

УДК 681.5

А.Б. Дымо, аспирант кафедры ТТ и СПУ

А.С. Морозова, преподаватель кафедры ИУСТ

ОТКРЫТАЯ МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ ПРОЕКТОВ

Постановка проблемы

Последние несколько лет открытая модель разработки (также известная как “Open Source Model” - “Модель открытых исходников”) все чаще используется предприятиями, разрабатывающими программное обеспечение по всему миру. Возникнув как модель разработки свободного программного обеспечения, она, не теряя своего значения для свободных проектов, приобрела неожиданную ценность и для коммерческого мира. Основными факторами, определяющими ее преимущество по сравнению с другими моделями можно назвать:

- уменьшение стоимости кодирования и тестирования;
- увеличение качества производимых продуктов;
- значительная гибкость жизненного цикла;
- расширенные возможности управления рисками.

Анализ исследований

До сих пор не разработано сколько-нибудь детальной модели жизненного цикла открытых продуктов. Имеющиеся исследования в области моделей жизненных циклов либо не упоминают открытую модель вообще, либо не рекомендуют ее. Так, например, Уокер Ройс, основоположник итерационной модели разработки, похожей в некоторых отношениях на открытую модель, говорит о невозможности коммерческого применения открытой модели [2]. В изданиях американского института качества программного

обеспечения упоминается об открытой и свободной модели как о возможных моделях [3], однако не приводятся никакие данные о них. Иными словами, открытая модель жизненного цикла программного обеспечения, эффективность которой подтверждена на практике, не нашла должного внимания в теоретических исследованиях.

Постановка задачи

Задачами данного исследования есть:

1. разработка теоретических основ открытой модели жизненного цикла программных проектов;
2. выделение составляющих жизненный цикл фаз модели и определение самой модели;
3. обозначение места и продолжительности этапов разработки, не входящих непосредственно в модель, однако необходимых для выполнения;
4. определение соответствия фаз и этапов разработки модели процессам, определенным в стандартах ISO/IEC 12207 и ДСТУ 3918-1999 на жизненные циклы ПО.

Основной материал

Итак, что же представляет собой открытая модель жизненного цикла. Как показывает практика разработки открытых и свободных проектов, все подобные проекты имеют циклический характер разработки. Цикличность обычно фиксируется в соответствующих документах [4].

Вышеозначенный циклический процесс обычно состоит из четырех основных фаз:

- 1) определения целей, альтернатив и ограничений, когда производится определение сущности проекта и выбора общего вектора всех процессов разработки;
- 2) оценки альтернатив, идентификации и разрешения рисков, когда

- производится настройка и оптимизация инфраструктуры и макропроцесса проекта, а также выполняется прототипирование;
- 3) разработки продукта, когда выполняются необходимые этапы проектирования и кодирования;
- 4) внедрения, сопровождения и планирования следующей фазы.

Означенные выше фазы, а также их характеристики четко указывают на то, что открытая модель разработки ПО является развитием спиральной модели жизненного цикла, предложенной Барри Бозмом.

Графическое представление модели показано на рис. 1. Предложенное представление отражает все основные особенности открытой модели, включая интенсивное использование прототипирования, открытых компонентов и открытость процесса разработки для других сторон.

