

УДК 622

С. А. ХАРИН, А. Ю. КОКИН, В. И. УСАЙТЕ

АНАЛИЗ ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА И МАШИНЫ В МЕНЕДЖМЕНТЕ ПРОЕКТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В менеджменте проектов программного обеспечения важнейшим условием разработки эффективных программ является учет различных аспектов взаимодействия человека и машины. Вопросы такого взаимодействия рассмотрены на примере автоматизации задач организации и управления в горном деле. В ходе исследований проведен анализ поведения человека при разработке ряда компьютерных программ на языке Java.

Ключевые слова: анализ, взаимодействие, человек, машина, менеджмент, проект, программное обеспечение, метод, объект, класс, параметры.

Сложность, многоаспектность системы “человек-машина” обусловлены тем, что это система, в которой человек-оператор или группа операторов взаимодействует с техническим устройством в процессе производства, управления, обработки информации. Известно, что требования к эргатическим системам обеспечиваются разрешением двух взаимосвязанных задач: сокращение объема информации путем рационального ее кодирования и уменьшение величины перемещения анализаторов при восприятии информации.

Важнейшая задача менеджмента разработки программного обеспечения – продвижение проекта к указанным в момент его начала результатам. Такая задача сводится к распределению и контролю эффективного использования имеющихся ресурсов проекта.

Анализ ранее опубликованных результатов исследований [1–4] и современного состояния практики менеджмента проектов программного обеспечения указывает на необходимость более активного использования различных аспектов взаимодействия человека и машины.

В менеджменте проектов программного обеспечения важнейшим условием разработки эффективных программ является учет различных аспектов взаимодействия человека и машины.

Цель статьи – анализ различных видов программ, использующих параметризованные методы и конструкторы, разработка эффективного программного обеспечения для исследования вопросов взаимодействия человека и машины.

Вопросы эффективности создания программного обеспечения связаны с комплексом факторов, в частности, таких как оптимальность среды разработки, то есть системы программных средств, используемых программистами для разработки программного обеспечения, возможности языка программирования, оптимальность кода по комплексу факторов (рис. 1).



Рис. 1. Факторы, влияющие на эффективность создания программ

Весьма важным является также уровень эффективности менеджмента создания программного обеспечения, особенно с учетом масштабности проектов, работы коллектива разработчиков и возникающих в их среде взаимоотношений, условий труда в широком смысле, работоспособности конкретных участников процесса, зависящей от множества факторов.

Проблема состоит в том, что для высокой эффективности создания программного обеспечения необходимо обеспечение оптимальных параметров не какого-либо одного, а всей совокупности факторов, влияющих на такую эффективность, что достаточно сложно. Кроме того, степень влияния каждого из указанных выше факторов на уровень качества программного обеспечения весьма различна, а изучение его представляет собой сложную задачу.

Разработка масштабных программ – это коллективный труд, требующий эффективного менеджмента. В группах специалистов, выполняющих программные проекты, возможны самые разнообразные организационные структуры. Например, для сложных, объемных по трудозатратам проектов иерархические цепочки “менеджер – менеджер по направлению – исполнитель работ” целесообразно увеличивать, дифференцируя работы и сферы ответственности для некоторых менеджеров по направлению.

Вместе с тем, для сравнительно небольших групп возможна следующая организация работ: обязанности главного менеджера распределяются по команде разработчиков, которая за счет внутренних механизмов решает, как планировать работы, как их распределять и контролировать.

В настоящее время получил широкое распространение язык программирования Java в интегрированной среде разработки модульных кроссплатформенных приложений Eclipse, использующих параметризованные методы. Это является существенным фактором обеспечения эффективности программирования.

Эффективное создание программных продуктов связано также с оптимальным лаконичным кодом, позволяющим решать поставленные задачи. Вместе с тем, излишняя лаконичность кода может приводить к дополнительным ошибкам, особенно у недостаточно опытных и в состоянии утомления программистов.

Поддержание работоспособности программистов – существенный фактор эффективного программирования. Вопрос сводится к управлению ее уровнем и использованию моментов наиболее высокой работоспособно-

сти для написания наиболее ответственных фрагментов программ. Это является признаком высокого уровня менеджмента программных продуктов в рамках взаимодействия человека и машины.

В современных условиях актуальной является разработка методов исследований соответствующего программного обеспечения, которые позволили бы служить в качестве инструментов изучения вопросов организации строительства. Рассмотрим решение задачи определения скорости строительства горных выработок с помощью следующих программ, написанных на языке Java (табл.).

