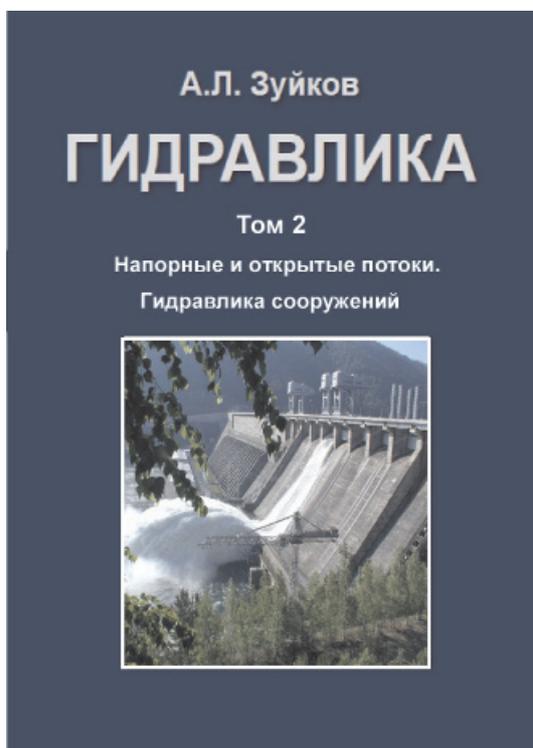


жидкости в простых и сложных трубопроводах с учетом потерь напора по длине и местных гидравлических сопротивлений; истечения жидкости из отверстий и насадков; равномерного, неравномерного и неустановившегося течения в открытых каналах и руслах, расчету гидравлического прыжка, движения двухфазных сред и грунтовых вод.



В Томе 2, Части 2. «Гидравлике сооружений» рассматриваются методы гидравлических расчетов основных гидротехнических сооружений и сопряжения бьефов, в том числе, водосливов, водосбросных плотин с уступом, водобойных и сопрягающих сооружений, шахтных и сифонных водосбросов, судоходных шлюзов, а также современные средства измерений гидравлических характеристик потоков в натуральных сооружениях и на моделях.

Учебник предназначен для студентов всех уровней, форм и профилей подготовки в высших технических учебных заведениях. Учебник полезен для аспирантов, инженерно-технических и научных работников в области гидравлики и механики жидкости.

Учебнику присвоен гриф «Рекомендовано Учебно-методическим объединением вузов РФ по образованию в области строительства в качестве учебника для студентов высших учебных заведений, обучающихся по направлению подготовки 270800 – «Строительство» (Письмо Минобрнауки России, УМО вузов РФ по образованию в области строительства от 10.04.2014 № 102-15/819).



Учебник награжден Грамотой III Уральского межрегионального конкурса «Университетская книга» (Екатеринбург) в номинации «Лучшее учебное издание в области строительства» и Дипломом участника 29 Международной книжной выставки – ярмарки Москва, ВДНХ 07–11 сентября 2016 года.

### ТЕСТИРОВАНИЕ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ КОМПЬЮТЕРНОЙ АЛГЕБРЫ (учебное пособие)

Ломакина Л.С., Вигура А.Н.

*Нижегородский государственный технический университет им. Р.Е. Алексеева, Нижний Новгород,  
e-mail: llomarina@list.ru*

В настоящее время в области информационных технологий и построения программ все большее место занимает построение крупномасштабных программных средств, обладающих

мощными интеллектуальными возможностями. Соответственно росту сложности программ возрастает количество выявляемых и остающихся в них дефектов и ошибок.

С технической точки зрения тестирование заключается в выполнении приложения на некотором множестве исходных данных к установить соответствие различных свойств и характеристик приложения заказанным свойствам. Как одна из основных фаз процесса разработки программного продукта (Дизайн приложения – Разработка кода – Тестирование), тестирование характеризуется достаточно большим вкладом в суммарную трудоемкость разработки продукта. Широко известна оценка распределения трудоемкости между фазами создания программного продукта: 40–20–40%, из чего следует, что наибольший эффект в снижении трудоемкости может быть получен прежде всего на фазах Design и Testing. Задачей ближайшего будущего является движение в сторону такого распределения трудоемкости (60–20–20%), чтобы суммарная цена обнаружения большинства дефектов стремилась к минимуму за счет обнаружения преимущественного числа ошибок программы на наиболее ранних фазах разработки программного продукта.

