УДК 004.415.53
JEL C88

ПРИМЕНЕНИЕ ДЛЯ IOT МЕТОДИКИ ВЫБОРА ПОДХОДА К ТЕСТИРОВАНИЮ ПРОГРАММНОГО ПРОДУКТА В РАМКАХ ПРОЦЕССА ЦИФРОВОЙ ТРАНСФОРМАЦИИ ПРОМЫШЛЕННОСТИ

Галимова Е. Ю.
ассистент
Санкт-Петербургский государственный
университет промышленных технологий и дизайна;
старший тест-инженер ООО «Октавиан.Спб»
г. Санкт-Петербург

Аннотация: Интернет вещей заключает в себе огромный потенциал для развития разработки и тестирования программных продуктов, так как включает в себя различные комбинации аппаратного и программного обеспечения, множество различных типов устройств, необходимость оперативного реагирования на запросы, поддержку больших объёмов и разнообразия типов данных, разнообразные протоколы, сенсорные взаимодействия. В данной статье предлагается методика выбора подкода к тестированию программного продукта IoT в концепции цифровой промышленности.

Ключевые слова: тестирование программного обеспечения, ручное тестирование, автоматизированное тестирование, качество программного обеспечения, метод выбора, интернет вещей, умное производство.

APPLICATION FOR IOT METHODOLOGIES FOR APPROACHING TESTING A SOFTWARE PRODUCT WITHIN THE DIGITAL TRANSFORMATION OF THE INDUSTRY PROCESS

E.U. Galimova
assistant
St. Petersburg State University of Industrial Technology and Design;
Senior Test Engineer, LLC «Octavian.Spb»
St. Petersburg, Russia

Annotation: The Internet of Things contains enormous potential for the development of software product development and testing, as it includes various combinations of hardware and software, many different types of devices, the need for quick response to requests, support for large volumes and a variety of data types, various protocols sensory interactions. This article proposes a method for choosing a subcode for testing the IoT software product in the concept of the digital industry.

Keywords: software testing, manual testing, automated testing, software quality, selection method, Internet of things, smart manufacturing.

Тенденции развития современной промышленности. Три промышленных революции привели к смене парадигмы в области производства – механизация с использованием воды и пара, массовое производство на сборочных линиях и автоматизация с использованием информационных технологий. Четвертая промышленная революция характеризуется внедрением в производство интернета вещей (IoT) и интернета концепций услуг, что позволяет создавать умные фабрики с

вертикально и горизонтально интегрированными производственными системами. В большинстве отраслей по всему миру внедряются очень гибкие процессы, которые можно очень быстро изменить, что позволяет индивидуализировать массовое производство [1]. Индивидуализация достигается путём использования современных элементов производства, которые способны передавать свои текущие данные на умные машины [2].

Машины получают информацию об окружающей среде, обмениваются между собой информацией и самостоятельно контролируют процессы производства и логистики. Данные собираются в больших количествах в течение всего жизненного цикла и хранятся децентрализованно для принятия локальных решений. При этом указанные данные остаются прозрачными для межмашинных взаимодействий. Такие системы называются кибер-физическими системами (cyber-physicalsystems, CPS) [3, 4]и представляют собой физические объекты (станки, транспортные средства, обрабатываемые детали), которые оснащены передовыми технологиями (RFID, датчики, микропроцессоры, встроенные системы) [1]. Кибер-физические системы характеризуются возможностью собирать данные о себе и своей среде, обрабатывать и оценивать эти данные, осуществлять подключение и взаимодействие с другими системами, инициировать выполнение определённых действий. Кроме того, в кибер-физические системы включены новые сервисы, которые могут заменить традиционные бизнес-модели, основанные исключительно на продажах продукции.

Кибер-физические системы появляются благодаря сложным сетям и интеграции встроенных систем, прикладных систем и инфраструктуры, включающей человеко-машинное взаимодействие. В отличие от традиционных систем, используемых для производства, кибер-физические системы можно рассматривать как системы систем, для реализации которых требуется сотрудничество различных дисциплин, таких как машиностроение, электротехника и информатика [5].

