

Тестирование работоспособности промышленного компьютера

Автор показывает, зачем и как необходимо применять тестирование промышленных компьютеров для небольших систем управления до их внедрения на объекте заказчика. Для проведения работы нужно сформировать базу в виде открытой операционной системы Ubuntu, после установки которой выполняется установка рабочей программной среды OpenSCADA. Далее автор предлагает сосредоточиться на тестах, проводимых на этапе предварительного тестирования (pre-testing) работоспособности. Описана методология тестирования Agile. Выполнен обзор инструментов функционального и нагрузочного тестирования.

Анна СЕРГЕЕВА
annserge@rambler.ru

Введение

Представьте ситуацию: вы приобрели полный комплект оборудования промышленной системы управления, достали его из упаковки, смонтировали на объекте, закрутили все крепежные болты, подключили и зафиксировали соединительные провода, установили и настроили полный пакет нужного программного обеспечения, запустили в эксплуатацию и... что-то пошло не так. Ну, или поначалу все выглядело вполне работающим, а проблемы возникли чуть позже. Что ж, теперь придется все это демонтировать, искать неисправности, определять причины возникновения и устранять их. А потом, естественно, монтировать заново.

Пожалуй, каждый системный интегратор не понаслышке знает, какой может быть цена аварий и сбоев оборудования на промышленных объектах. Вот если бы была возможность удостовериться в работоспособности такого оборудования до монтажа и запуска в эксплуатацию...

Автор утверждает: мало того, что такая возможность есть, но и как опытный инженер службы обеспечения качества настаивает на необходимости проведения предварительного тестирования промышленных компьютеров.

Конечно, вы можете протестировать промышленные компьютеры у себя в офисе. Но дело в том, что сам по себе компьютер — это только часть системы. Есть еще периферия, линии связи и сеть. Причем, как это нередко бывает, часть периферии невозможно протестировать «на столе», поскольку, например, клапаны могут быть вмонтированы в трубопроводы «намертво». А вам все же хочется удостовериться в работоспособности такого оборудования, что при промышлен-

ных вариантах его исполнения имеет особую важность.

Для инженеров службы обеспечения качества и системных интеграторов автор предлагает следующий подход к проведению предварительного тестирования работоспособности промышленного оборудования. На первом этапе на компьютер устанавливается открытая, а следовательно — бесплатная и доступная операционная система (ОС). Затем выполняется настройка драйверов, если это необходимо, и проводится установка программной среды окружения. Далее осуществляется предварительное тестирование работоспособности компьютера.

Выбор ОС. Почему Ubuntu 12.04 LTS?

В качестве операционной системы для тестирования оборудования автор предлагает использовать Ubuntu [1]. Это операционная система, базирующаяся на Debian GNU/Linux.

Ее основные достоинства:

- Свободное, то есть абсолютно бесплатное программное обеспечение, его распространенность и доступность.
- Хорошая финансовая поддержка спонсора и основного разработчика — компании Canonical [2]. А это означает, что проект Ubuntu является долгосрочным и активно развивается.
- Широкое сообщество пользователей и разработчиков. Нужно заметить: такая постоянно растущая популярность доказывает, что система высоко зарекомендовала себя на рынке операционных систем.
- Надежность и гибкость плюс масштабируемость системы, что немаловажно для промышленных систем управления.

- Напомним также, что под стандартной ОС легко протестировать любую стандартную периферию, создать сетевые приложения для дистанционного тестирования и графические приложения для вывода информации о прохождении тестов для оператора [3].

Несколько слов о версиях этой операционной системы. Релизы Ubuntu, имеющие индекс LTS (Long Term Support, долгосрочная поддержка), имеют более длительную поддержку от компании Canonical, чем большая часть релизов Ubuntu. Заявлено следующее запланированное обновление пакетов: для пользовательских версий — на срок до трех лет, а для серверных — до пяти лет, с учетом оплачиваемой техподдержки от Canonical на протяжении этого периода. До момента выхода новой версии LTS также время от времени выходят пакеты обновлений для текущей версии LTS. Они имеют то же самое кодовое наименование, только различаются дополнительной цифрой, указываемой после номера версии.

На момент написания статьи (октябрь 2013 года) в качестве LTS-версии разработчик заявил релиз Ubuntu 12.04 LTS с кодовым именем Precise Pangolin («Педантичный ящер»). Обратим внимание на то что, начиная с текущей версии техническая поддержка для релизов LTS составит пять лет, как для персональных и планшетных компьютеров, так и для серверов.