Тестирование осуществляется на заданном заранее множестве входных данных  $X$  и множестве предполагаемых результатов  $Y_{\text{эт}}$  –  $(X, Y_{\text{эт}})$ , которые задают график желаемой функции. Определяется соответствуют ли выходные данные –  $Y_{\text{вых}}$  (вычисленные по входным данным –  $X$ ) желаемым результатам –  $Y_{\text{эт}}$ , т.е. принадлежит ли каждая вычисленная точка  $(x, y_{\text{вых}})$  графику желаемой функции  $(X, Y_{\text{эт}})$ . При выявлении  $(x, y_{\text{вых}})$  не принадлежащей  $(X, Y_{\text{эт}})$  запускается процедура исправления ошибки, которая заключается во внимательном анализе (просмотре) протокола промежуточных вычислений, приведших к  $(x, y_{\text{вых}})$ .

Тестирование программного обеспечения представляет собой процесс или последовательность процессов, целью которых является нахождение ошибок в программе. Программа должна вести себя предсказуемо. Не следует, однако, думать, что целью тестирования является демонстрация того, что программа не содержит ошибок. Практически любая программа (кроме самых простых) содержит ошибки, и если тест их не выявил, то такой тест можно считать неудачным. Наоборот, целью тестирования является выявление ошибок в предположении, что они в программе есть. Однако, выявить с помощью тестирования все ошибки, особенно в сложной системе, практически нереально.

Тестирование в случае программ предполагает, что программа выполняется для некоторых исходных данных (тестового набора), и выходные данные сравниваются с эталонными. Если они не совпадают, в программе присутствует ошибка, и ее уже можно найти, так как мы знаем

набор исходных данных, который ее провоцирует. На первый взгляд все кажется простым, однако вскоре возникает вопрос о том, как же выбирать тестовые наборы. Решение этой проблемы сильно зависит от тестируемой программы и от целей ее разработки. Желательно, чтобы выбираемые тестовые наборы давали наибольшую вероятность обнаружения ошибки.

Основная проблема тестирования – определение достаточности множества тестов для истинности вывода о правильности реализации программы, а также нахождения множества тестов, обладающего этим свойством. Задача о выборе конечного набора тестов  $(X, Y_{\text{эт}})$  для проверки программы в общем случае неразрешима. Поэтому для решения практических задач остается искать частные случаи решения этой задачи.

В настоящей работе рассмотрены подходы к автоматизации решения различных задач тестирования программных систем, причем основное внимание уделено структурному тестированию на граф-моделях программ, выполняемому на ранних стадиях жизненного цикла (модульному тестированию и интеграционному дизайн-тестированию).

Рассмотрены следующие проблемы:

- Выбор множества тестовых путей, обеспечивающего заданное тестовое покрытие.
- Автоматическая оценка полноты тестирования.
- Инструментирование и динамический анализ программ.
- Генерация тестовых воздействий на основе алгебраических моделей программ и символического выполнения.

В первой главе рассматриваются существующие методы верификации программного обеспечения, их достоинства, недостатки и применимость. В сравнении приводятся функциональный и структурный подходы к тестированию программ.

Вторая глава посвящена тестированию программ на основе граф – моделей. Приведены используемые в рамках структурного тестирования граф-модели программ, основанные на них критерии полноты тестирования и методы выбора тестовых путей и тестовых воздействий, особое внимание уделено минимальным покрытиям управляющим графам и их использованию с целью выбора минимально грубого множества тестов.

В третьей главе рассматривается проблема определения полноты тестирования на различных его этапах – как при тестировании на этапе разработки, так и при верификационном тестировании. Приводятся существующие подходы к оценке полноты тестирования, основанные на инструментировании тестируемой программной системы.

В четвертой главе рассматриваются технические аспекты автоматизации тестирования – состав систем автоматизации, методы выбора тестовых воздействий.

В пятой и шестой главах рассматривается авторский подход к автоматизации модульного и дизайн-тестирования программных систем, основанный на алгебраической модели программы и динамическом символьном выполнении.

