

после чего с использованием правила $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{a} \cdot \mathbf{c}) - \mathbf{c} \cdot (\mathbf{a} \cdot \mathbf{b})$ и с учетом того, что $\int \dot{\omega}(t) dt = \dot{\alpha}$, приближенно представлен в виде

$$\sum_{\text{по катионам}} m_i \cdot (\langle \dot{r}_i \rangle \times (\dot{r}_i(t) - \dot{r}_i(0))) = \sum_{\text{по катионам}} m_i \cdot (\dot{\alpha} \cdot \langle \dot{r}_i^2 \rangle - \langle \dot{r}_i \rangle \cdot (\langle \dot{r}_i \rangle \cdot \dot{\alpha})). \quad (2)$$

Соотношение (2) представляет собой систему трёх линейных уравнений, которую использовали для получения вектора $\dot{\alpha}$.

При временах эволюции модельной системы порядка 1 мкс требовалось уточнение угла поворота с учётом диффузионного перемещения катионов, не являвшегося твердотельным вращением. Для этого использовали метод Нелдера-Мида [2]. В качестве целевого параметра выступала средняя плотность атомов в узлах сетки, в идеальном случае равная единице. Переменными метода были три угла, на которые необходимо повернуть сетку, чтобы она совпала с кристаллической решеткой.

Результаты работы получены в рамках выполнения государственного задания Минобрнауки России 1.4539.2017/8.9.

1. Хокни Р., Иствуд Дж., Численное моделирование методом частиц, Мир (1987).
2. Nelder J.A., Mead R., Computer Journal, 7, 308—313 (1965).

МОДЕЛЬ УПРАВЛЕНИЯ УРОВНЕМ КАЧЕСТВА ПРОГРАММНЫХ ПРОДУКТОВ

Данилин А.О., Петрухнова Г.В.

Воронежский государственный технический университет, г. Воронеж, Россия
E-mail: gvpetruhnova@mail.ru

MODEL OF MANAGING THE LEVEL OF QUALITY OF SOFTWARE PRODUCTS

Danlin A.O., Petrukhnova G.V.

Voronezh state technical university, Voronezh, Russia

Annotation. The article considers the questions reveal the quality of software products based on international standards. Discusses the model of managing the level of software quality.

Управление качеством программных продуктов при разработке программного обеспечения (ПО) актуально. Снижение качества ПО влечёт за собой истощение трудозатрат, экономических и стратегических ресурсов, снижение привлекательности продукта на рынке и повышение рискованности проекта.

Предотвратить такие последствия можно путём масштабирования совокупности процессов контроля и управления качеством ПО.

Модель состоит из 6 этапов проектирования уровня качества (УК) и трёх-этапной верификации полученных данных. Этапы могут сопрягаться с бизнес-процессами организации или позиционироваться самостоятельно.

Этап 1П. Формализация УК строится на основе многомерной иерархической структуры множеств, отражающих 6 основных требований к качеству в соответствии со стандартами серии ИСО. К ним добавляются уникальные для проекта требования к качеству, изложенные в техническом задании, и назначаются приоритеты требований к разработке.

Этап 2П. В зависимости от типа выявления УК (автоматизированного или посредством правил, выполняемых экспертами) собранные на предыдущем этапе требования распределяются по категориям, обозначенным стандартами качества.

Результатом этапов 3П и 4П является создание двух прототипов качества: основного (который может изменяться в ходе реализации проекта) и эталонного (неизменного до сдачи проекта в эксплуатацию).

Этап 5П. Реализация технических процессов с целью обеспечения верификации УК и синхронизации основного прототипа качества с эталонным.

Этап 6П. Данные для формирования требуемого УК систематизируются и шаблонизируются на случай возврата к предыдущим этапам в рамках предстоящей разработки ПО. На карте мониторинга (контроля изменения УК) выставляется начальный диапазон значений УК, исходя из выявленных эталонных показателей.

По завершении этапа 6П формируются данные для тестирования ПО и верификации УК. Этапы верификации УК следующие.

Этап 1В. При проверке соответствия значений параметров моделей основного прототипа качества с эталонным выполняется проверка УК и полученные данные импортируются на карту мониторинга.

Этап 2В. Выявление возможных отклонений от рассчитанного значения УК.

Этап 3В. Аудит УК, обеспечивающий гибкость разработки и допускающий модификацию начальных требований к УК исходя из особенностей реализации и эксплуатации.

Потребность в результатах этапов модели определяется для каждого проекта индивидуально. Модель может использоваться как при итеративной разработке ПО, так и в рамках иных распространённых методологий разработки.