Установка ОС

Итак, с операционной системой мы определились, самое время решать вопрос ее установки. Установить ОС Ubuntu можно различными способами.

Например, с имеющегося компакт-диска. Это тривиальная задача, с ней справится любой штатный системный администратор.

Однако не стоит забывать, что надежность хранения данных на компакт-дисках, из-за того, что они могут сломаться и поцарапаться, не идет ни в какое сравнение с флэш-картой.

Начнем с того, что далеко не каждое устройство снабжено оптическим приводом, а это значит, что вы не можете установить операционную систему с компакт-диска. В частности, это особенно важно, когда необходимо поставить систему на промышленный компьютер. Ничего страшного, ведь разъем-то для USB у такого устройства, как правило, имеется, и всегда можно воспользоваться установкой с флэш-накопителя. На наш взгляд, у такого подхода гораздо больше плюсов.

При создании флэш с загрузочным пакетом операционной системы Ubuntu для компьютеров, работающих на основе какой-либо версии Ubuntu, предназначена специальная встроенная утилита `usb-creator-gtk`. Она представлена на рис. 1.

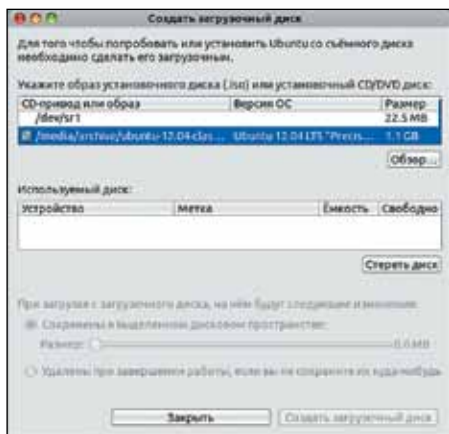


Рис. 1. Интерфейс Ubuntu-утилиты `usb-creator-gtk` для создания загрузочного диска

Для пользователей Windows существует утилита UNetbootin, которая быстро и удобно создаст необходимую загрузочную флэшку Ubuntu. Дистрибутив программы можно скачать на странице [4]. Интерфейс утилиты приведен на рис. 2.

Наконец, отметим, что такой подход к установке системы Linux позволяет зна-

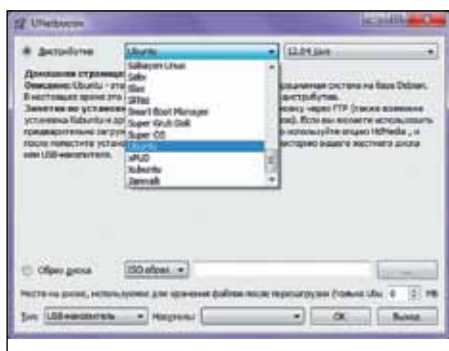


Рис. 2. Интерфейс Windows-утилиты UNetbootin для создания загрузочного диска

чительно сократить затраченное время. Так, выгода тем выше, чем на большее количество компьютеров система должна быть поставлена.

Установка и настройка программной среды

Итак, на данном этапе мы имеем компьютер с установленной ОС Ubuntu 12.04 LTS.

Также напомним, что, при необходимости, выполняется дополнительная настройка тех или иных драйверов, в зависимости от конкретной реализации оборудования.

Затем следует провести инсталляцию и настройку конкретной среды программных приложений, которые планируется задействовать для работы системы. Как правило, устанавливается SCADA-система (Supervisory Control And Data Acquisition), диспетчерское управление и сбор данных), такая, например, как свободно распространяемый проект OpenSCADA, предлагаемый на рынке программ для АСУ ТП (автоматизированных систем управления технологическими процессами).

Система OpenSCADA находит свое применение как на промышленных объектах — в роли полнофункциональной системы SCADA, так и во встраиваемых системах — в виде среды исполнения [5]. Пример проекта системы автоматизированного управления, реализованного в OpenSCADA, приведен на рис. 3.

Эта программная среда предназначена для сбора, архивирования, визуализации данных, выдачи сигналов управления и прочих операций, которые свойственны полнофункциональной системе SCADA. В ней соблюдены принципы многоплатформенности, модульности и масштабируемости.

