ”**

№ пор.	Назва змістового модуля і теми
	<b>Змістовий модуль 1. Загальні питання розробки великих програмних систем</b>
1	Сучасна парадигма циклу розробки програмного продукту
2	Технології розробки програмних продуктів
	<b>Змістовий модуль 2. Моделювання, проектування та супровід великих програмних систем</b>
3	Методології моделювання та проектування великих програмних систем
4	Особливості менеджменту розробок і супроводу програмних систем. Маркетинг програмних продуктів
Разом годин: 108	

**ЗМІСТ**  
**дисципліни**

**“МЕТОДОЛОГІЯ РОЗРОБКИ ПРОГРАМНИХ ПРОДУКТІВ  
І ВЕЛИКИХ ПРОГРАМНИХ СИСТЕМ”**

**Змістовий модуль 1. Загальні питання розробки великих програмних систем**

**Тема 1. Сучасна парадигма циклу розробки програмного продукту**

Програмна інженерія як дисципліна. Вивчення основних етапів повного циклу розробки програмних продуктів — дослідження проблеми, визначення економічної доцільності, проектування структури системи, розподіл на компоненти, розробка компонентів та їх безпосереднє програмування і верифікація, інтеграція компонентів у єдиний продукт, тестування, розробка документації та валідація, впровадження у замовника або продаж на ринку, підтримка користувачів. Розробка та підтримка нових версій продукту.

Основні типи програмних продуктів щодо їх розміру, цільового призначення та способу розповсюдження. Програми для одного користувача. Коробкові продукти. Корпоративні продукти. Web-орієнтовані системи. Системи, що розробляються за окремим замовленням та серійні продукти.

Типові задачі, що вирішуються корпоративними Web-орієнтованими системами. Вимоги до продуктів, що використовуються на Web-серверах. Потенціал Web-систем, проблеми їх масштабування. Врахування проблеми Internet-хакінгу на стадії проектування системи.

*Література* [1–4; 10; 14; 15]

## ***Тема 2. Технології розробки програмних продуктів***

Огляд поширених об'єктних технологій: COM, DCOM, .Net, JavaBeans.

Ознайомлення з принципами Net-об'єктної технології. Керування пам'яттю у .Net. Порівняння .Net з її попередниками. Спільне використання COM- і Net-об'єктів в одному середовищі. Особливості інсталяції Net-об'єктів. Керування версіями Net-об'єктів, серіалізація об'єктів.

Особливості вибору інтегрованих середовищ розробки програмних систем. Розробка настольних програм, орієнтованих на одного користувача. Знаходження помилок і тестування програмних продуктів. Особливості розробки великих програмних систем, що складаються з багатьох незалежних компонентів. Особливості колективної розробки.

Середовища, продукти і засоби, що найчастіше використовуються при розробці коробкових програмних продуктів.

CASE-системи. Зв'язок .Net із CASE-системами.

Роль і місце баз даних у сучасних програмних комплексах. Порівняння малих “настольних” баз даних із корпоративними серверними базами даних. Огляд MS SQL Server. Механізми транзакцій. Універсальні інтерфейси доступу до баз даних. Технологія MS ADO. Зв'язок з CASE-системами проектування. Застосування баз даних у Web-орієнтованих системах.

*Література* [1; 5–9; 11–17]

## **Змістовий модуль 2. Моделювання, проектування та супровід великих програмних систем**

### ***Тема 3. Методології моделювання та проектування великих програмних систем***

Важливість створення моделей інформаційних систем, що розробляються. Огляд методів моделювання в історичному контексті. Універсальна мова моделювання UML. Залежність типу моделі від масштабів проекту і типу програмної системи, що розробляється.

Огляд методів проектування і розробки програмних систем в історичному контексті. Rational Unified Process (RUP) як сучасна і одна з найбільш стандартизованих методологій розробки програмних систем. Ітераційний підхід порівнянно з підходом типу “водоспаду”. RUP як технологія і як продукт.

CASE-засоби і особливості їх застосування. CASE-система Rational Rose. Застосування UML у Rational Rose. Аналіз системи, що проектується у різних проєкціях. Автоматизація створення проєктної документації. Зв'язок Rational Rose з інтегрованими середовищами розробки.

*Література* [1; 6; 8–10; 13; 14]

### ***Тема 4. Особливості менеджменту розробок і супроводу програмних систем. Маркетинг програмних продуктів***

Специфічні ознаки, властиві корпоративним системам. Середовища, продукти і засоби, що найчастіше використовуються при розробці корпоративних систем. Співробітництво з організацією замовником, починаючи з ранніх етапів розробки. Особливості впровадження і підтримки великих систем.

Оцінювання часу і розміру колективу, що необхідні для розробки окремих компонентів залежно від інтегрального середовища розробки, що застосовується. Оцінювання затрат на інтеграцію компонентів у єдиний продукт. Визначення необхідних матеріальних затрат. Остаточне визначення економічної доцільності проєкту.

Особливості управління колективами, що розробляють програмні системи. Підбір кадрів. Розподіл завдань між розробниками згідно з компонентною структурою системи. Контроль за процесом розробки і звітність згідно з рекомендаціями RUP.

