DOI: 10.17117/na.2016.12.02.076

<http://ucom.ru/doc/na.2016.12.02.076.pdf>

Поступила (Received): 31.12.2016

Кучерено Д.А., Котиков П.Е.
О сложности модернизации ранее
существующих программных систем

Kutcherenko D.A., Kotikov P.E.
On the complexity of upgrading the pre-existing software systems

В статье проводится анализ основных аспектов сложности принятия решений о создании новой версии программного обеспечения или усовершенствовании существующего

Ключевые слова: программное обеспечение, программные системы динамический анализ, обновление программного обеспечения

The article analyzes the main aspects of the complexity of the decision-making on the establishment of a new version of the software or improving an existing one

Key words: software, software systems dynamic analysis software update

Кучеренко Дмитрий Анатольевич

Студент

Российский государственный

гидрометеорологический университет

г. Санкт-Петербург, Малоохтинский пр., 98

Kutcherenko Dimitri Anatolovich

Student

Russian state hydrometeorological university

Saint-Petersburg, Malookhtinsky ave., 98

Котиков Павел Евгеньевич

Кандидат технических наук, доцент

Российский государственный

гидрометеорологический университет

г. Санкт-Петербург, Малоохтинский пр., 98

Kotikov Pavel Evgenivich

Candidate of Technical Sciences, Student

Russian state hydrometeorological university

Saint-Petersburg, Malookhtinsky ave., 98

Что лучше сделать новое или изменить прежнее? Каковы критерии?

Независимо, что за устройство вы приобрели, для работы потребуется программное обеспечение. В своем развитии оно не стоит на месте. Программное обеспечение (ПО) развивается совместно с развитием технологий. Все сталкивались с автоматической установкой выпущенных обновлений на устройство. Очень многие, при этом, видели более новую версию ПО для устройства. Но мало кто задумывался, что подвергло разработчиков сделать новую версию ПО, а не выпустить очередное обновление. Можно ли создать такое ПО, для которого не потребуются обновления или проекты более новых версий?

Каждое программное средство, предназначенное для работы на конкретной программно-аппаратной платформе всегда имеет срок службы. Он зависит от стремления и возможностей разработчика поддерживать свое решение в процессе эксплуатации заказчиками. Если разработчик отказывается поддерживать выпущенный им продукт, то у использующих его возникает желание

модернизации программного обеспечения. Цель, – соответствие возросшим потребностям или изменениям в применении. Зачем это нужно? Как можно заказать модификацию программы (мобильного приложения)? В чем причины возникновения потребности в модернизации программного обеспечения?

Возникнуть они могут, исходя из многих факторов. Однако чаще всего потребность модернизировать программу или целый комплекс возникает по причине расширения спектра задач, которые должно охватывать ПО.

Все современные программные системы являются очень сложными. Такая сложность вызвана многими причинами. Основной из них стала логическая сложность самих решаемых задач. Прежде, ЭВМ применяли в узких сферах науки и техники. Они применялись там, где задачи были детерминированы, но требовали сложных и больших вычислений. Ныне созданы развитые компьютерные сети. Также, появилась возможность возложить на сети и мощные ЭВМ решение ресурсоемких задач, о которых раньше даже не задумывались. При компьютеризации охватываются все новые предметные области. Для уже освоенных предметных областей интенсивно усложняются сложившиеся постановки конкретных задач. Сама сложность проектирования и разработки программных средств естественно увеличивается за счет сложности формального описания и определения требований к таким системам, а также в результате отсутствия средств описания поведения дискретных систем (с большим числом их состояний, при недетерминированной последовательности входных воздействий). Вызвана она коллективным характером проектирования и разработки, а также естественной необходимостью существенного увеличения степени повторяемости программных кодов. Эти факторы всегда напрямую связаны с самой сложностью объекта, для автоматизации процессов на котором ведется проектирование и разработка программного обеспечения. Есть еще одно важное основание для моделирования жизненного цикла программного обеспечения. Пристального внимания требует изучение самих требований при развитии программных проектов. Традиционные технологии всегда рассматривают определение и анализ требований в пределах предварительного этапа. Он предшествует собственно разработке. Тогда выявляется вся информация для последующего конструирования. Считается, что успешность дальнейшей работы над проектом непосредственно зависит от того, как полно и действительно тщательно выполнен этап анализа. Известно, что внесение корректив в уже определенные требования приводит к дальнейшей необходимости нового повторения проектирования и повторения всех других последующих этапов. Можно считать, что последующее изменение требований уже в процессе разработки всегда рассматривается как ошибка аналитического этапа.

Точное определение понятия требования указывает, что в реальности очень трудно, исходя из реально аморфных и противоречивых пожеланий заказчика, выявить, что конкретно и в каком виде должно быть воплощено в программном изделии. Требования первичны по отношению к программной разработке, определяют все ее развитие, являются начальным звеном в слагаемых качества конструируемых программ. Ясно, что задача управления требованиями должна усматриваться в качестве одной из главных задач каждого проекта, претендующего на последующую реальную полезность.

Вне зависимости от состояния начальной проработки требований к разработке, нет оснований рассчитывать, что эти требования всегда будут неизменными. В следующий момент развития появятся совсем новые требования или изменятся старые требования, а иные – исчезнут. Сложность управления процессом изменения требований во времени, – даже не в этом. Она в том, что изменения одних требований влияет на другие. Требуется отслеживать эти влияния. Изменения требований распространяется на все действующие элементы проекта, в том числе, – на программные продукты.

Отслеживание требований и распространение изменений, – это еще один мотив развития действующих моделей жизненного цикла программного средства. При создании модели жизненного цикла всегда следует обозначить этапы, на которых производятся действия по трассировке (отслеживанию). Поэтому нам важно разделять два варианта работы с требованиями непосредственно в проекте объектно-ориентированного решения.

Существует модель жизненного цикла с мини-циклами обработки требований. Она адекватно описывает процесс осуществляемой разработки проекта программного обеспечения. Это относится к его стационарному периоду, когда начальный период (итерация) развития проекта уже осуществлен. Первая итерация бывает всегда особой. Именно тогда закладываются основы сопутствующих типов иных требований, также глоссария и иных составляющих. Они используются далее в стационарный период. Возможно выделить и другие особенности начального периода, обусловленные заданием направления развития проекта на будущую жизнь программы. Неудачный выбор, – нет дальше наращивания возможностей. Удачный выбор, – дальше нет переделок. Такова реальность!

Фундаментальное значение для моделирования фазы завершения проекта имеет характер обратной связи с пользователем системы. Как правило, это сообщения о ходе эксплуатации: мнения, заключения, рекламации, нарекания, а также конкретные претензии. Такие сообщения возможно классифицировать, ранжировать по степени важности для проекта в дальнейшем.

Таким образом, можно сказать, что все ПО со времени потребует обновлений. Именно поэтому и возникает вопрос, что лучше: сделать новое ПО или обновить старое? Как итог анализа, можно утверждать, что проблема достаточной обоснованности действий по обновлению или переизданию программного обеспечения остается на сегодня совершенно актуальной и важной.

Список используемых источников:

1. Котов С.Л., Палюх Б.В., Федченко С.Л. Разработка, стандартизация и сертификация программных средств и информационных технологий и систем. Тверь: ТГТУ, 2006. 104 с.
2. Гаврилова И.В. Разработка приложений. М.: ФЛИНТА, 2012. 242 с.
3. Емельянова Н.З., Партыка Т.Л., Попов И.И. Проектирование информационных систем. М.: Форум, 2009. 432 с.
4. Национальный открытый университет. URL: www.intuit.ru