

Министерство образования и науки, молодежи и
спорта Украины

Харьковский национальный университет

Ревенчук Илона Анатольевна

Программной инженерии

Раздел 3 - Выбор и подгонка модели жизненного цикла разработки ПО

6.050103 - Программная инженерия, 8.05010301 -
Программное обеспечения систем, 8.05010302 -
Инженерия программного обеспечения

Харьков

2005

Содержание

Введение.....	3
Теория.....	4
Алгоритм выбора приемлемой модели ЖЦ разработки ПО.....	4
Категории характеристик проекта.....	4
Алгоритм подгонки модели жизненного цикла разработки ПО.....	7
Практика.....	9
Вопросы для самопроверки.....	9
Выводы.....	10
Перечень ссылок.....	11

Раздел 3 - Выбор и подгонка модели жизненного цикла разработки ПО

В разделе рассматриваются: матрицы подгонки ЖЦ, алгоритм выбора приемлемой модели ЖЦ разработки ПО, алгоритм подгонки модели ЖЦ разработки ПО.

Теория

Алгоритм выбора приемлемой модели ЖЦ разработки ПО

1. Проанализировать отличительные категории проекта.
2. Ответить на вопросы каждой категории, обведя кружочком слова "да" или "нет".
3. Расположить по степени важности категории или вопросы, относящиеся к каждой категории, относительно проекта.
4. Воспользоваться упорядоченными категориями для разрешения противоречий, возникающих при сравнении моделей.

Категории характеристик проекта

Категория требований

Категория требований состоит из вопросов относительно требований, которые предъявляет пользователь к проекту. В терминологии их иногда называют свойствами системы, которая будет поддерживаться данным проектом.

Таблица 3.1 - Матрица категории требований

Требования	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
1. Являются ли требования легко определяемыми и/или хорошо известными?	Да	Да	Нет	Нет	Да	Нет
2. Могут ли требования заранее определяться в цикле?	Да	Да	Нет	Нет	Да	Да
3. Часто ли будут изменяться требования в цикле?	Нет	Нет	Да	Да	Нет	Нет
4. Нужно ли демонстрировать требования с целью определения?	Нет	Нет	Да	Да	Да	Нет
5. Требуется ли для демонстрации возможностей проверка концепции?	Нет	Нет	Да	Да	Нет	Нет
6. Будут ли требования отражать сложность системы?	Нет	Нет	Да	Да	Нет	Да

7. Обладает ли требование функциональными свойствами на раннем этапе?	Нет	Нет	Да	Да	Да	Да
---	-----	-----	----	----	----	----

Категория команда разработчиков

Категория команда разработчиков. По возможности, в состав команды разработчиков лучше всего отобрать персонал еще до того, как будет выбрана модель жизненного цикла. Характеристики такой команды играют важную роль в процессе выбора, поскольку она несет ответственность за удачное выполнение цикла и может оказать помощь в процессе выбора.

Таблица 3.2 - Матрица категории команда разработчиков

Команда разработчиков	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
8. Являются ли проблемы предметной области проекта новыми для большинства разработчиков?	Нет	Нет	Да	Да	Нет	Нет
9. Является ли технология предметной области проекта новой для большинства разработчиков?	Да	Да	Нет	Да	Нет	Да
10. Изменяются ли роли участников проекта во время ЖЦ?	Нет	Нет	Да	Да	Нет	Да
11. Являются ли инструменты, используемые проектом, новыми для большинства разработчиков?	Да	Да	Нет	Да	Нет	Нет
12. Могут ли разработчики проекта пройти обучение?	Нет	Да	Нет	Нет	Да	Да
13. Является ли структура более значимой для разработчиков, чем гибкость?	Да	Да	Нет	Нет	Нет	Да
14. Будет ли менеджер проекта строго отслеживать прогресс команды?	Да	Да	Нет	Да	Нет	Да
15. Важна ли легкость распределения ресурсов?	Да	Да	Нет	Нет	Да	Да
16. Приемлет ли команда равноправные обзоры и инспекции, менеджмент/обзоры заказчика, а также стадии?	Да	Да	Да	Да	Нет	Да

Категории коллектив пользователей

Категория коллектив пользователей. На начальных фазах проекта можно получить четкое представление о коллективе пользователей и его будущей взаимосвязи с командой разработчиков на протяжении всего проекта. Такое представление поможет вам при выборе подходящей модели, поскольку некоторые модели требуют усиленного участия пользователей в процессе разработки и изучения проекта.

