

МЕТОД МОДЕЛИРОВАНИЯ ПРОГРАММНЫХ СИСТЕМ ДЛЯ ОЦЕНКИ ТЕХНИКО-ЭКОНОМИЧЕСКИХ ПАРАМЕТРОВ ПРОЕКТОВ ИХ РАЗРАБОТКИ

к.т.н., Демирский А.А.

Введение

Возрастающая сложность объектов автоматизации и увеличение их количества, а также переход к комплексным интегрированным решениям, приводят к увеличению сложности и количества проектов по созданию программных средств (ПС). Массовое создание сложных ПС промышленными методами и большими коллективами специалистов требует четкой организации процесса разработки ПС на основе планирования работ по проекту с оценкой необходимых ресурсов и времени для их реализации. Точность и оперативность подобных оценок напрямую влияет на качество разрабатываемых ПС и риск неуспешного завершения проекта.

Управление проектами разработки ПС обладает рядом существенных особенностей, которые затрудняют планирование подобных процессов. Это обусловлено следующими факторами:

1. Управление проектом разработки ПС связано всегда с созданием нового продукта или серьезным улучшением предыдущей системы (уже существующей). Подобный процесс всегда имеет высокую степень неопределенности.

2. Управление проектом разработки ПС является одноразовой деятельностью в силу того, что каждый проект уникален.

3. Техничко-экономические параметры проекта разработки ПС (трудоемкость, стоимость, длительность работ) определяются с трудом и часто являются не достоверными. Основная проблема заключается в широком спектре количественных и качественных показателей, которые с различных сторон характеризуют проект разработки ПС, и невысокая достоверность оценки их значений экспертами.

4. Трудно измерить производительность труда разработчиков проекта в связи с отсутствием единых подходов к измерению объемов проделанной работы. Как показывает опыт, ошибки прогнозирования показателей трудоемкости и стоимости могут достигать сотен и более процентов [1].

Таким образом, без должных подходов к прогнозированию технико-экономических параметров проектов разработки ПС, направленных на уменьшение неопределенности, велика вероятность перехода проекта в состояние «стихийности», которое в большинстве случаев приводит к неудовлетворительным результатам.

Согласно стандарту ГОСТ Р ИСО/МЭК 12207 «Информационные технологии – Процессы жизненного цикла программ» начальным этапом процесса «Разработка» является этап анализа, цель которого - выявление, классификация и формализация информации обо всех аспектах предметной области, влияющих на свойства конечного продукта, и именно этот этап оказывает определяющее влияние на качество результатов всего проекта. На данном этапе разработчику создаваемого ПС необходимо провести оценку реализуемости проекта в условиях и ресурсах, предлагаемых заказчиком. Именно на этом этапе сторонами, участвующими в разработке ПС, принимаются договорные обязательства. Следствием недостатков или отсутствия технико-экономического обоснования проектов разработки ПС является неверная оценка преимуществ новой программной разработки, недооценка роли других конкурирующих предложений, неизбежный перерасход средств и снижение качества ПС.

На сегодняшний момент времени решение задач оценки технико-экономических параметров проектов разработки ПС производится, в основном, с помощью экспертных методов. Такие оценки имеют сильную зависимость от компетенции и объективности эксперта. Подобные оценки могут быть легко и быстро получены, но, одновременно, и трудно проверяемы, и слабо обоснованы.

Как показывают исследования Б. Бозма и Л. Путнэма [2, 3], характер проектируемой системы и сложность ее программного обеспечения, являются самыми значительными факторами, от которых зависят затраты ресурсов на разработку ПС. Под понятием «сложность ПС» подразумевается некая характеристика (или их набор) ПС определяющая, прежде всего, трудоемкость его разработки. В современных методах оценки ПС практически не представлены формализованные процедуры для прогнозирования сложности разрабатываемого ПС, адаптируемые к условиям конкретного проекта. Применение существующих методов оценки ПС на практике оказывается весьма трудоемким и доступно только сертифицированным специалистам, кроме того, они не учитывают использование современных технологий разработки ПС.

1. Метод моделирования программных систем для оценки их сложности разработки

Для оценки сложности ПС существует достаточное количество метрик: Холстеда, Маккейба, Майерса, Джилба, Хансена, Вудворда. Однако все они подразумевают обязательное наличие для анализа либо текста самой программы, либо подробного алгоритма ее работы. Этот факт делает неприемлемым использование данных подходов на ранних этапах жизненного цикла ПС, когда у разработчика имеются только

концептуальные модели архитектуры будущего ПС и отдельные эскизные проектные решения.