Огляд особливостей програмних продуктів порівняно з іншими видами товарів. Стратегії маркетингу програмних продуктів. Особливості Інтернет-маркетингу. Захист програмних продуктів та боротьба із софт-піратством.

Маркетингові дослідження та оцінка економічної доцільності розробки коробкових програмних продуктів. Організація підтримки користувачів.

*Література [1–3; 10; 15–17]*

### **ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ**

1. Якими питаннями займається програмна інженерія?
2. Особливості у розробці великих програмних проєктів.
3. Основні етапи повного циклу розробки програмного продукту.
4. Які завдання вирішуються на етапі дослідження проблеми?
5. Що враховується при визначенні економічної доцільності розробки програмного продукту?
6. Що потрібно робити при визначенні структури системи, що проєктується?
7. Чи можливо не лише інтуїтивне визначення якості програмної системи?
8. Чи можливо керувати якістю програмного продукту?
9. Що та які спеціалісти роблять під час безпосереднього програмування компонентів?
10. Як і ким здійснюється інтеграція окремих компонентів в єдиний продукт?
11. Що означає термін верифікація системи (проєкту)?
12. Які завдання вирішуються під час тестування програмних продуктів?
13. Охарактеризуйте основні джерела помилок в ході розробки системи.
14. Які вам відомі методи тестування систем?
15. Для чого використовують засоби документування результатів тестування?



16. Сутність валідації системи.
17. Яка роль відводиться питанню повторного використання об'єктів (систем) у програмній інженерії?
18. Які є типи програмних продуктів з точки зору їх розміру?
19. Які є типи програмних продуктів з точки зору способу їх розповсюдження?
20. Чи завжди розробка настольного продукту потребує меншої кваліфікації, ніж корпоративного? Обґрунтуйте.
21. Що таке корпоративні продукти? Опишіть основні риси таких продуктів.
22. Особливості систем, що розробляються за окремим замовленням, порівняно із серійними системами?
23. Що таке Web-орієнтовані системи?
24. Які завдання можуть вирішуватися за допомогою Web-орієнтованих систем?
25. Особливості впровадження Web-орієнтованих систем.
26. Проблеми захисту інформації у Web-орієнтованих системах і як вони вирішуються.
27. Як розроблялися програмні продукти до появи об'єктно-орієнтованих методів розробки?
28. Найпоширеніші зараз об'єктно-орієнтовані промислові технології.
29. Опишіть основні ознаки Net-технології.
30. Як використовувати Net-об'єкти разом з СОМ об'єктами?
31. Особливості інсталяції Net-об'єктів.
32. Як здійснюється контроль за версіями Net-об'єктів?
33. Навіщо потрібні інтегральні середовища розробки?
34. Найпоширеніші промислові інтегральні середовища розробки.
35. Опишіть основні риси інтегрального середовища, з яким ви працюєте.
36. Дайте порівняння VB.Net (C#.Net, C++.Net тощо) із іншими інтегральними середовищами розробки.
37. Як створити проєкт і скомпілювати модуль, що виконується, у середовищі VB.Net (C#.Net, C++.Net тощо)?
38. Як виявляти помилки у програмі за допомогою середовища VB.Net (C#.Net, C++.Net тощо)?
39. Зв'язок інтегральних середовищ розробки із CASE-системами.
40. Роль і місце баз даних у сучасних програмних системах.

41. Чи існують типи систем, які не потребують баз даних, якщо “так”, які саме типи?
42. Які моделі клієнт/сервер ви знаєте?
43. Що таке інтегральна цілісність даних і якими засобами вона забезпечується?
44. Що таке транзакції і навіщо вони потрібні?
45. У чому проблема масштабування програмних систем, зокрема баз даних?
46. Охарактеризуйте універсальний інтерфейс доступу до баз даних MS ADO і як він працює?
47. Охарактеризуйте корпоративну систему керування базами даних, з якою ви працюєте чи вивчаєте (зокрема, комп'ютерна платформа, модель даних, інтерфейси доступу, системи-оболонки розробки клієнтських програм, журналізація подій, питання масштабування баз тощо).
48. Що таке моделювання інформаційної системи? Його особливості.
49. Навіщо потрібно моделювання інформаційної системи та які проблеми вирішуються шляхом моделювання?
50. Які є сучасні методи моделювання?
51. Які основні риси мови моделювання UML?
52. Як в UML моделюється поведінка системи?
53. Хто ще, окрім розробників програмних систем, може використовувати мову UML і для чого?
54. Яким чином опис мовою UML залежить від типу програмної системи?
55. Які існують зараз та існували раніше технології розробки програмних систем?
56. Навіщо потрібно дотримуватися певної стандартизованої технології при розробці програмних систем?
57. Що таке технологія RUP і які її основні риси?
58. Основні етапи розробки згідно з RUP.
59. Основні технологічні процеси згідно з RUP.
60. Порівняйте підхід “водоспаду” та ітераційний підхід.
61. Порівняйте RUP як продукт і як технологію.
62. Які є сучасні CASE засоби для розробки програмних систем і на яких технологіях вони базуються?
63. Як виконується трансформація проекту у програмну систему?