Книга предназначена для студентов направления «Информатика и вычислительная техника», магистрантов, обучающихся по программе «Диагностические и информационно-поисковые системы», аспирантов и специалистов в области разработки и обеспечения качества программного обеспечения.

### **ЧЕЛОВЕКО-КОМПЬЮТЕРНОЕ ВЗАИМОДЕЙСТВИЕ (учебное пособие)**

Магазанник В.Д.

*Московский государственный технический  
университет им. Н.Э. Баумана, Москва,  
e-mail: V.Magaz@tushino.com*

Гриф УМО вузов РФ по университетскому политехническому образованию.

Рассматриваемое учебное пособие закрывает давно существующий пробел в учебных материалах по человеко-компьютерному взаимодействию. Значительный рост числа пользователей компьютерами в последние годы выявил множества проблем, связанных с недостаточным учетом «человеческого фактора» при разработке интерфейсов и, в целом, всей системы взаимодействия «человек-компьютер». Особенно актуально это направление для пользователей-любителей, а именно эта категория пользователей и составляет большинство, формирующее, помимо прочего, рыночную судьбу программных продуктов. Этим и объясняется, что в так наз. развитых странах на факультетах компьютерных наук в обязательном порядке присутствует весьма объёмный (обычно, годовой или 1,5-годовой) курс по человеко-компьютерному взаимодействию.

В учебном пособии представлен весь спектр вопросов человеко-компьютерного взаимодействия. Учебное пособие является дополненным и переработанным изданием вышедшего в 2007 г. в изд. Логос учебного пособия этого автора под таким же названием. Многие разделы переработаны, улучшена структура, большинство разделов дополнены новым материалом. Автор собрал и обобщил большое количество публикаций, материалов конференций, периодики, сведений, размещенных на соответствующих сайтах и форумах, касающихся человеко-компьютерного взаимодействия, использованы также учебные программы ряда зарубежных университетов.

Рассмотрены требования к компоновке компьютеризированных рабочих мест и планировке офисных помещений, а также к мониторам и средствам ввода информации.

Много внимания уделено подходам и методам описания характеристик потенциальных пользователей, правилам и процедурам построения их профилей и последующей сегментации. Изучение этой темы для студентов компьютерных специальностей будет особенно полезно, ибо учит с самого начала разработки любого программного продукта подробно представлять себе будущего пользователя, тщательно его описывать и постоянно держать его образ в голове.

Основное место отведено подходам, методам и инструментарию разработки, оценки, тестирования и прототипирования пользовательских интерфейсов. Подробно описываются показатели, методы и процедуры оценивания интерфейсов на каждой стадии разработки программного продукта или сайта.

Центральный аспект оценки интерфейса – периодическое юзабилити-тестирование изложено достаточно полно. Важно, что показано место этого тестирования в общей процедуре оценки интерфейса, избежав частой ошибки его преувеличения, но в то же время и продемонстрировать действительное значение тестирования такого рода. Особое внимание уделено вариантам и инструментарию создания навигационных структур, что наиболее актуально для web-сайтов. Во всех разделах содержится не только повествовательный материал, но также инструменты и практические приемы создания описываемых объектов интерфейса.

Построению прототипов интерфейса посвящен большой раздел, и это не случайно. Освещены средства итерационного прототипирования, рассмотрены виды прототипов, программные пакеты для их создания. В целом, процедуры такого рода носят универсальный характер (макетирование технических систем на стадиях разработки известно давно, но обычно это довольно дорогостоящая процедура), особенность же программных систем и существующие программные пакеты построения прототипов вносят много специфики и делают прототипирование захватывающим творческим процессом, образующим вместе с юзабилити-тестированием стержень разработки пользовательского интерфейса.

Особое внимание уделено возможностям и правилам использования мультимедиа в метафорических конструкциях. Следует отметить, что столь обширного освещения возможностей и правил работы с мультимедиа-приложениями в учебной литературе не было.

Отдельная тема посвящена базовым принципам дизайна интерфейса и типовым элементам интерфейса. Изложение базовых принципов дизайна интерфейса сопровождается примерами, описаниями и, что наиболее важно, средствами реализации и наличными ресурсами в Интернете. Типовые элементы интерфейса рассмотрены