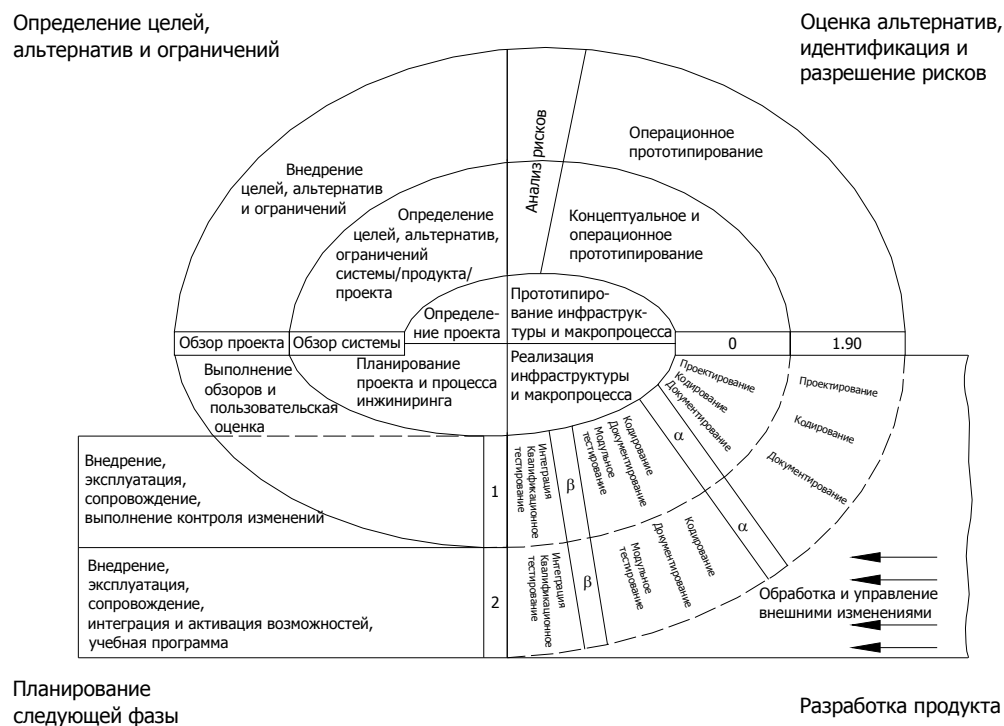


Рис. 1. Открытая модель

Фазы модели идентифицируются следующим образом.

1. Оценка альтернатив, идентификация и разрешение рисков
Оценивание альтернатив, связанных с целями и ограничениями;

идентификация и устранение рисков.

2. Разработка продукта

Создание продукта (детальное проектирование и кодирование) на основании информации, полученной на предыдущей фазе.

3. Планирование следующей фазы

Использование информации, полученной на этапе следующего уровня продукта с целью выполнения перехода к планированию на следующем проходе спирали.

Детальная идентификация процессов, действий и задач модели приведена в таблице 1.

Таблица 1. Процессы, действия и потенциальные задачи, выполняемые в рамках открытой модели жизненного цикла ПО

| Процессы | |
|--|---|
| Действия | Задачи |
| 1. <u>Определение проекта</u> – разработка предварительной цели и плана проекта, с помощью которых приближенно описываются графики и поставляемые продукты | |
| начало выполнения проекта | распределение ресурсов проекта |
| | управление планом проекта |
| исследование концепции | идентификация идей либо потребностей |
| | формулирование потенциальных подходов к удовлетворению потребностей |
| 2. Прототипирование инфраструктуры и макропроцесса | |

| Процессы | |
|--|--|
| Действия | Задачи |
| прототипирование среды проекта | обзор сред конфигурирования проекта |
| | установка и апробация сред конфигурирования проекта |
| 3. Реализация инфраструктуры и макропроцесса | |
| установка среды проекта | выбор и установка среды конфигурирования проекта |
| создание прототипа системы | реализация демонстрационного прототипа, реализующего основные системные концепции на установленной среде проекта |
| 4. Планирование проекта и процесса инжиниринга | |
| повторное планирование управления проектом | повторное формулирование потенциальных подходов |
| | уточнение идей либо потребностей |
| приемка среды проекта | апробация демонстрационного прототипа, реализующего основные системные концепции |
| 5. Определение целей, альтернатив и ограничений системы/продукта/проекта | |
| системное распределение | анализ функций на уровне системы/продукта |
| | разработка системной архитектуры |

| Процессы | |
|--|---|
| Действия | Задачи |
| специфицирование требований к продукту | уточнение и разработка требований к ПО |
| | определение требований к интерфейсу |
| 6. Анализ рисков | |
| идентификация рисков | определение источников возникновения рисков |
| | идентификация потенциальных рисковых событий |
| 7. Концептуальное и операционное прототипирование – уточнение требований и спецификаций для потенциально наиболее критичных частей системы с целью изучения степени выполнимости этих частей | |
| изучение выполнимости | выполнение имитаций |
| | создание и выполнение сравнительных тестов |
| исследование критичных частей системы | эскизное проектирование критичных частей продукта |
| | прототипирование и реализация критичных частей продукта |
| идентификация дальнейших требований к продукту | уточнение и дальнейшая разработка требований |
| | определение требований к интерфейсу |
| 8. Разработка продукта. Одновременное проектирование, кодирование и документирование | |
| проектирование | разработка проекта архитектуры |
| | разработка проекта интерфейса |
| | разработка проекта базы данных |

| Процессы | |
|--|---|
| Действия | Задачи |
| кодирование | создание исходного кода |
| | генерирование объектного кода |
| документирование | генерация оперативной проектной документации |
| 9. Разработка продукта. Одновременное кодирование, документирование и модульное тестирование | |
| кодирование | создание исходного кода |
| | генерирование объектного кода |
| документирование | генерация оперативной проектной документации |
| модульное тестирование | разработка и сопровождение плана тестирования |
| | разработка тестовых требований |
| | создание тестовых данных |
| | выполнение тестов |
| 10. Разработка продукта. Интеграция и квалификационное тестирование | |
| интеграция | разработка и сопровождение плана интеграции |
| | выполнение интеграции |
| квалификационное тестирование | разработка и сопровождение плана тестирования |
| | выполнение тестов |
| 11. Обработка и управление внешними изменениями | |

