

УДК 621.391

ОСОБЕННОСТИ ТЕСТИРОВАНИЯ ПРИЛОЖЕНИЙ ИНТЕРНЕТА ВЕЩЕЙ

И. В. Худоев¹, Р. Я. Пирмагомедов²

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича:

¹аспирант кафедры сетей связи и передачи данных, khvanches@gmail.com

²кандидат технических наук, доцент кафедры сетей связи и передачи данных, its.pto@yandex.ru

Аннотация. В статье рассматриваются особенности тестирования устройств и приложений Интернета Вещей. Проанализирована возможность применения существующих средств и методов тестирования для приложений Интернета Вещей. Приведен набор требований к тестированию, которые необходимо принимать во внимание при планировании и организации процесса тестирования приложений Интернета Вещей.

Ключевые слова: тестирование, Интернет Вещей, медицинские приложения.

Hudoev I., Pirmagomedov R.

Test Features of the Internet of Things Applications

Abstract. The article discusses the features of testing devices and applications of the Internet of Things. The possibility of using existing tools and test methods for the Internet of Things applications. A set of testing requirements that must be taken into account in the planning and organization of testing IoT applications.

Key words: testing, Internet of Things, medical applications.

Введение

Интернет Вещей позволяет интегрировать физические и виртуальные объекты. Виртуальная реальность, которая недавно была доступна только на экране монитора, сегодня интегрируется с реальным миром, предоставляя пользователям совершенно новые возможности. Пользователь получает возможность взаимодействовать с объектами на другом конце света и получать необходимые услуги, которые стали реальностью благодаря взаимодействию вещей [1].

На рынке представлено множество решений в области Интернета Вещей, начиная от готовых приложений, и заканчивая открытыми платформами, которые пользователь может сконфигурировать и настроить так, как ему необходимо (*Arduino, Intel Galileo, ESP* и др.).

Стремительный рост и многообразие различных решений привели к ряду проблем, как в области коммуникационного взаимодействия Интернет Вещей между собой, так и безопасности этих устройств для окружающей среды, и человека. На текущем этапе развития технологий Интернета Вещей все более актуальным становится вопрос испытания и сертификации приложений и устройств Интернета Вещей, прежде чем они будут готовы для общественного потребления. Особенно остро, стоит проблема безопасности. «Дыры» в безопасности отдельных приложений Интернета Вещей позволяют злоумышленникам получить доступ не только к личным данным,



но и организовать полноценную слежку за пользователем, и даже оказывать непосредственное физическое воздействие на здоровье. Другим аспектом этого вопроса, является нерациональность архитектуры и технологий использованных при разработке приложений Интернета Вещей, а также качество самих контроллеров, используемых в устройствах. Проблемы подобного рода приводят к быстрому износу датчиков, перегрузке контроллеров, уменьшают срок службы приложений Интернет Вещи, приводят к удорожанию стоимости, поломкам и искажениям данных [3]. Особенно остро вопрос о качестве и надежности приложений Интернета Вещей стоит в том случае, когда от работы данного приложения зависят жизнь и здоровье человека [4, 5].

В данной статье рассматриваются основные аспекты тестирования приложений Интернета Вещей, с акцентом на приложения, используемые в медицине. Проанализируем их общность и специфичность с традиционными моделями клиент-серверных приложений.

Архитектура приложений Интернета Вещей

В отличие от ранее известных клиент-серверных приложений, приложения Интернета Вещей плотно интегрированы с физическим миром. Рассматривая архитектуру, подобных приложений [6], как сложную систему (рис. 1) можно выделить следующие составляющие подсистемы и уровни:

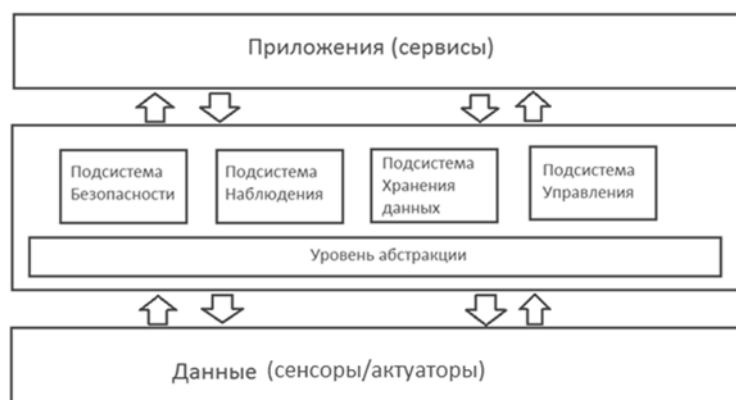


Рис. 1. Архитектура приложения

может явиться границей между пользователем и системой. Что приводит нас к необходимости подробного исследования пользовательского опыта, его ожиданий и предугадывания желаний.