Интернет вещей как часть умного производства. Умное производство – это интенсивное использование данных, полученных в результате применения информационных технологий на уровне цеха и выше, для обеспечения интеллектуальных, эффективных и быстрых операций [6]. Умное производство включает в себя различные технологии, наряду с кибер-физическими производственными системами, это интернет вещей (IoT), робототехника, автоматизация, анализ больших объёмов данных и облачные вычисления [7, 8].

С внедрением умного производства не происходит линейной замены людей на искусственный интеллект и автоматизацию в цехах. Люди будут заниматься проектированием индивидуальных решений в конкретных областях.

Функционал для тестирования разрабатываемого программного обеспечения должен входить в состав любого современного инструмента информатизации промышленного предприятия. Актуальными являются тенденции на унификацию, стандартизацию и независимое подтверждение качества [9].

В то время как встраиваемые системы уже давно существуют в формате бытовой электроники, IoT дал им новую жизнь. Ранее эти системы были по существу автономными и могли работать в основном изолированно. Теперь связанные объекты должны общаться друг с другом, чтобы функционировать надлежащим образом. Следовательно, разработчики программных продуктов должны применять различные способы оптимизации взаимодействия между устройствами (device-to-device, D2D) и между устройствами, серверами и человеком (device-to-server, D2S) [10].

Языками программирования, обычно используемыми во встроенных системах, являются C и C++. Они более низкоуровневые, чем те, которые используются в веб-разработке. Код на C и C++ обычно имеет лучшую производительность во время

выполнения, но разработка программ более длительная по срокам. Код сложнее, так как требуется использовать явное управление памятью. Отсутствует переносимость кода и требуется кросс-компиляция между средой разработки и целевой средой.Независимые от предыдущих работ проекты Greenfieldотносительно редко встречаются во встроенном программном обеспечении, так как проекты часто зависят от монолитно унаследованного кода.

Парадигма IoT в большей степени ориентирована на информационно-коммуникационные технологии [11]. Цель IoTв области вычислений [12] состоит в том, чтобы соединить физический мир с виртуальным миром и облегчить связь между всеми связанными сущностями [13, 14].

IoTподразделяется на промышленный, применяемый в таких отраслях и проектах, как тяжёлое машиностроение, общественный транспорт, «Умные города» [15], здравоохранение; на пользовательский – смартфоны,термостаты, смарт-часы, телевизоры, бытовые электроприборы, системы умного дома (рис. 1).

Типы вычислений в промышленном IoT

Разработано несколько типов вычислительных архитектур промышленного IoT [16]:

*Облачные вычисления (Cloud Computing).*С помощью IoT и облачных вычислений производится отправка и обработка сенсорных данных в облаке. Есть модуль загрузки, который принимает данные и сохраняет их в очень большом хранилище, затем применяется параллельная обработка (например, Hive, Spark или Azure HD Insight). Полученная информация используется для принятия решений.

Туманные вычисления (Fog Computing). Благодаря туманным вычислениям IoT становится мощнее. Вместо того, чтобы полностью отправлять ваши данные в облако и ждать, пока сервер обработает их и ответит, используется локальный процессор или компьютер.Можно контролировать, администрировать и управлять локальной сетью периферийных устройств без внешних зависимостей от облачных серверов [17]. Граничные узлы или шлюзы с поддержкой туманных вычислений предоставляют распределённым сетям возможность локального принятия решений и временного хранения данных для анализа. Такой тип распределения гарантирует, что даже если облачные сервисы недоступны, IoT сможет функционировать локально с прогнозируемыми ограничениями. Туманные вычисления разделяют данные по признаку критичности по времени обработки. Данные с высокой степенью критичности обрабатываются локально. Применяются алгоритмы максимального использования ресурсов граничных узлов и оптимального использования пропускной способности сети.

Граничные (периферийные) вычисления (Edge Computing). Данная архитектура подразумевает сосредоточение вычислительных мощностей на периферийных устройствах [18]. Edge Computing приближают нас к источнику данных, позволяют применять машинное обучение в области датчика, использовать интеллект на сенсорных узлах. Граничные вычисления могут подключаться к облаку, но им не нужно облако для функционирования. Данная архитектура выигрывает, когда есть очень большие требования по времени выполнения запросов.