На ранних этапах процесса разработки ПС в качестве характеристики определяющей сложность создаваемого программного продукта обычно используется его размер. Использование размера ПС как единственной характеристики сложности, в рамках одного класса программ, является приемлемым допущением для решения задач оценки технико-экономических параметров проекта разработки ПС на ранних этапах жизненного цикла [4]. Размер ПС в современной практике, среди всех факторов, влияющих на параметры проекта, изменяется в самом широком диапазоне на три - четыре порядка от 10^4 до 10^7 строк текста программ. Поэтому методам его оценивания уделяется большое внимание.

Таким образом, на начальных этапах жизненного цикла для обеспечения качественного планирования необходимо быстро получить достаточно точную оценку сложности разрабатываемого ПС для принятия решений по проекту. Для формирования такой оценки предлагается использовать метод определения размера ПС на основе оценок размера отдельных его программных объектов. Под программными объектами будем понимать структурные элементы ПС, для создания которых требуются большие трудозатраты. Все программные объекты в рамках одного типа ПС можно объединить в классы. Для каждого класса разрабатывается отдельная модель для оценки его сложности в зависимости от значений характеристик (свойств) программных объектов. Построение таких моделей подразумевает наличие базы данных статистики по уже реализованным проектам. Это позволит исключить или уменьшить влияние экспертных оценок и учесть массу факторов связанных с производительностью труда разработчиков.

Можно выделить следующие основные этапы процедуры оценки размера ПС:

Шаг 1. Выбор класса ПС. Выбор класса определяет набор программных объектов ПС и используемых прогнозных моделей для оценки их сложности, которые должны быть предварительно созданы и настроены по историческим данным.

Шаг 2. Подсчет экземпляров программных объектов, которые должны быть реализованы при создании ПС.

Шаг 3. Определение значений характеристик программных объектов, используемых в выбранной модели оценки их размера.

Шаг 4. Получение частных оценок размеров программных объектов разрабатываемого ПС с использованием настроенных моделей.

Шаг 5. Получение итоговой оценки размера ПС, путем аккумулялирования значений частных оценок.

Ключевым этапом процедуры является построение моделей оценки размера объектов ПС на основе значений их характеристик. Сложность

экземпляров класса объектов ПС определяется с помощью базы знаний (БЗ), построенной на правилах нечеткого вывода. Роль БЗ – идентификация нелинейной зависимости $y = f(X)$, где X – вектор значений количественных свойств класса объектов ПС; y – значение размера объекта ПС в условных единицах измерения.

Для построения частных моделей оценки сложности объектов ПС предлагается использовать метод двухэтапной идентификации нелинейных зависимостей с помощью нечетких баз знаний. Первый этап – структурная идентификация – представляет собой формирование нечеткой базы знаний, которая грубо отражает взаимосвязь между входами и выходом с помощью лингвистических правил «ЕСЛИ – ТО». Лингвистические правила генерируются в результате экстракции нечетких знаний из имеющейся статистики. На втором этапе проводится параметрическая идентификация исследуемой зависимости путем нахождения таких параметров нечеткой базы знаний, которые минимизируют отклонение модельных результатов и имеющейся статистики. Данный подход реализуется путем использования нечетких нейронных сетей, способных одновременно формировать нечеткие правила и адаптировать функции принадлежности путем модификации весов связей в процессе обучения.

Основная особенность рассматриваемых моделей заключается в том, что заключения правил представляются не в виде принадлежности выходной переменной к нечетким множествам, а в форме функциональных зависимостей (нейронная продукционная сеть Такаги-Сугено-Канга) [5]:

$$P_i: \text{ЕСЛИ } x_1 \text{ есть } A_{i1} \text{ И } \dots \text{ И } x_j \text{ есть } A_{ij} \text{ И } \dots \text{ И } x_m \text{ есть } A_{im}, \quad (1)$$

$$\text{ТО } y = c_{i0} + \sum_{j=1}^m c_{ij} x_j, i = 1, \dots, n,$$

где A_{ij} – нечеткое множество, отражающее значение характеристики программного объекта x_j , с функцией принадлежности $\mu(x_j)$; x_j – количественная характеристика объекта ПС, определяющая его сложность; y – четкая выходная переменная равная искомому размеру объекта ПС, определяемая четкой функцией от значений характеристик объекта ПС; n – количество правил; m – количество входных переменных.

При ограниченном наборе обучающих данных возможно использование упрощенной модели, когда заключения правил заданы синглетами (выходная переменная представляется константой: $y = c_i$).