Уровень управления. Базисом уровня управления является подуровень абстракции – прослойка необходимая для нивелирования недостатков и особенностей отдельных компонентов уровня данных и предоставления вышестоящим подсистемам универсального интерфейса для общения и управления контроллерами в физическом мире. Подсистема безопасно-

Уровень данных преимущественно состоит из контроллеров с сенсорами и/или актуаторами расположенных в физическом мире. Следует учесть, что уровень данных является не только точкой соприкосновения системы с физическим миром, сам по себе, но также



сти – необходима для аутентификации пользователей системы и идентификации оборудования, подключаемого к системе, проведения шифрования потоков данных. Подсистема хранения данных – обеспечивает хранение данных с сенсоров, для их дальнейшей обработки и анализу приложений третьего уровня. Подсистема мониторинга – подсистема обеспечивающая контроль за состоянием периферийных устройств, оперативно реагирующая на поломки и сбои в работе. Подсистема управления – обеспечивает управление потоками данных от нижнего уровня, контролирует заданные параметры качества обслуживания.

Уровень приложений – уровень, который предоставляет пользователю заявленные поставщиком сервисы с определенным параметром качества обслуживания. Может быть реализован как в виде мобильных приложений, так и в виде веб сервисов.

В процессе разработки рассмотренной системы, тестирование включает в себя тестирование оборудования, программного обеспечения, анализ интерфейсов взаимодействия и потока данных в режиме реального времени. Приступить к тестированию стоит с момента описания архитектуры, конкретных задач и способов их решения и продолжаться во время эксплуатации систем, как для оценки рисков по введению новых компонентов и сервисов, так и для поиска путей оптимизации работы уже имеющейся системы.

Тестирования приложений Интернета вещей

Модель тестирования сервис-ориентированной архитектуры, которая была описана выше, должна проходить в соответствии с моделью жизненного цикла программного обеспечения [7].



Рис. 2. Модель разработки сервисно-ориентированной программного обеспечения

В соответствии с сервисно-ориентированной моделью разработки программного обеспечения (рис. 2) будет построена и модель тестирования с учетом особенностей Интернета Вещей, обеспечения необходимого качества обслуживания, организации безопасности системы и сокращения времени разработки до выхода готового продукта на рынок. Стоит она из следующих фаз.

Фаза моделирования (*Modelling*). В фазе моделиро-



вания, акцент делается на создании службы работающих в соответствии с функциональными характеристиками. Таким образом, основное внимание необходимо уделять функциональному тестированию, т. е. модульным и интеграционным тестам.

В фазе сборки (*Composition*), основное внимание уделяется созданию простейших услуг, составляющих другие услуги. На этом этапе, становится все более важным, обнаружить простейшие услуги, необходимые для достижения целей составной сервис. В этой фазе, необходимо реализовать тесты, которые эффективно представляют составной сервис. Подобные тесты должны подтвердить, что основные услуги и сервисы доступны или могут быть адекватно эмулированы.

В фазе развертывания (*Deployment*), как правило происходит частичный ввод в эксплуатацию ряда сервисов. Для того, чтобы иметь возможность оценить успешность ввода в эксплуатацию, необходимо проверить сервисы на наличие несоответствий, противоречий и дублирования. В центре внимания данной фазы должно стать нагрузочное тестирование, оценивающее могут ли развернутые сервисы предоставлять услуги с заявленным качеством обслуживания.

В фазе управление (*Execution*) необходимо проводить стрессовое тестирование для проверки поведения системы в случае возможных перегрузок, а также тестов по определению максимальной емкости отдельных компонентов, интегрированных в систему. Полученные результаты могут быть использованы для настройки сигналов тревоги, которые срабатывают, если качество обслуживания находится под угрозой

В рассмотренной модели тестирования приложений Интернета Вещей ведущую роль играет уровень автоматизации тестирования. Это вызвано необходимостью провести большой объем работ, и воссоздать ситуации, которые связаны с риском для реального периферийного оборудования. Для имитации ситуаций подобного рода используют виртуальные эмуляторы. Еще один важный аспект – это выбор средств мониторинга за элементами инфраструктуры Интернета Вещей. Перед тем как выбрать инструмент, стоит проанализировать на сколько сильно влияет работа самих систем мониторинга на работу контролеров.