Таблица

Программы, используемые для анализа

№	Вид программы
1	Программа, включающая метод внутри класса
2	Программа, включающая возвращающий скорость метод
3	Программа, которая использует параметризованный метод
4	Класс, использующий параметризованный конструктор
5	Конструктор для разрешения конфликтов пространства имен

Анализ сопоставимых по выполняемым задачам фрагментов программ расчета скоростей проходки выработок по числу знаков отражён на рис. 2. Очевидно, что наименьшим их числом характеризуется вариант 5 программы, представленный конструктором с ключевым словом `this`. Вместе с тем, такие программы подвержены опасности совершения программистом ошибки во время инициализации переменных. По этому фактору выглядит приемлемой весьма простая программа версии 1, включающая метод внутри класса, которая не слишком сильно превышает по числу знаков версию 5.

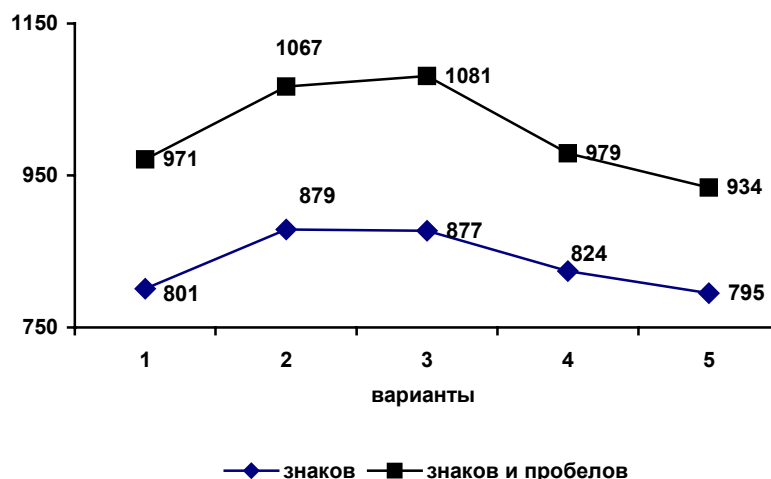


Рис. 2. Анализ фрагментов программ по числу знаков

Представляет интерес анализ числа ошибок, совершаемых программистами при изменении продолжительности их работы, безотносительно к

используемым при этом вариантам программ, дополнительный учет чего был бы с социологической точки зрения в существующих условиях весьма затруднителен. Режим работы программистов предусматривал короткий отдых в течение каждого часа и непродолжительный обеденный перерыв и отражал индивидуальный стиль работы каждого.

Отмечено, что совокупное часовое количество ошибок (рис. 3) опрошенных программистов в первый период рабочего дня со временем даже несколько сократится по сравнению с самым его началом, но затем будет возрасти с нарастающей интенсивностью. Общий же характер зависимости количества ошибок от времени работы программистов наблюдаемой группы может быть с удовлетворительной точностью описан экспоненциальной функцией.

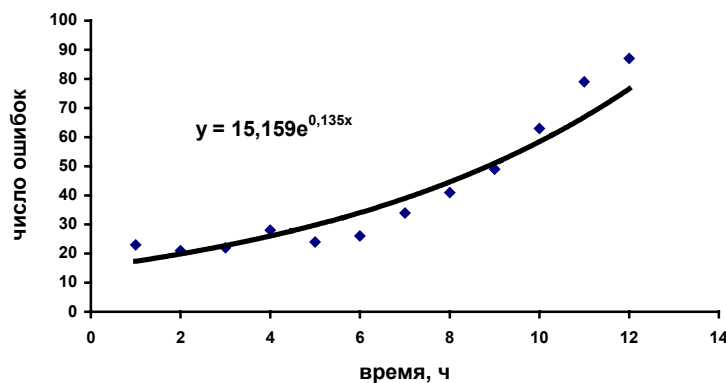


Рис. 3. Динамика ошибок при изменении времени работы

Совершение ошибок будет вынуждать программиста к их исправлению, что при значительной продолжительности рабочего дня будет приводить к заметной потере эффективности процесса создания программ. По этой причине продолжительность работ, отличающихся сложностью и требующих постоянного внимания, к которым относится работа программиста, не должна быть велика, даже если возникает стремление сократить за счет этого сроки завершения проекта программного обеспечения.

Вместе с тем нами не рассматривался вопрос о характере ошибок, их классификации, влиянии конкретных ошибок на темпы создания программного продукта ввиду сложности таких исследований.