Таблица 3.3 - Матрица категории коллектив пользователей

Коллектив пользователей	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
17. Будет ли присутствие пользователей ограничено в жизненном цикле?	Да	Да	Нет	Да	Нет	Да
18. Будут ли пользователи знакомы с определением системы?	Нет	Нет	Да	Да	Нет	Да
19. Буду ли пользователи ознакомлены с проблемами предметной области?	Нет	Нет	Да	Нет	Да	Да
20. Будут ли пользователи вовлечены во все фазы жизненного цикла?	Нет	Нет	Да	Нет	Да	Нет
21. Будет ли заказчик отслеживать ход выполнения проекта?	Нет	Нет	Да	Да	Нет	Нет

Категория тип проекта и риски

И, наконец, уточним, что собой представляют тип проекта и риски, которые были рассмотрены как элементы, определение которых осуществляется на фазе планирования. В некоторых моделях предусмотрен менеджмент рисков высокой степени, в то время как в других он не предусмотрен вообще. Выбор модели, которая делает возможным менеджмент рисков, не означает, что вам не нужно составлять план действий, направленный на минимизацию выявленных рисков. Такая модель просто обеспечивает схему, в рамках которой можно обсудить и выполнить данный план действий.

Таблица 3.4 - Матрица категории тип проекта и риски

Тип проекта и риски	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
22. Будет ли проект идентифицировать новое направление продукта для организации?	Нет	Нет	Да	Да	Нет	Да
23. Будет ли проект иметь тип	Нет	Да	Да	Да	Да	Да

системной интеграции?						
24. Будет ли проект являться расширением существующей системы?	Нет	Да	Нет	Нет	Да	Да
25. Должна ли быть высокая степень надежности?	Нет	Да	Нет	Да	Нет	Да
26. Будет ли финансирование проекта стабильным на всем протяжении ЖЦ?	Да	Да	Да	Нет	Да	Нет
27. Ожидается ли длительная эксплуатация продукта в организации?	Да	Да	Нет	Да	Нет	Да
28. Будет ли система изменяться с применением непредвиденных методов, на этапе сопровождения?	Нет	Нет	Да	Да	Нет	Да
29. Является ли график ограниченным?	Нет	Нет	Да	Да	Да	Да
30. Будет ли проект идентифицировать новое направление продукта для организации?	Нет	Нет	Да	Да	Нет	Да
31. Являются ли "прозрачными" интерфейсные модули?	Да	Да	Нет	Нет	Нет	Да
32. Доступны ли повторно используемые компоненты?	Нет	Нет	Да	Да	Да	Нет
33. Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)?	Нет	Нет	Да	Да	Нет	Нет

Алгоритм подгонки модели жизненного цикла разработки ПО

1. Ознакомьтесь с различными моделями.
2. Просмотрите и проанализируйте возможные виды работ: разработка, модернизация, сопровождение и т.д.
3. Выберите самый подходящий жизненный цикл, используя для этого матрицы критериев: высокая степень риска, пользовательский интерфейс, высокая надежность, время доставки на рынок/выпуска продукта, приоритеты пользователя, уточнение требований, ожидаемый срок эксплуатации системы, технология, размер и сложность, возможный параллелизм, а также интерфейсы для существующих и новых систем.
4. Проанализируйте, насколько выбранный жизненный цикл соответствует стандартам вашей

организации, ваших заказчиков или типа проекта - ISO, IEEE и т.д.

5. Сформулируйте набор фаз и действий, образующих каждую фазу.
6. Определите внутренние и внешние производимые продукты.
7. Определите шаблоны и внутреннее содержимое поставляемых продуктов.
8. Определите действия по обзору, инспектированию, верификации и аттестации, а также стадии проекта.
9. Выполните оценку эффективности схемы жизненного цикла и проведите ее модернизацию там, где это необходимо.

Практика

Вопросы для самопроверки

1. Алгоритм выбора приемлемой модели ЖЦ разработки ПО.
2. Примеры матриц категорий.
3. Алгоритм подгонки модели жизненного цикла разработки ПО.

Вывод к разделу 3 - Выбор и подгонка модели жизненного цикла разработки ПО

В разделе рассматривались матрицы подгонки ЖЦ, алгоритм выбора приемлемой модели ЖЦ разработки ПО, алгоритм подгонки модели ЖЦ разработки ПО.

Перечень ссылок

Источники, использованные в материалах

Американский национальный стандарт ANSI/PMI 99-001-2004. Руководство к Своду знаний по управлению проектами. Введ. 2004.- Третье издание. (Руководство PMBOOK®). 401с.

Мари Кантор. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения . СПб. Вильямс. 2002. -642с.

Управление программными проектами. Достижение оптимального качества при минимуме затрат. Роберт, Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер / М-СПб-К. Вильямс. 2003. -1118с.

Уокер Ройс. Управление проектами по созданию программного обеспечения . М. Лори. 2002. -450с.

Элейн Маркел. Microsoft Project 2002. Библия пользователя. М. Диалектика. 2003. -880с.

Microsoft Project 2003 course certification materials. Trainer kit. [Электронный ресурс] Режим доступа: <http://www.cheltenhamcourseware.com/>