Легкотуманные вычисления (Mist Computing). Данный тип вычислений развивается как улучшение Fog Computing и Edge Computing. Мы могли бы просто использовать сетевые возможности устройств IoT и распределять рабочую нагрузку, использовать динамические интеллектуальные модели, осуществляющие облегчённые вычисления в сетевой структур5е с использованием микроконтроллеров и микросхем. Последних нет ни в Fog Computing, ни в Edge Computing. Данная архитектура работает

с туманными вычислениями и облачной платформой. Установка этой парадигмы сможет обеспечить высокую скорость обработки данных.

Особенности процесса тестирования программных продуктов IoT

Размер IoT растет с беспрецидентной скоростью. Gartner, фирма, занимающаяся исследованиями в IT-сфере, подсчитала, что к 2020 году к IoT может быть подключено более 26 миллиардов устройств [19]. В то же время растущее число трансграничных атак может подтолкнуть правительства ряда государств к выделению и закрытию своей части интернета. Государства опасаются также ослабления контроля над глобальными онлайн-компаниями.

IoT предъявляет уникальный комплекс требований к методике тестирования, так как объединяет в себе характеристики современных веб-служб и встроенных систем. Применения классических методов тестирования не достаточно. Подробное описание стандартных подходов к тестированию программного продукта приведено в работе [20].

Связность и безопасность являются основными уязвимостями в IoT. Используется широкий спектр протоколов, таких как 4GLTE, Wi-Fi, ZigBee, Bluetooth и другие, что требует дополнительных расширенный проверок в процессе тестирования. Нужно протестировать разнообразные сценарии подключения, исследовать слабые, доступные места для сетевых атак. IoT является многоуровневой системой, требующей тщательного тестирования на каждом уровне. Нижний уровень – это вычислительная инфраструктура, которая содержит приложения и аналитику. Далее уровень сетевых устройств и датчиков. На верхнем уровне располагаются локальные и глобальные средства связи, которые соединяют устройства друг с другом и с сетями, такими как Интернет или корпоративная глобальная сеть. Очень важно вручную тестировать отдельные компоненты на производительность, надёжность, совместимость и безопасность непосредственно в процессе интеграции.

Тестирование безопасности должно предотвратить взлом поставщиков облачных услуг. Отказ в работе облачных провайдеров может остановить работу сотен предприятий, организаций здравоохранения, государственных учреждений. Нужно проводить тестирование на проверку возможности Отказа от обслуживания (Distributed Denial-of-Service, DDoS) (рис.1).

Distributed denial of service атаки

DDoS-атаки считаются одними из самых простых и дешёвых. Их цель - сделать сервер не работоспособным. Иногда такие атаки являются отвлекающим фактором от других, более опасных атак, связанных с проникновением внутрь системы.

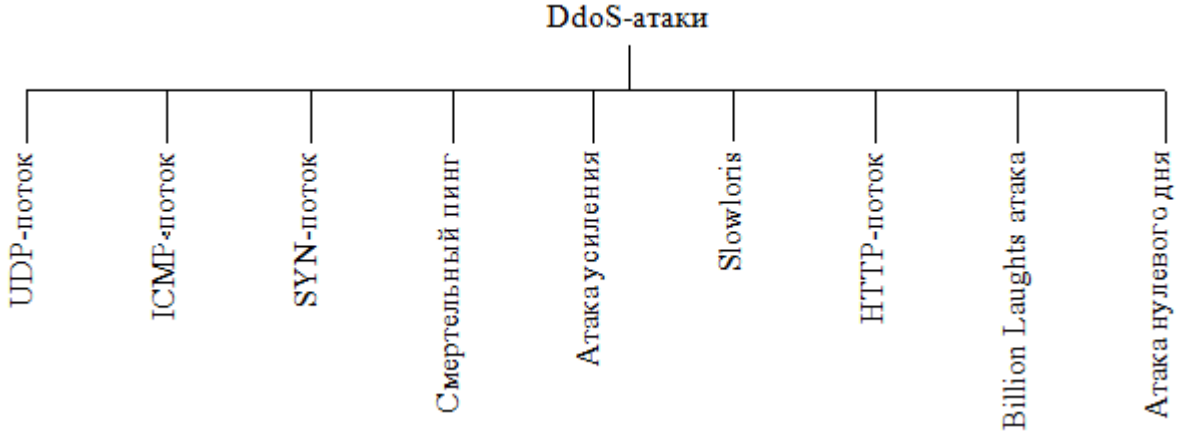

Рисунок 1 – Распространённые типы DDoS-атак.