Анализ ошибок, совершаемых программистом за аналогичный для всех вариантов период времени, относящийся к промежутку наиболее высокой работоспособности человека, примерно соответствующей середине рабочего дня обычной продолжительности показывает, что количество таких ошибок сравнительно невелико для варианта программы, использующего конструктор с ключевым словом для разрешения конфликтов пространства имен, наименьшим же их числом характеризуется первый вариант программы, включающий метод внутри класса, который не является оптимальной по количеству знаков (хотя и близкий к таковому), но отличающийся простотой (рис. 4).

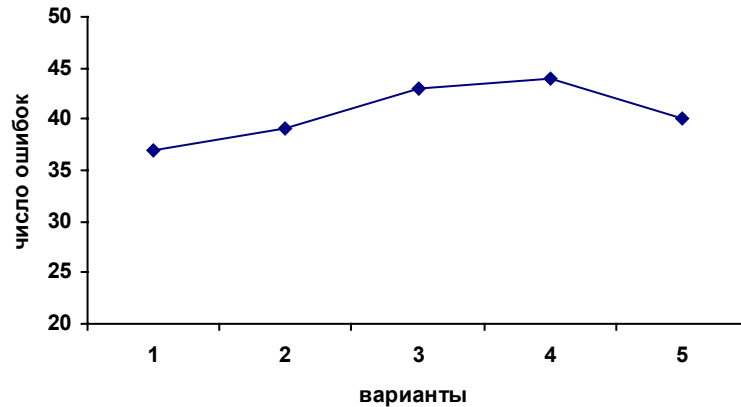


Рис. 4. Число помилок по варіантам кода програм

Выводы. В менеджменте проектов программного обеспечения важнейшим условием разработки эффективных программ является учет различных аспектов взаимодействия человека и машины. Для высокой эффективности создания программного обеспечения необходимо обеспечение оптимальных параметров всей совокупности факторов, влияющих на такую эффективность.

В ходе исследования разработан и проведен анализ ряда компьютерных программ для автоматизации вопросов изучения организации на языке Java в интегрированной среде Eclipse, использующих параметризованные методы. Анализ показал, что с точки зрения относительной простоты кода и вероятности ошибок программистов представляется использование типа программы, включающей метод внутри класса. Число ошибок, совершаемых программистами исследуемой группы, в целом росло по мере увеличения продолжительности работы. Общий характер зависимости числа ошибок от времени работы может быть с удовлетворительной точностью описан экспоненциальной функцией.

Дальнейшие исследования вопросов менеджмента проектов программного обеспечения целесообразно распространить на более детальный учет различных аспектов взаимодействия человека и машины.

Список использованной литературы

1. Фатрелл Р. Т. Управление программными проектами: достижение оптимального качества при минимальных затратах / Р. Т. Фатрелл, Д. Ф. Шафер, Л. И. Шафер ; пер. с англ. – Москва : Вильямс, 2003.
2. Брауде Э. Технология разработки программного обеспечения / Э. Брауде ; пер. с англ. – Санкт-Петербург : Питер, 2004.
3. Липаев В. В. Методы обеспечения качества крупномасштабных программных средств / В. В. Липаев. – Москва : РФФИ СИНТЕГ, 2003.
4. Гецци К. Основы инженерии программного обеспечения / К. Гецци, М. Джазайери, Д. Мандриоли ; пер. с англ. – Санкт-Петербург : БХВ-Петербург, 2005.

Статья поступила в редакцию 22.08.2014.

Харін А., Кокін А. Ю., Усайте В. І. Аналіз взаємодії людини й машини в менеджменті проектів програмного забезпечення

У менеджменті проектів програмного забезпечення найважливішою умовою розробки ефективних програм є врахування різних аспектів взаємодії людини й машини. Питання такої взаємодії розглянуто на прикладі автоматизації задач організації та управління в гірничій справі. У ході досліджень проведено аналіз поведінки людини при розробці ряду комп'ютерних програм мовою Java.

Ключові слова: аналіз, взаємодія, людина, машина, менеджмент, проект, програмне забезпечення, метод, об'єкт, клас, параметри.

Kharin S., Kokin A., Usayte V. Analysis of the Interaction Between Man and Machine in the Management of Software Projects

In project management software essential to the development of effective programs is the account of the various aspects of human-machine interaction. The issues of such interaction are considered as an example for automation, organization and management of villas in mining. The studies analyzed human behavior in the development of a number of computer programs in Java.

Key words: analysis, interaction, man machine, management, project, software, method, object, class, parameters.