| Процессы | |
|---|--|
| Действия | Задачи |
| управление внешними изменениями | создание и сопровождение документов, регламентирующих возможность и процедуру внесения внешних изменений |
| | управление и отслеживание внесенных изменений |
| обработка внешних изменений | отбор изменений согласно критериям качества |
| | интеграция изменений |
| | корректировка изменений |
| 12. Выполнение обзоров и пользовательская оценка | |
| анализ структуры системы | уточнение требований к архитектуре системы |
| идентификация исправленных требований | проведение пользовательской оценки |
| | управление результатами пользовательской оценки |
| | управление базой данных запросов на изменения и исправления |
| 13. Внедрение, эксплуатация, сопровождение, выполнение контроля изменений | |
| установка производственной системы | создание и сопровождение плана установки |
| | распределение ПО |
| | установка ПО |
| | приемка ПО |

| Процессы | |
|--------------------------|--|
| Действия | Задачи |
| эксплуатация и поддержка | выполнение операций в системе |
| | обеспечение технической помощи и выполнение консультаций |
| | поддержка журналов запросов о помощи |
| процесс сопровождения | анализ запросов |
| | внесение изменений |

Как было указано выше, модель нашла свое применение не только в свободных, но и в коммерческих проектах. Зачастую, факт использования модели в таких проектах предъявляет требования соответствия процессов модели к стандартам на процессы жизненного цикла ПО. Из рис. 2 видно, все процессы, определяемые стандартами ISO/IEC 12207 [5,6] и ДСТУ 3918-1999 [1] нашли свое отображение в модели.

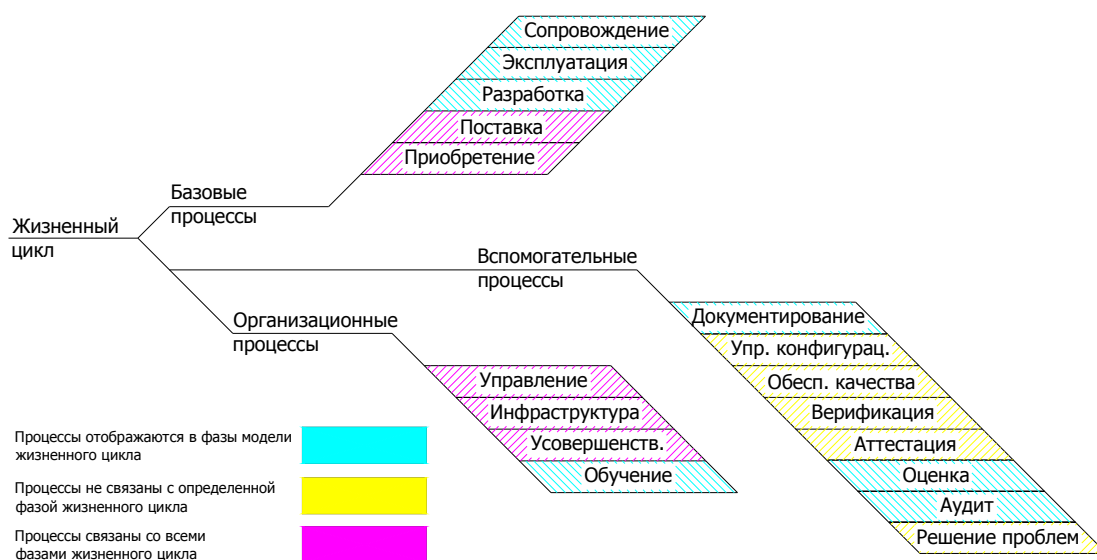


Рис. 2. Соответствие модели стандартам ISO/IEC 12207 и ДСТУ 3918-1999

Выводы

1. Собраны, проанализированы и структурированы теоретические

сведения об открытой модели жизненного цикла программных проектов.

2. Выделены составляющие жизненный цикл фаз модели, дано определение модели, приведено графическое представление модели (рис. 1).
3. Выделены и обозначены этапы разработки, не входящие непосредственно в модель (рис. 2).
4. Идентифицирована структура действий и задач, выполняемых в процессе разработки по открытой модели (таблица 1).

ЛИТЕРАТУРА

1. Писаревский М. Решат ли новые стандарты проблемы ЖЦ ИСУ? // COMPUTERWORLD Украина. – 2002. – №33. – С. 2-6.
2. Ройс У. Управление проектами по созданию программного обеспечения. Унифицированный подход. М.: Издательство “ЛОРИ”, 2002. – 426 с.
3. Шафер Д.Ф., Фартрелл Р.Т., Шафер Л.И. Управление программными проектами: достижение оптимального качества при минимуме затрат. М.: Издательский дом “Вильямс”, 2003. – 1136 с.
4. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002. – 496 с.
5. Lewis Gray ISO/IEC 12207 Software Lifecycle Processes // Crosstalk. – 1997. – №4. – pp. 4-7.
6. Singh Raghu Introduction to International Standard ISO/IEC 12207: Software Life Cycle Processes. – Tutorial slides. – 1995.