DDoS атаки бывают нескольких типов[21]:

UDP-поток. Цель атаки – переполнить случайные порты на удалённом хосте. Для этого используется поток пакетов протокола пользовательских датаграмм (User Datagram Protocol, UDP).В ходе атаки хост многократно проверяет наличие приложение, прослушивающего этот порт, и если приложение не найдено, то отвечать пакетом ICMP 'Destination Unreachable'. Этот процесс истощает ресурсы хоста. В конечном итоге хост может стать недоступен.

ICMP (Ping) поток. Механизм аналогичен UPD-потоку. Отличие состоит в том, что Ping-поток переполняет целевой ресурс пакетами эхо-запроса (ICMP), то есть пакеты отправляются максимально быстро, не ожидая ответов. Этот тип атаки может пожирать как исходящую, так и входящую пропускную способность, поскольку пострадавшие сервера пытаются ответить пакетами эхо-ответа (ICMP), что приводит к значительному общему замедлению работы системы.

SYN-поток. Используется известная уязвимость в последовательности соединений TCP, так называемое «трёхстороннее рукопожатие». На запрос SYN, для инициирования соединения TCP с хостом, должен быть получен ответ SYN-ACK от этого хоста, который затем подтверждается ответом ACK от запрашивающей стороны. В сценарии потока SYN запрашивающая сторона отправляет несколько запросов SYN, но либо не отвечает на ответ SYN-ACK хоста, либо отправляет запросы SYNc поддельного IP-адреса. В любом случае, хост-система продолжает ожидать подтверждения для каждого из запросов, выделяя под это ресурсы до тех пор, пока не будут установлены новые соединения, что в итоге приводит к отказу в обслуживании.

Смертельный пинг. Злоумышленники отправляют на компьютер несколько искажённых пингов. Максимальная длина IP-пакета вместе с заголовком равна 65 535 байтов. Однако уровень канала передачи данных обычно накладывает ограничения на максимальный размер единовременной отправки данных. Например, 1500 байтов по сети Ethernet. В таком случае большой IP-пакет разделяется на несколько частей, затем хост-получатель повторно собирает IP-фрагменты в полный пакет. В сценарии "Смертельный пинг" производится злонамеренное манипулирование фрагментами. В результате у получателя оказывается IP-пакет большего размера, чем 65 535 байт. Это приводит к переполнению буферов памяти, выделенных для пакета, вызывает отказ в обслуживании других пакетов.

Атака усиления. При атаках с усилением NTP злоумышленник использует общедоступные серверы протокола сетевого времени (Network Time Protocol, NTP) для перегрузки целевого сервера трафиком UDP. Атака определяется как атака усиления, потому что отношение запросов к ответам в таких сценариях находится где-то между 1:20 и 1: 200 или более. Это означает, что любой злоумышленник, который получает список открытых NTP-серверов, может легко создать разрушительную DDoS-атаку с высокой пропускной способностью.

Slowloris. Целенаправленная атака, позволяющая одному веб-серверу отключить другой сервер, не затрагивая другие службы или порты в целевой сети. Slowloris делает это, удерживая как можно больше подключений к целевому веб-серверу как можно более длительное время. Это достигается путем создания соединений с целевым сервером, но отправляются только части запросов. Slowloris постоянно отправляет больше заголовков HTTP, но никогда не завершает запрос. Целевой сервер сохраняет каждое из этих ложных соединений открытым. В конечном итоге это приводит к переполнению максимального количества одновременных пулов соединений и приводит к отказу в дополнительных подключениях для реальных клиентов.

HTTP-поток. Используются GET запросы в большом количестве, отсылаемые на восьмидесятый порт. Сервер оказывается максимально загружен и не может обрабатывать остальные запросы.

