

ІНФОРМАТИКА

УДК 004.651.4(045)

В.А. Бородин, к.т.н., доц.*Київський національний університет ім. Т.Г. Шевченка***С.М. Креденцар, к.т.н., доц.***Національний авіаційний університет***ЗАСТОСУВАННЯ МЕТОДОЛОГІЇ RATIONAL UNIFIED PROCESS
ДЛЯ ПОБУДОВИ АЕРОНАВІГАЦІЙНИХ СИСТЕМ**

Запропоновано застосування методології Rational Unified Process (RUP) при побудові програмного забезпечення для авіаційних комплексів та систем. Такий підхід виявився корисним для структуризації та оптимізації робіт по створенню інтерактивних авіаційних комплексів.

Вступ. Сучасні служби спостереження за повітряною обстановкою в аеропортах вимагають все більшої функціональності для обробки інформації, що стає наслідком розвитку обчислювальної техніки та збільшення кількості повітряних суден. У багатьох прикладних аеронавігаційних системах, наприклад, в комплексах, які спостерігають за літаками в аеропортах, вводяться все більше автоматизованих функцій для контролю за інформацією, для передбачення дій об'єктів, за якими спостерігають, для відображення географічної інформації та ін. [1, 2].

Таким чином, програмне забезпечення таких систем стає достатньо складним і його розробка стає варто- та часомістким процесом, що вимагає застосування спеціальних методологій для його розробки, впровадження та використання. Використання однієї з найпопулярніших таких методологій – методології RUP для розробки програмного забезпечення (ПЗ) аеронавігаційних систем розглянуто в даній роботі.

Аналіз останніх досліджень показав, що на сьогоднішній день існує достатня кількість розвинених технологій створення ПЗ, які є добре стандартизовані та практично опробовані. Провідною методологією, у якій інструментально підтримуються всі етапи життєвого циклу розробки ПЗ, є методологія Rational Unified Process (RUP). RUP – програмний продукт, розроблений компанією Rational Software. RUP пропонує ітеративну модель розробки, що містить чотири фази:

- початок;
- розробка;
- побудова;
- впровадження.

Кожна фаза може бути розбита на етапи (ітерації), у результаті яких випускається версія для внутрішнього або зовнішнього використання. Проходження крізь чотири основні фази називається циклом розробки, кожний цикл завершується генерацією версії системи. Якщо після цього робота над проектом не припиняється, то отриманий продукт продовжує розвиватися й знову пройде ті ж фази. Сутність роботи в рамках RUP – це створення й супровід моделей на базі Universal Modelling Language (UML).

В основі методології RUP, як і багатьох інших програмних методологій, що поєднують інженерні методи створення ПЗ, лежить “покроковий підхід”. Він визначає етапи життєвого циклу, контрольні точки, правила робіт для кожного етапу й, тим самим, упорядковує проектування й розробку ПЗ. Для кожного етапу життєвого циклу методологія задає:

- склад і послідовність робіт, а також правила їхнього виконання;
- розподіл повноважень серед учасників проекту (ролі);
- склад і шаблони формованих проміжних і підсумкових документів;
- порядок контролю й перевірки якості.

У сучасній літературі, особливо вітчизняній, розробці методології побудови ПЗ для авіації присвячена незначна кількість робіт [2–7]. Більшість праць, що вивчають життєвий цикл таких програмних комплексів розглядаються у відриві від практичних методологій супроводження таких програмних систем. Разом із тим, потреба у використанні таких методологій є вже відомим та практично доведеним фактом. У даній статті представлена спроба усунення цього протиріччя.

Метою даної роботи є побудова життєвого циклу ПЗ аеронавігаційних систем на базі стандартів RUP методології.

Викладення основного матеріалу. Rational Unified Process – це процес моделювання й побудови ПЗ із об'єктів із застосуванням мови UML [4]. Він містить теоретичний і прикладний аспекти подання й тлумачення створюваних моделей для проектованої предметної області.

Теоретичний аспект процесу моделювання моделей ПЗ підтримується методами і поняттями формальних теорій. Формалізація моделей у RUP забезпечується засобами UML і дає можливість строго описувати вимоги й перетворювати їх до готового продукту [7].

Основу процесу моделювання становлять прецеденти – варіанти використання для визначення вимог до системи. Головний елемент проектування – модель варіантів використання, на основі якої розробляють моделі аналізу, проектування й реалізації системи. Кожну модель аналізують на відповідність моделі варіантів використання, до якої належить вхідні дані для пошуку й специфікації класів і підсистем, для підбору й специфікації тестів, а також при плануванні ітерацій розробки й інтеграції ПЗ. У процесі моделювання створюється така модель:

- варіантів використання, що відображають взаємодію між користувачами та ПЗ;
- аналізу, яка забезпечує специфікації вимог до системи й опис варіантів використання як взаємодії між концептуальними класифікаторами;
- проектування, орієнтованого на створення статичної структури й інтерфейсів системи, реалізацію варіантів використання у вигляді набору кооперацій (взаємодій) між підсистемами, класами й інтерфейсами;
- реалізації, яка містить компоненти системи у вихідному вигляді на мові програмування;
- тестування;
- розміщення компонентів і виконання в операційному середовищі комп'ютерів.

