УДК 004.415.2

А.А. Стенин, В.П. Пасько, В.А. Лемешко

ИНФОРМАЦИОННО-ЛОГИЧЕСКАЯ МОДЕЛЬ УПРАВЛЕНИЯ ПРОЦЕССОМ РАЗРАБОТКИ ИННОВАЦИОННЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Аннотация: В данной статье предложена информационно-логическая модель процесса управления разработкой инновационных программных продуктов. Суть предлагаемого подхода заключается в том, что их разработка интерпретируется, как информационный объект, который содержательно и структурно меняется в процессе его создания.

Ключевые слова: процесс разработки программных продуктов, информационно-логическая модель управления, управляющее воздействие, состояние процесса.

Введение. Сегодня в принятии решений задействуется «поле знаний», неопределённое не только по объёму, но и по размерности. В процессе выработки решения учитываются динамично изменяющиеся экономические, технологические, организационные, природные и др. аспекты. Источниками информации могут служить знания опытных специалистов, литература, нормативные материалы научных и проектных организаций, Интернет-ресурсы [1,2,3].

Анализ моделей процесса разработки инновационных программных продуктов. Информационная структура процесса управления созданием инновационных программных продуктов (ИПП) должна предусматривать возможность пополнения системы управления новыми знаниями. Каждое наблюдение может добавлять и новое знание о программном продукте, непредусмотренное регламентом, но, в перспективе, повышающее эффективность управления. Несмотря на высокую степень неопределённости в процессе создания программного продукта, обусловленное наличием человеческого фактора, сам процесс создания программного продукта является вполне детерминированным и состоит из ряда этапов, которые могут быть построены по разным схемам. Рассмотрим наиболее известные модели разработки ИПП [4,5].

Каскадная модель. Первой моделью, получившей широкую известность и действительно структурирующей процесс разработки, является каскадная или водопадная (рис.1) [6,8]. Разделяет процесс создания программного продукта на последовательные этапы (следует отметить, что она уже применялась тогда различными разработчиками, однако ни количество, ни содержание этапов не унифицировалось).

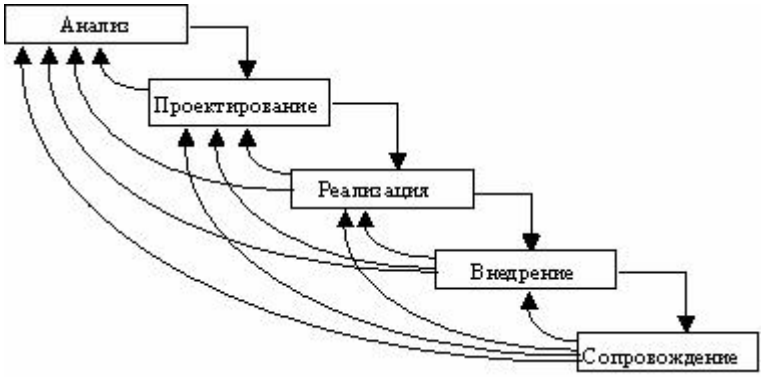

Рисунок 1 - Каскадная модель

Однако практическое использование данной модели выявило множество ее недостатков, главный из которых в том, что она больше подходит для традиционных видов инженерной деятельности, чем для разработки ПО. В частности, одной из самых больших проблем оказалась ее «предрасположенность» к возможным несоответствиям полученного в результате продукта и требований, которые к нему предъявлялись. Основная причина этого заключается в том, что полностью сформированный продукт появляется лишь на поздних этапах разработки, но так как работу на разных этапах обычно выполняли различные специалисты и проект передавался от одной группы к другой, то по принципу испорченного телефона оказывалось так, что на выходе получалось не совсем то, что предполагалось вначале.

V-образная модель. Была предложена именно для того, чтобы устранить недостатки каскадной модели, а название – V-образная, или шарнирная – появилось из-за ее специфического графического представления (рис. 2)[7,10].

 ISSN 1562-9945

Рисунок 2 - V-образная модель

V-образная модель дала возможность значительно повысить качество ПО за счет своей ориентации на тестирование, а также во многом разрешила проблему соответствия созданного продукта выдвигаемым требованиям благодаря процедурам верификации и аттестации на ранних стадиях разработки (пунктирные линии на рисунке указывают на зависимость этапов планирования/постановки задачи и тестирования/приемки[7]).