*Billion Laughs атака (XML-бомба).*Нацелена на анализаторы XML-документов, так называемая экспоненциальная атака. Механизм реализации: экземпляр атаки состоит из 20 объектов, каждый из которых в свою очередь тоже состоит из 20 объектов. Экземпляр атаки снабжен документом, состоящим из одного экземпляра самого большого объекта, который в свою расширяется до одного миллиарда копий первого объекта.

DDoS-атаки нулевого дня. Этот термин применяется ко всем новым неизвестным атакам, для которых ещё не созданы исправления в коде. Этот термин хорошо известен среди хакеров, которые торгуют уязвимостями нулевого дня.

В настоящее время разрабатываются различные методики защиты от DDoS-атак, в том числе основанные на нейронных сетях [22].

Применяется в первую очередь нагрузочное и стресс-тестирование для выявления уязвимостей для DDoS-атак. Например, в приложении Apache JMeter есть функционал для имитации подключений большого количества удалённых хостов. Нагрузка может исследоваться как максимальная пропускная способность канала. Можно имитировать атаки на сетевой сервис, например, на протокол Secure Sockets Layer (SSL). Данный протокол основан на асимметричной криптографии и симметричном шифровании. На уровне тестирования самого приложения можно использовать запросы на получение больших объёмов данных, например, через функционал поиска. Другая техника тестирования – использовать подмену максимального количества записей в строке запроса. Например, предлагают найти 5 последних записей, а мы в ручную меняем в HTTP-запросе число 5 на 10000000000. Для тестирования уязвимостей для DDoS-атак можно использовать Zip-бомбу: файл выглядит как небольшой архив, мы его распаковываем и получаем такой же архив и так далее. Может распаковываться антивирусом.

Методика выбора подхода к тестированию программных продуктов IoT

Аналитик, менеджер продукта или разработчик дают ответы "да" или "нет" на предлагаемый список вопросов о требованиях к программному продукту и его характеристиках (рис 2). Данный список базируется на качественных свойствах программного продукта, подробно о которых рассказывается в работе [23].

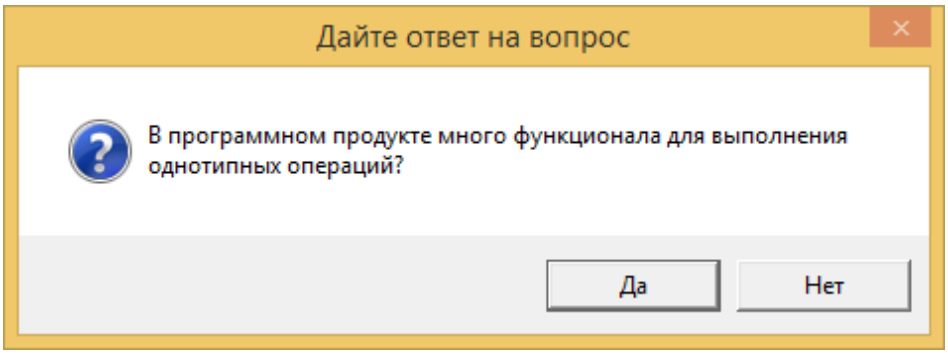

Рисунок 2 – Вариант интерфейса для аналитиков и/или программистов.

Утвердительные ответы на вопросы 1-8 говорят в пользу выбора автоматизированного тестирования, на вопросы 9-16 - в пользу ручного, на вопросы последней группы - в пользу смешанного.

Автоматизированное тестирование (Automated Testing)

1. В программном продукте много функционала для выполнения однотипных операций?
2. Регулярно выходят новые версии программного продукта?
3. Важно ли установить максимальное число пользователей, одновременно работающих с программным продуктом?
4. Нужно ли выявлять границы удовлетворительной производительности?
5. Критично ли проверить поведение программного продукта при стрессовых нагрузках?
6. Важна ли зависимость времени выполнения операций от их интенсивности выполнения?
7. Это многоплатформенное приложение?
8. Проводится smoke(дымовое) тестирование?