Инструменты тестирования

Автоматизация тестирования играет ведущую роль в организации процесса тестирования приложений Интернета Вещей. Некоторые подходы, такие как UML 2.0 Testing Profile предоставляют возможность для проектирования и разработки тестов методом черного ящика, но не дают указаний как использовать данный профиль. Абстрактный UML 2.0 Testing Profile (U2TP) создает лишь план действий, который в последующем должен быть реализован на соответствующем языке программирования. Для тестирования сер-



висов в этом случае используются TTCN-3 [9] или JUnit [10]. TTCN-3 определяет реализацию тестовых сценариев и метод их исполнения. Существует несколько инструментов обеспечения условий для создания тестов и их исполнение. Взаимодействие непосредственно с контролерами реализуется средствами C/C++, Python. Все перечисленные выше технологии требуют высокий пороговый уровень вхождения. Далеко не каждый инженер тестирования обладает подобными навыками, а тратить время на обучения вышеперечисленным технологиям и языкам программирования достаточно затратно как по времени, так и по финансовым аспектам.

Для тестирования физического уровня сети можно использовать EuWin Platform, которая предоставляет возможности работы со многими видами реализаций физического уровня Интернета Вещей. Помимо него, важным инструментом в проектировании и тестировании физического расположения устройств является эмулятор EEBuildingSim, который позволяет эмулировать поведение устройств в зависимости от их пространственного расположения.

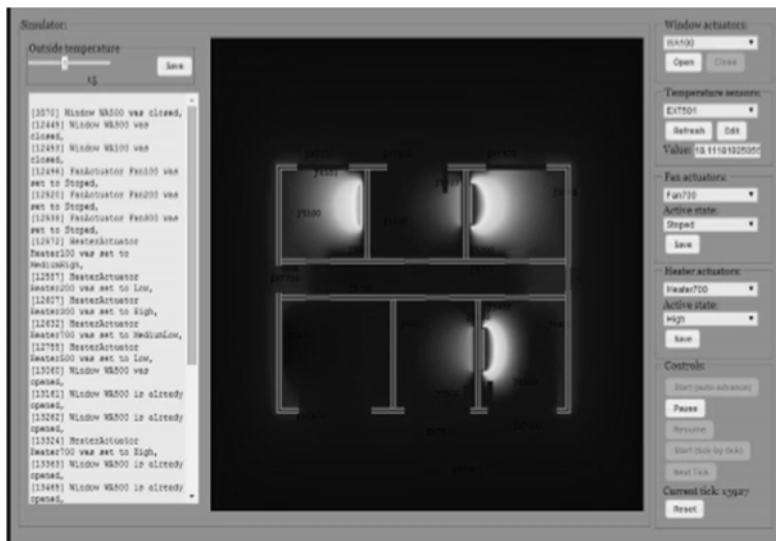


Рис. 3. Пользовательский интерфейс программы EEBuildingSim

имеет интеграцию с тестовым окружением, а также со средой исполнения продукта. Разработка тестов поддерживается тест-компонентой, которая основана на инструменте TTWorkbench. Простейшие и составные сервисы интегрированы в SCE, которая запускает процесс тестирования, как только составной сервис готов к развертыванию.

Особенности тестирования медицинских приложений

Применительно к медицинским приложениям Интернета Вещей необходимо выдвинуть еще ряд дополнительных требований к тестированию, ввиду специфики их применения:

- 1) Удобство использования.

Для проведения тестирования сервисов существуют решения по графическому созданию тестов. Примером такого решения является Service Composition Environments (SCE) – платформа, которая предоставляет графический интерфейс редактирования бизнес-процессов и связанных с ними услуг. Данная среда



Тестирование удобства использования должно проводиться на целевых группах. Внимание в данном случае следует уделять эстетическим вопросам использования периферийных устройств. Если предполагается размещение устройств непосредственно на теле человека, то они должны гармонично входить в повседневную жизнь, являясь частью одежды или неприметным дополнением, аксессуаром, который не сковывает движение, не мешает в работе. Целью данного вида исследования будет достижение ситуации, при которой пользователь не будет ощущать изменения, связанных с введением в его жизнь, устройств Интернета Вещей.

2) Анализ и тестирование экологичности.