Таким образом, результатом выполнения описанных выше действий является сформированная база, готовая для дальнейшего проведения предварительного тестирования

(pre-testing) работоспособности промышленного компьютера.

Здесь относятся, в частности, проверка срабатывания всех возможных датчиков, индикации нормальной работы и аварий, реакции на различные входные воздействия, а также ведение бортового журнала и формирование отчетов по событиям.

Но для того чтобы процесс тестирования не превратился в набор хаотичных манипуляций, необходимо применять системный подход. Поэтому приведем немного теоретических данных.

О методологиях тестирования

Как правило, вся команда, трудящаяся над созданием программно-аппаратных систем, включая менеджеров, проектировщиков, разработчиков и тестировщиков, ставит перед собой основную, самую важную задачу: разработать продукцию, обеспечив ей максимально высокое качество, по возможности сокращая сроки выпуска без снижения функциональности и затрат на исполнение.

И здесь чрезвычайно важно правильно подойти к вопросу выбора методологии разработки, поскольку проекты неизбежно приходится перерабатывать в связи с изменяющимися требованиями и незапланированными решениями. Вследствие этого результат может вовсе не оправдать ожидания: велико число дефектов, некоторые функциональные узлы могут быть отключены, сроки сдачи проектов отодвинуты на неопределенное время. В этот список, пожалуй, следует добавить и недовольство клиентов.

Существующие на сегодня итерационные методики (Agile, Extreme programming, Scrum и др.) значительно сокращают цикл «проектирование – разработка – тестирование – внедрение» за счет того, что более эффективно реагируют на модификацию требований, а также дают возможность разработчикам

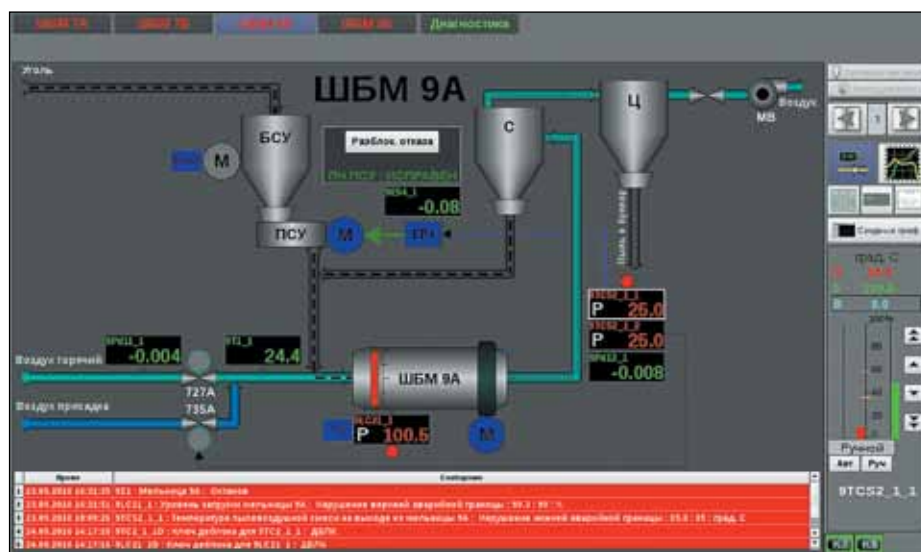


Рис. 3. Проект OpenSCADA — система автоматизированного управления шаровых мельниц

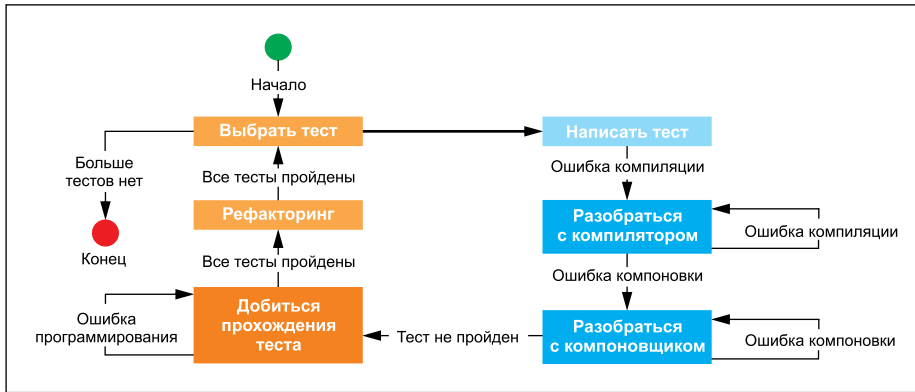


Рис. 4. Цикл разработки и выполнения тестов по Agile

и отладчикам более оперативно получать обратную связь по вопросам качества.