Однако в целом V-образная модель является всего лишь модификацией каскадной и обладает многими ее недостатками. В частности, и та и другая слабо приспособлены к возможным изменениям требований заказчика. Если процесс разработки занимает продолжительное время (иногда до нескольких лет), то полученный в результате продукт может оказаться фактически ненужным заказчику, поскольку его потребности существенно изменились.

Важен также вопрос планирования показателей ожидаемой функциональности, поскольку в этих моделях он является не более чем допущением: в частности, определить, какую скорость обработки данных обеспечит создаваемый продукт либо сколько он будет занимать памяти, на этапе постановки задачи практически невозможно. Если подобные требования четко зафиксированы, то вполне вероятно, что полученное решение не будет им удовлетворять, хотя известно это станет только на завершающих этапах разработки, когда основные ресурсы уже израсходованы.

Спиральная модель. Предложенная Барри Боэмом в 1988г., стала существенным прорывом в понимании природы разработки ПО, хо-

тя, по большому счету, является объединением двух моделей: каскадной и на основе создания прототипов (рис. 3) [2,6,8].

Рисунок 3 - Спиральная модель Боэма

Спиральная модель Боэма сфокусирована на проектировании. Фактически, разработка ПО происходит лишь на последнем витке спирали по обычной каскадной модели, однако этому предшествует несколько итераций проектирования на основе создания прототипов – при этом каждая итерация включает стадию выявления и анализа рисков и наиболее сложных задач.

Поскольку спиральная модель в основном охватывает именно проектирование, то в первоначальном виде она не получила широкого распространения в качестве метода управления всем жизненным циклом создания ПО. Однако главная ее идея, заключающаяся в том, что процесс работы над проектом может состоять из циклов, проходящих одни и те же этапы, послужила исходным пунктом для разработки предлагаемой ниже информационно-логической модели управления процессом разработки ИПП.

Информационно-логическая модель управления. Суть предлагаемого подхода заключается в том, что процесс разработки ИПП интерпретируется, как информационный объект, который содержательно и структурно меняется в процессе его создания. Следовательно, процесс разработки ИПП может быть описан упорядоченной последовательностью состояний разрабатываемого ИПП, последнее из которых представляет готовый программный продукт, т.е.

$$S_0 \to S_1 \to S_2 \to \ldots \to S_i \to \ldots \to S_n. \qquad (1)$$

 ISSN 1562-9945

При этом каждое состояние S_i характеризуется некоторой совокупностью параметров $m_i^1, m_i^2, \ldots, m_i^m$,, где i=. При этом, каждое состояние S_i характеризует степень завершённости разработки ИПП.

Будем, в дальнейшем полагать, что каждому промежуточному состоянию Si соответствуют две интегральные оценки P_i и Q_i, которые исчерпывающе характеризуют степень завершённости ИПП с количественной и качественной сторон. Очевидно, что функции P и Q на упорядоченном множестве состояний Si (i=) должны иметь возрастающий характер. Для полноты описания процесса разработки ИПП будем предполагать, что интегральные оценки P и Q независимы друг от друга.

Далее, процесс разработки ИПП разбиваем на отдельные подпроцессы (шаги разработки), соответствующие принятым согласно (1) этапам разработки. Обозначим приращение интегральных характеристик, достигнутое на i-м этапе через ΔP_i и ΔQ_i.

Процесс управления разработкой ИПП состоит в том, что на каждом шаге задаётся управляющее её воздействие $u(i)$, которое определяет значения ΔPi и ΔQ_i и переводит степень разработки ИПП из состояния ($P_{i\text{-}1}$, $Q_{i\text{-}1}$) в состояние (P_i, Q_i). Управление $u(i)$ можно рассматривать как выбор одного из альтернативно возможных способов обеспечения инновационности программному продукту. При этом перевод в новое состояние реализуется выполнением определённого множества процедур.

Естественно, что на каждом шаге i на управляющее воздействие $u(i)$ налагается ряд ограничений естественного и искусственного характера. Иначе, $u(i)$ может принимать значения из некоторого множества возможных управляющих воздействий, т.е.

$$u(i) \in V(i). \qquad (2)$$

Будем считать, что для $i = 0$ $P_0 = Q_0 = 0$.

Значения интегральных характеристик на последующих шагах определим формулами:

$$\begin{cases} P_i = \varphi(u(k), P_{i-1}); \\ Q_i = \gamma(u(k), Q_{i-1}); \\ [(P]_i, Q_i) = f(u(i), [(P]_{i-1}, Q_{i-1})); i = \overline{1,n}. \end{cases} \qquad (3)$$

Под (P_i, Q_i) будем понимать множество всех состояний процесса разработки ИПП, в которое его можно перевести из начального состояния за і шагов, пользуясь управляющим воздействием $u(k) \in V(k), k = \overline{1,i}$.