Ручное тестирование (Manual Testing)

9. Важно ли обнаружить максимальное количество «косметических» дефектов в программном продукте?
10. Проводится ли тестирование удобства использования?
11. Планируется ли использование программного продукта людьми с ограниченными возможностями?
12. Важно ли провести процесс тестирования с позиции потенциального пользователя?
13. Важна ли дешевизна тестирования в краткосрочной перспективе?
14. Нужно ли проверять нестандартные сценарии использования?
15. Проект небольшой по объему?
16. Будет ли проверяться формат и правильность печати документов?

Смешанное тестирование (Semi-Automated Testing)

17. Будет ли проводиться системное тестирование?
18. Основные проектные ресурсы выделены на функциональное тестирование?
19. В программной реализации много сложных структур данных, ветвлений, циклов и так далее?
20. В каждой последующей итерации тестирования нужно вводить новые наборы исходных данных вручную?
21. Важно ли проверить работу программного продукта с использование некорректных исходных данных?
22. Будет ли проводиться проверка соответствия нормативным требованиям?
23. Будет ли в процессе тестирования применяться моделирование?
24. Будет ли проводиться тестирование API (Application programming interface)?

Для каждого ответа аналитиков и/или программистов тестировщики расставляют веса (рис. 3).

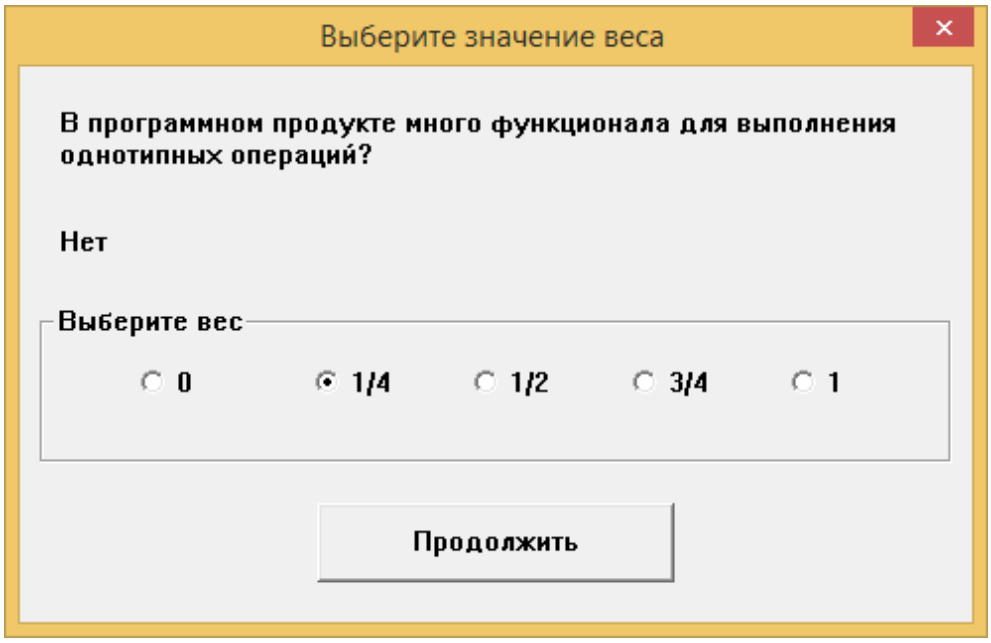

Рисунок 3 – Вариант интерфейса для тестировщиков.

На основе полученных весов решается многокритериальная задача [24]. Используется метод линейной свёртки. Подробно алгоритм построения линейной свёртки описан в работе [25]. В результате выполнения алгоритма выдаётся рекомендация об автоматизированном, ручном или смешанном подходе к тестированию программного продукта.

Список источников

1. Thoben, K.-D.,Wiesner, S., and Wuest, T."Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples

International Journal of Automation Technology, Vol. 11, No. 1, 2017. Pp. 4 – 16.

2. K. Shirase and K. Nakamoto, “Simulation Technologies for the Development

of an Autonomous and Intelligent Machine Tool,” Int. J. Automation Technol., Vol.7, No.1, pp. 6-15, 2013.