База знаний трактуется как некоторое разбиение пространства влияющих факторов на нечеткие подобласти, в каждой из которых значение функции отклика рассчитывается как линейная комбинация входов. Правила являются своего рода переключателями с одного линейного закона «входы-выход» на другой, тоже линейный. Границы подобластей размытые, следовательно, одновременно могут выполняться несколько линейных законов, но с различными весами. Результирующее

значение выхода y определяется как суперпозиция линейных зависимостей, выполняемых в данной точке X^* n -мерного факторного пространства (2).

$$y = \frac{\sum_{i=1}^n \left(c_{i0} + \sum_{j=1}^m c_{ij} x_j \right) \prod_{j=1}^m \exp \left[- \left(\frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]}{\sum_{i=1}^n \prod_{j=1}^m \exp \left[- \left(\frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]}. \quad (2)$$

В выражении (2) подразумевается использование функции Гаусса для задания функций принадлежности всех нечетких множеств входных переменных, где a и b – параметры функции Гаусса.

Использование разработанного метода позволит добиться следующих преимуществ:

- метод позволяет использовать имеющуюся статистическую информацию по завершённым проектам разработки ПС как количественной, так и качественной природы, отображая ее в терминах естественного языка – в форме нечетких продукционных правил ЕСЛИ – ТО. Это позволяет учесть массу факторов связанных с технологическим процессом разработки ПС в конкретной проектной организации;
- предлагаемый метод не предполагает наличия каких-либо сведений о виде функциональных зависимостей входов и выходов модели;
- получаемые с помощью разработанного метода модели допускают достаточно понятную интерпретацию.

2. Результаты применения предлагаемого подхода к оценке сложности ПС

С использованием предложенного метода были разработаны модели оценки размера объектов ПС этапа проектирования интерфейса пользователя для класса систем обработки данных. На данном этапе количество типов объектов, которые можно использовать при подсчете размера ПС, увеличивается, что позволяет построить более точные модели для оценки сложности разрабатываемого ПС. В качестве элементов ПС на данном этапе выделялись следующие классы объектов: экранные формы, диалоги, отчеты, файлы данных (таблицы РСУБД), функции импорта/экспорта данных. Для измерения размера в моделях использовались метрики LOC (line of code) – строки кода и функциональные баллы (functional point) [4]. Данные для построения моделей были получены исходя из анализа ПС, поступивших на сертификационные испытания в ООО «ГИЦ ПС ВТ» (Главный испытательный центр программных средств и вычислительной техники).

Полученные математические модели встраиваются в стандартизированную процедуру оценки, которая описывает порядок их применения для получения адекватных прогнозов. В правильно оцениваемом проекте входные данные включают техническую спецификацию, приоритеты и ограничения. Эти входные данные регулируются до тех пор, пока процесс оценки не даст приемлемый результат. Исторические данные также относятся к входным данным оценки и используются для калибровки предположений о сложности и производительности.

Были проведены оценки ряда реальных проектов среднего и крупного масштаба с размером ПС от 10^4 до 10^6 в строках исходного кода. Все оценки проводились на этапе проектирования интерфейса пользователя ПС. В результате оценки было получено, что оценочные данные о размере проекта, рассчитанные с использованием созданных моделей отличаются от фактических на 3-5% для крупных проектов и 5-10% для средних. Данные цифры говорят о достаточно высокой степени точности оценок, полученных на начальных этапах разработки [1]. Рост точности оценки для крупных проектов объясняется увеличением количества программных объектов, из которых оно состоит. Это приводит к уменьшению влияния отдельных больших смещений в оценке на общий результат. Следовательно, нецелесообразно применять предложенный метод оценки размера ПС для очень малых проектов (менее 5000 строк кода).

Результаты работы были использованы в НИОКР «Разработка методического и инструментального обеспечения поддержки планирования материальных и временных затрат на разработку программного обеспечения АИС органов государственной власти, создаваемого в рамках работ по реализации мероприятий федеральной целевой программы «Электронная Россия (2002-2010 годы)». Указанная НИОКР зарегистрирована в едином реестре результатов научно-технической деятельности Роснауки номер 13293.6904003214.07.1.001.2.

На данный момент, предложенный метод оценки сложности разработки ПС был адаптирован для автоматизированных банковских систем. Результаты работы внедрены в Банке России.

Список литературы

1. McConnell S. Professional Software Development. Boston, MA: Addison-Wesley. 2004.
2. Boehm B.W. Software cost estimation with COCOMO II. Prentice Hall PTR. New Jersey. 2000.
3. Putnam L. H. Five Core Metrics. New York, NY: Dorset House. 2003.
4. Липаев В.В. Техничко-экономическое обоснование проектов сложных программных средств. М.:СИНТЕГ, 2004.
5. Борисов В. В. Нечеткие модели и сети. М.: Горячая линия – Телеком, 2007.