64. Що таке Rational Rose і які основні ознаки цього продукту?
65. Як за допомогою Rational Rose створити макет системи у VB.Net (C#.Net, C++.Net тощо)?
66. Як поєднані між собою Rational Rose і мова моделювання UML?
67. Як риси програмного продукту як товару?
68. Чим суттєво відрізняється програмний продукт від інших видів товарів?
69. Що таке коробкові програмні продукти?
70. Як визначити економічну доцільність при розробці коробкового програмного продукту?
71. Які середовища найчастіше використовуються при розробці коробкових програмних продуктів?
72. Особливості розповсюдження коробкових програмних продуктів.
73. Яка повинна бути документація до коробкових програмних продуктів?
74. Особливість корпоративних програмних продуктів?
75. Як визначити економічну доцільність при розробці корпоративного програмного продукту?
76. Які середовища найчастіше використовуються при розробці корпоративних програмних продуктів?
77. Особливості впровадження корпоративних програмних продуктів.
78. Яка повинна бути документація до корпоративних програмних продуктів?
79. Як організується взаємодія між розробниками та замовниками при створенні корпоративних програмних продуктів?
80. Які ви можете навести рекомендації щодо підбору складу колективу розробників програмної системи?
81. Як оцінюються витрати на розробку окремих компонентів програмної системи?
82. Як оцінюються витрати на інтеграцію програмної системи з іншими системами (зокрема, власного виготовлення та сторонніх компаній)?
83. Як визначаються матеріальні витрати при розробці програмної системи (наведіть кілька підходів)?
84. Як визначається загальна економічна доцільність проекту по розробці програмної системи?

85. Сутність маркетингу програмних продуктів.
86. Особливості Інтернет маркетингу програмних продуктів.
87. Як здійснюється підбір кадрів про розробці програмного проекту?
88. Як правильно розподіляти роботу між виконавцями згідно з компонентною структурою системи?
89. Як контролювати виконання етапів робіт?
90. Яка звітність рекомендована RUP у процесі розробки програмного проекту і як її використовувати?

## **СПИСОК ЛІТЕРАТУРИ**

### *Основна*

1. *Бабенко Л. П., Лаврищева К. М.* Основи програмної інженерії. — К.: Знання, 2001. — 269 с.
2. *Калянов Г. Н.* CASE-технологии. Консалтинг при автоматизации бизнес-процессов. — М.: Горячая линия-Телеком, 2000. — 320 с.
3. *Новоженев Ю. В.* Объектно-ориентированные технологии разработки сложных программных систем. — М., 1996.
4. *Маклаков С. В.* Case-средства разработки информационных систем. — М.: Диалог-Мифи, 2001. — 304 с.
5. *Платт Д. С.* Знакомство с Microsoft. Net. — М.: Издат.-торг. дом “Русская редакция”, 2001. — 240 с.
6. *Троелсен Э.* С# и платформа .Net. Библиотека программиста. — СПб.: Питер, 2004. — 796 с.
7. *Мамаев Е. В.* MS SQL Server 7.0: Проектирование и реализация баз данных. — СПб.: БХВ-Санкт-Петербург, 2000. — 416 с.
8. *Крачтен Ф.* Введение в Rational Unified Process. — М.: Издат. дом “Вильямс”, 2002. — 240 с.
9. *Буч Г., Рамбо Д., Джекобсон А.* Язык UML. Руководство пользователя. — М.: ДМК Пресс, 2001. — 432 с.
10. *Державний стандарт України.* Основні напрямки оцінювання та відбору CASE-інструментів. ДСТУ 3919–1999. 2000 р.

### *Додаткова*

11. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++. — СПб.: Невский диалект. — 1999.
12. *Фронковьяк Д., Гарсиа М., Уолен Э.* Руководство администратора MS SQL Server. — М.: Издат.-торг. дом “Русская редакция”, 2000. — 672 с.

13. *Microsoft Corporation*. Разработка распределенных приложений на Microsoft Visual Basic 6.0. — М.: Издат.-торг. дом “Русская редакция”, 2000. — 400 с.
14. *Профимов С. А.* CASE0-технологии. Практическая работа в Rational Rose. — М.: Бином, 2002. — 284 с.
15. *Клоков Ю. К., Папушин В. К., Хамитов Р. Р.* Методы повышения надежности программного обеспечения // Зарубежная радиоэлектроника. — 1984. — № 6. — С. 3–22.
16. *Интернет-сайт* <http://www.rational.com>
17. *Интернет-сайт* <http://www.idef.com>



## ***ЗМІСТ***

Пояснювальна записка .....	3
Тематичний план дисципліни “Методологія розробки програмних продуктів і великих програмних систем” .....	5
Зміст дисципліни “Методологія розробки програмних продуктів і великих програмних систем” .....	5
Питання для самоконтролю.....	8
Список літератури.....	12



**МАУП**

Зам. № ВКЦ-2640

Міжрегіональна Академія управління персоналом (МАУП)  
03039 Київ-39, вул. Фрометівська, 2, МАУП