3. R. Baheti and H. Gill, “Cyber-physical systems,” The impact of control technology, pp. 161-166, 2011.

4. КуприяновскийВ. П., Намиот Д. Е., Синягов С. А. Кибер-физические системы как основа цифровой экономики// International Journal of Open Information Technologies. – 2016.- Т. 4. - № 2. – С. 18-25.

5. E. Geisberger and M. Broy, “agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems,” Springer, Berlin, Heidelberg, 2012.

6. E. Wallace and F. Riddick, “Panel on Enabling Smart Manufacturing,” State College, USA, 2013.

7. A. Fedorov, E. Goloschchapov, O. Ipatov, V. Potekhin, V. Shkodyrev, and S. Zobnin, “Aspects of Smart Manufacturing Via Agent-based Approach,” Procedia Engineering, Vol.100, pp. 1572- 1581, 2015.

8. S. Mittal, M. Kahn, J. Davis, and T. Wuest, “Smart Manufacturing: Characteristics and Technologies,” Columbia, SC, USA, 2016.

9. Покатаева Е. Программное обеспечение цифрового предприятия. Станкоинструмент. № 3 (012). 2018. С. 56-67.

10. Vassili van der Mersch Automated Testing for the Internet of Thingshttps://nordicapis.com/automated-testing-for-the-internet-of-things/ Retrieved: May, 2019

11. L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey, Computer Networks,” Vol.54, No.15, pp. 2787-2805, 2010.

12. L. Tan and N. Wang, “Future internet: The Internet of Things,” in: 2010 3rd Int. Conf. on Advanced Computer Theory and Engineering (ICACTE 2010), Chengdu, China, V5-376-V5-380.

13. D. Miorandi, S. Sicari, F. de Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” Ad Hoc Networks, Vol.10, No.7, pp. 1497–1516, 2012.

14. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems,” Vol.29, No.7, pp. 1645-1660, 2013.

15. Жигадло В. Э. «Особенности построения системы управления мегаполисом на примере Санкт-Петербурга». Региональная информатика (РИ-2016). Юбилейная XV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2016). Санкт-Петербург, 26-28 октября 2016 г.: Материалы конференции.\ СПОИСУ. – СПб. С. 34.

16. Parikshit Joshi. The 4 computing types for the Internet of things.https://www.iotforall.com/4-computing-types-for-iot/ Retrieved: May, 2019

17. Dhruv Shah. Implementing Fog Computing for IoT Ecosystems.https://dzone.com/articles/implementing-fog-computing-for-iot-ecosystem Retrieved: May, 2019

18. Как Edge Computing может трансформировать IT-инфраструктуру. https://www.cloud4y.ru/about/news/kak-edge-computing-mozhet-transformirovat-it-infrastrukturu/ Retrieved: May, 2019

19. Kyle Nordeen. Update your testing strategy for the IoT with Automation. https://www.getzephyr.com/insights/update-your-testing-strategy-iot-automation Retrieved: May, 2019

20. Галимова Е. Ю. Тестирование "черного ящика". Анализ методов // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1, Естественные и технические науки: периодический научный журнал. - 2012. - № 1. - С. 43-46.

21. DDos Attacks, https://www.imperva.com/learn/application-security/ddos-attacks/ Retrieved: May, 2019

22. Частикова В.А., Картамышев Д.А., Власов К.А. Нейросетевой метод защиты информации от DDoS-атак. Современные проблемы науки и образования. №1 (часть 1). 2015. https://science-education.ru/ru/article/view?id=18343 Retrieved: May, 2019

23. Галимова Е. Ю., Коваленко А. Н. «Метод выбора между ручным и автоматизированным тестированием, основанный на свойствах программного продукта». Журнал«ВестникДонскогогосударственноготехническогоуниверситета», №4, 2016. С. 134-140.

24. T’Kindt Vincent, Multicriteria Scheduling Problems // MultipleCriteria Optimization, Springer Science & Business Media, 2002, pp 445-491.

25. Галимова Е. Ю., Коваленко А. Н. статья «Выбор способа тестирования как решение многокритериальной задачи» [Электронный ресурс] // «Инженерный вестник Дона», 2016, №3, URL: http://ivdon.ru/ru/magazine/archive/n3y2016/3756 Retrieved: May, 2019