Любое устройство, расположенное в непосредственной близости с человеком, потенциально может оказать негативное воздействие на жизнь и здоровье, как в долгосрочной перспективе, так и в экстремальных ситуациях. А, следовательно, пользователь должен иметь возможность деактивировать устройство. Помимо этого, паразитное воздействие устройства должно быть сведено к минимуму. Расположенные на теле устройства не должны вызывать раздражений, являться причиной кожных заболеваний.

3) Устойчивость элемента к воздействиям окружающей среды.

Как устройство влияет на человека, так и человек с окружающей средой влияет на устройство. Следует учитывать особенности среды, в которой будет развернуто устройство, действий, которые человек будет с ним производить.

4) Особенности нагрузочного тестирования

В отличие от веб- и мобильных приложений, расположенные в физической среде устройства, могут не реагировать на действия пользователя. В таком случае любая ошибка или искажение может привести к серьезным последствиям и навредить здоровью человека. Исключением подобных ситуаций и гарантом стабильной работы приложения является нагрузочное тестирование. При его проведении необходимо уделять внимание таким параметрам как полоса пропускания сети, задержки в сети [8], искажения, потери пакетов, проверка механизмов резервирования как сети, так и системы. Проводить нагрузочное тестирование следует и после ввода системы в эксплуатацию. Минимально оно должно быть представлено в наблюдении и сборе данных о состоянии системы. После сбора данных о состоянии системы их необходимо проанализировать, составить прогнозы о жизнеспособности системы и в случае необходимости запланировать ввод в эксплуатацию дополнительных компонентов системы или усиление старых.

Заключение

Для обеспечения качественной и бесперебойной работы приложений Интернета Вещей требуется переосмысление существующих методов тестирования клиент-серверных приложений. Новые методы тестирования должны не только выявлять недостатки в разрабатываемых системах,



но и опережать и даже предугадывать потребности потребителей. Учитывая, что приложения Интернета Вещей переходят от развлекательной нацеленности на решение задач и предоставление услуг, носящих жизненно важный характер (медицинские приложения), на плечи тестировщиков ложится ещё большая ответственность.

Список используемых источников

1. Кучерявый А. Е., Прокопьев А. В., Кучерявый Е. А. Самоорганизующиеся сети. СПб.: Любавич, 2011. 312 с.
2. Maurizio A. Spirito and Maria T. Delgado Internet of Things Application Scenarios, Pilots and Innovation // Communications Building the Hyperconnected Society IoT Research and Innovation Value Chains, Ecosystems and Markets. 2015. Vol. 43. pp. 119–143.
3. Никитин Б. К., Пирмагомедов Р. Я. Надежность сегментов сети доступа и характеристики волоконно-оптического кабеля // Информатизация и связь. 2012. № 9. С. 26–29.
4. Пирмагомедов Р. Я., Худоев И. В. Обзор тенденций развития наносетевых технологий в медицине // Актуальные проблемы в науке и образовании. V Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 3 т.; Т. 1 / под ред. С. В. Бачевского. СПб. : СПбГУТ, 2016. С. 508–512. URL: <https://mail.google.com/mail/u/0/#inbox?compose=158d2d09b569891e>
5. Пирмагомедов Р. Я., Кучерявый Е. А., Глушаков Р. И., Киричек Р. В., Кучерявый А. Е. Живые организмы в киберпространстве проект «Биодрайвер» // Электросвязь. 2016. № 1. С. 47–52.
6. Eike S. Reetz, Daniel Kümper and Ralf Tönjes Anders Lehmann Test Driven Life Cycle Management for Internet of Things based Services: a Semantic Approach // VALID 2012 : The Fourth International Conference on Advances in System Testing and Validation Lifecycle, 2012. pp. 21–27.
7. Ralf Tönjes, Eike Steffen Reetz, Klaus Moessner, Payam Barnaghi A Test-driven Approach for Life Cycle Management of Internet of Things enabled Services // Future Network and Mobile Summit 2012 Conference Proceedings Paul Cunningham and Miriam Cunningham (Eds) IIMC International Information Management Corporation, 2012. pp. 1–8.
8. Pirmagomedov R., Hudoev I., Glushakov R., Kirichek R., Koucheryavy A. Analysis of Delays in Medical Applications of Nanonetworks // 2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). pp. 80–86.
9. Internet of Things Environment for Service Creation and Testing PROJECT FINAL REPORT. URL: <http://ict-iotest.eu/iotest/>.
10. The testing and test control notation version 3 (ttn-3). European Standard 201874, 2002/2003.