Ці моделі представляють різними видами діаграм, наприклад, у моделі варіантів використання діаграми use-case, у моделях аналізу – діаграми класів, кооперацій і станів. Дані моделі – взаємозалежні, вони семантично перетинаються й визначають систему як єдине ціле. Наприклад, варіант використання у відповідній моделі може мати відношення залежності до кооперації в моделі проектування, що задає реалізацію. Моделі на кожній ітерації процесу RUP уточнюють або розширюють моделі попередніх ітерацій процесу. Наприклад, модель аналізу складається з діаграм класів, станів і кооперацій.

Основу моделі аналізу становить діаграма варіантів використання, що містить у собі діаграми класів, взаємодії, що задають можливі сценарії варіантів використання в термінах взаємодії об'єктів на етапі аналізу.

Варіанти використання специфікують тип відносин між діючою особою (актором), користувачем і системою. На високому рівні абстракції вони представляються впорядкованою послідовністю дій або альтернатив.

Варіант використання в UML – це різновид класифікатора, операціями якого є повідомлення, що одержуються екземплярами конкретного варіанта використання. Методи задають реалізацію операцій у термінах послідовностей дій, виконуваних екземплярами варіанта використання.

Із практичної точки зору RUP – це впорядкований набір кроків і етапів ЖЦ, що виконуються ітеративно. Цей процес є керованим як у змісті завдання вимог, так і в реалізації функціональних можливостей ПЗ із заданим рівнем якості й гарантованих витрат згідно з графіком робіт. Оцінка якості всіх кроків і дій учасників процесу базується на певних критеріях.

Кроки при виконанні RUP управляються прецедентами, тобто технологічним маршрутом від ділового моделювання й вимог до випробувань. Екземпляр прецеденту – це послідовність дій, виконуваних системою з спостережуваним результатом для конкретного суб'єкта. Функціональні можливості системи визначаються набором прецедентів, кожний з яких являє собою деякий потік подій. Опис прецеденту визначає те, що відбудеться в системі, коли прецедент буде виконаний. Кожний прецедент орієнтований на завдання, яке він повинен виконати. Набір прецедентів встановлює всі можливі шляхи (маршрути) виконання системи. Прецеденти відіграють роль у кожному з п'яти основних робіт процесу: формуванні вимог, аналізу, проектуванні, реалізації й випробуванні.

RUP містить п'ять основних етапів, виконуваних на всіх фазах процесу розробки ПЗ. Завершення цих етапів називається ітерацією, при якій закінчується випуск проміжного продукту. На кожній ітерації цикл повторюється, починаючи зі збору й уточнення вимог.

RUP використовує ітеративну модель розробки. Наприкінці кожної ітерації (яка в ідеалі триває від 2 до 6 тижнів) проектна команда повинна досягти запланованих на дану ітерацію цілей, створити або доробити проектні артефакти й одержати проміжну, але функціональну версію кінцевого продукту. Ітеративна розробка дозволяє швидко реагувати на мінливі вимоги, виявляти й усувати ризики на ранніх стадіях проекту, а також ефективно контролювати якість створюваного продукту.

Створення ПЗ аеронавігаційної системи.

Для опису проекту в розглядуваній методології використовуються стандартний шаблон опису, що допомагає уніфікувати процес розробки та документування.

1. Назва: планування.

2. Контекст використання: розробка плану робіт, контрольних точок. Результат: опис архітектури та модулів системи.

3. Область дії: аеронавігаційні системи.

4. Рівень: узагальнений.

Після опису цих чотирьох пунктів переходять до описання акторів та їх взаємодії.

Основна діюча особа: КБ розробки системи (розробник).

Учасники й інтереси: стейкхолдери системи – Украерорух, аеропорт.

Передумова: технічні засоби, що містять засоби отримання інформації про повітряну ситуацію, блок відображення, обробки та зберігання інформації.

Мінімальні гарантії: у випадку неузгодження проекту буде повторений цикл розробки.

Гарантії успіху: підтвердження про узгодження робіт – перехід на наступний рівень.

Тригер: початок робіт виникає при надходженні замовлення.

Основний сценарій:

1. Отримання вказівок від стейкхолдерів.
2. Уточнення технічних параметрів системи.
3. Розробка вимог до ПЗ.
4. Опис функцій програмного забезпечення.
5. Затвердження запропонованих функцій.