Такое множество назовём множеством достижимости (P_i , Q_i), которое определяется с помощью рекуррентных соотношений (4) вида

$$\begin{cases} [[(P]]_k, Q_k) = F[u(k), [[(P]]_{k-1}, Q_{k-1})] \\ u(k) \in V(k), k = \overline{1,i}, i = \overline{1,n} \end{cases} \quad (4)$$

В задании на разработку ИПП обязательно указываются требования, которым должен удовлетворять ИПП после окончания её разработки. Исходя из этого можно определить показатели P_n и Q_n, характеризующие конечное состояние разработки, которое должно принадлежать некоторой области допустимых значений (P_n^*,), т.е.

$$[[(P]]_n, Q_n) \in (P_n^*,). \quad (5)$$

Таким образом, процесс разработки ИПП с управляющими воздействиями $V(u(i))$, будет допустимым, если $u(i)$ переведут ИПП из начального состояния в конечное, которое будет удовлетворять условию (5).

Исходя из этого, для успешного достижения цели – разработки ИПП необходимо выполнение условия

$$[[(P]]_n, Q_n) \in (P_n^*,), \quad (6)$$

Условие (6) означает, что множество всех состояний разработки ИПП должно находиться во множестве допустимых состояний ИПП в соответствии с предъявленными требованиями. В противном случае, при изменившемся инновационном прогнозе необходимо либо изменить техническое задание на разработку, изменив тем самым $[[(P]]_i^*,), i = \overline{1,n}$, либо расширить область возможных управляющих воздействий $u(i)$, $i = \overline{1,n}$.

Пусть в результате выполнения (i-1) шагов процесс разработки ИПП перешёл в состояние $[[(P]]_{i-1}, Q_{i-1})$. Тогда множество допустимых управляющих воздействий на i-м шаге определиться следующим образом:

$$\begin{cases} V^*(i) = \{u(i): (P_i, Q_i) = f[u(i), [[(P]]_{i-1}, Q_{i-1})]\} \\ (P_i, Q_i) \in [[(P]]_i^*, Q_i^*), i = \overline{1,n} \end{cases} \quad (7)$$

В результате процесс управления разработкой ИПП в окончательном виде можно записать как:

$$u(i) \in V(i) \cap V^*(i), i = \overline{1,n} \quad (8)$$

Условие (8) означает, что с точки зрения возникших инноваций при разработке ИПП возможно изменение управляющих воздействий в допустимых значениях в соответствии с изменениями текущих и ко-

 ISSN 1562-9945

нечных требований. Условию (8) на каждом шаге может удовлетворять несколько управляющих воздействий.

Выводы. Процесс разработки ИПП является многоальтернативным, т.е. возникает проблема многокритериальности, которая, как правило, требует привлечения интеллектуальных систем поддержки принятия решений (ИСППР). Это обусловлено тем, что, во-первых, наличие человеческого фактора в процессе разработки ИПП вносит большую долю неопределённости, и, во-вторых необходимо посмотреть весь спектр допустимых решений в области использования ИПП, что требует разработки автоматизированных методов извлечения знаний данной предметной области из различных источников информации, в том числе и из Интернет ресурсов.

ЛИТЕРАТУРА

1. Липаев В.В. и др. Технология проектирования комплексов программ АСУ. — М.: Радио и связь, 1983. – 235 с.

2. Боэм Б.У. Инженерное проектирование программного обеспечения. — М.: Радио и связь, 1985. – 512 с.

3. Вендров А.М. CASE -технологии. Современные методы и средства проектирования информационных систем. — М.: Финансы и статистика, 2000.- 352 с.

4. Winston W. Royce Managing the Development of Large Software Systems Proc. IEEE WESKON, 1970, №6. –pp. 1-9.

5. Орлов С.А. Технологии разработки программного обеспечения. – СПб.: Питер. -2002. – 464 с.

6. Richard W. Selby. Software Engineering: Barry W. Boehm's LifetimeContributions to Software Development, Management, and Research. — John Wiley & Sons, 2007. - 834 с.

7. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. – М.: Лори. -2003. – 590 с.

8. Соловьев Н. А., Чернопрудова Е. Н. Системы автоматизации разработки программного обеспечения. – Оренбург. – ООО ИПК «Университет». – 2012. – 230 с.