Розширення:

1. Отримання вказівок – неузгодженість вказівок – повторне одночасне узгодження.
2. Уточнення технічних параметрів системи – недостатність технічних характеристик стосовно вимог – узгодження з замовником: або збільшення технічних можливостей, або корегування вимог (крок 1).

Так виглядає планування загального сценарію розробки програмного забезпечення аеронавігаційної системи.

Наступний крок розробки є більш складним оскільки містить вже окремі плани розробки кожного зі складових системи.

Зокрема, розглянемо розробку модуля системи відображення.

1. Назва: розробка системи відображення.

2. Контекст використання: програмний модуль відображення.

3. Область дії: відображення повітряної обстановки на екрані аеронавігаційної системи.

4. Рівень: під функція.

Після опису цих чотирьох пунктів переходять до описання акторів та їх взаємодії.

Основна діюча особа: програміст системи відображення (девелопер).

Учасники й інтереси: модуль обробки інформації, екран відображення, користувач системи.

Передумова: наявність блоку зберігання відображень, фіксований формат зображень, швидкодія системи.

Мінімальні гарантії: у випадку невиконання проекту буде повторений цикл розробки.

Гарантії успіху: підтверджене випробування модуля.

Тригер: початок робіт виникає після закінчення написання модуля зберігання зображень та обробки інформації.

Основний сценарій:

1. Написання функції відображення статичних об'єктів на моніторі.
2. Написання функції відображення динамічних об'єктів на моніторі.
3. Написання функції відображення формулярів об'єктів.
4. Написання функції отримання вказівок користувача.
5. Написання функції відображення змін динамічної ситуації.
6. Тестування написаних функцій.

Для розробки інтерфейсу стандарт передбачає також запис варіантів використання функцій. Сценарії варіантів використання зручно подати у таблицю, як це показано в таблиці 1. Інформація при цьому приймає більш структурований вигляд.

Таблиця 1

Сценарії варіантів використання

Актор	Дія
Система	Відображає повітряну обстановку
Користувач	Робить певне наближення
Система	Відображає наближення
Користувач	Вибирає потрібний об'єкт
Система	Показує докладну інформацію про об'єкт

Висновки. У цій роботі запропоновано методологію Rational Unified Process (RUP) для створення програмного забезпечення аеронавігаційних комплексів та систем. Ця методика важлива в плані інтеграції програмного забезпечення в комплексах реального часу та аеронавігаційних системах.

Дану схему методології програмного забезпечення можна також використовувати для створення інших програмних систем в авіаційних комплексах.

ЛІТЕРАТУРА:

1. An object-oriented framework for the integration of interactive animation techniques / R.C. Zeleznik, D.B. Conner, M.M. Wloka and all. – Computer Graphics (SIGGRAPH'91 Proceedings). – 25. – July, 1991. – Pp. 105–112.
2. Бородин В.А. Методы и средства представления и анализа динамической обстановки в геоинформационных комплексах реального времени : дис. ... канд. техн. наук : 05.13.06 / В.А. Бородин. – К., 2005. – 140 с.
3. Бородин В.А. Структура програмних засобів відображення та аналізу динамічних сцен в геоінформаційних комплексах реального часу / В.А. Бородин // Збірник наукових праць НДЦ ОТ і ВБУ. – № 20. – К. : НДЦ ОТ і ВБУ, 2003. – С. 88–95.
4. Брауде Э. Технологии разработки программного обеспечения / Э.Брауде. – СПб. : Питер, 2004. – 655 с.
5. Вельбицкий И.В. Технология программирования / И.В. Вельбицкий. – К. : Техника, 1984. – 279 с.
6. Якобсон А. Унифицированный процесс разработки программного обеспечения / А.Якобсон, Г.Буч, Дж.Рамбо. – СПб. : Питер, 2002. – 496 с.
7. Analyzing requirements and defining Microsoft. Net solution architectures. – Microsoft Press 2000. – 491 p.

БОРОДИН Віктор Анатолійович – кандидат технічних наук, доцент кафедри Аеронавігаційних систем Інституту аеронавігації Київського національного університету ім. Т.Г. Шевченка.

Наукові інтереси:

- оперативні комплекси відображення поточної обстановки на екранах колективного користування в режимі реального часу.

КРЕДЕНЦАР Світлана Максимівна – кандидат технічних наук, доцент кафедри Аеронавігаційних систем Інституту аеронавігації Національного авіаційного університету.

Наукові інтереси:

- аеронавігаційні геоінформаційні системи реального часу;
- методи побудови геоінформаційних систем відображення поточної обстановки на екранах колективного користування в режимі реального часу.

Подано 26.10.2011