В данном классе методология Agile наиболее успешна [6]. Она предполагает предварительную запланированную разработку тестов, а процесс тестирования включает в себя большое число итераций с обязательным обеспечением непрерывной обратной связи.

Для проекта Agile характерно разбиение больших задач на более мелкие части, так, чтобы была возможность отладить каждую из них независимо от других, а также максимально обеспечить их независимость от других задач (например, задача обслуживания LCD-индикаторов, задача контроля GPRS, задача опроса АЦП и т. д.).

При этом участники проекта наблюдают общую картину, но в своих планах они могут сосредоточить внимание на конкретных небольших сюжетах. При таком подходе обеспечивается большая управляемость, а также значительно увеличиваются шансы на выполнение требований по функциональности и соблюдением запланированных сроков и надлежащим качеством.

На рис. 4 показан цикл разработки и выполнения тестов по методологии Agile.

Такой методологии также свойственно следование корректировке планов, а также частое и раннее тестирование, а это значительно упрощает внесение изменений, которые неизбежно возникают в ходе существования всего проекта. По возможности тесты автоматизируются и повторяются после каждой модификации внедряемого объекта, что помогает избежать нежелательных послед-

ствий. Отметим, что фаза раннего тестирования, предполагаемая в Agile, соответствует концепции предварительного тестирования, предлагаемая автором в этой статье.

Особенности встраиваемых систем

Если для разработчиков встраиваемого программного обеспечения такой подход весьма интересен, то в индустрии встраиваемых аппаратно-программных систем все обстоит немного сложнее.

Для стандартного предприятия IT-разработки программисты и тестировщики спокойно рассчитывают, что продукция, ими создаваемая, будет развернута на типовой, стандартной стабильной аппаратуре и вступить во взаимодействие с отлаженными стандартными программными компонентами.

А вот во встраиваемых приложениях все с точностью до наоборот. Программное обеспечение практически всех уровней либо заново разрабатывается, либо подвергается модификации. (Впрочем, все это может происходить и параллельно.)

Еще один немаловажный момент. Тестирование встраиваемого программного обеспечения в обязательном порядке должно выполняться на реальном целевом оборудовании, а последнее, в свою очередь, может быть «сырым» или ограниченно доступным.

Вот и получается, что, даже если отдельные единичные тесты легко выполняются обособленно, все равно считать их завершенными не получится, пока они не пройдут тесты

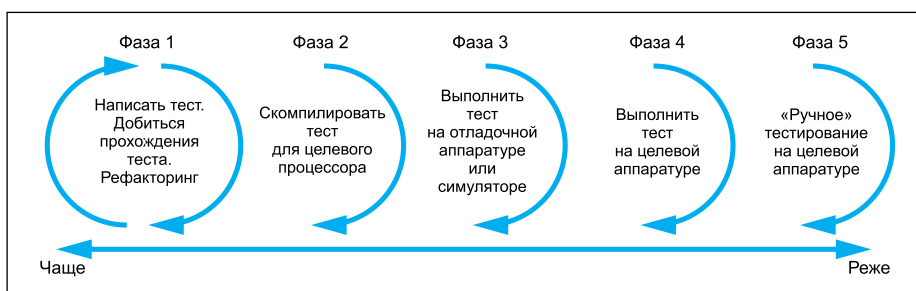


Рис. 5. Повторное выполнение регрессивных тестов в разных средах

не «на столе», а в «боевых» условиях, то есть на реальной аппаратуре и в окружении реального целевого программного окружения.

На рис. 5 предлагается схема пофазового проведения тестов в разных средах.

Преимущества Agile

С точки зрения команды участников проекта Agile имеет следующие преимущества:

- Концентрация на каждом отдельно взятом функциональном узле аппаратуры. Определенные, какие сценарии отказов необходимо проверять.
- Профилактика дефектов, поскольку функциональные узлы проверяются по мере их подключения к системе; автоматизация тестов позволяет мгновенно обнаруживать и устранять проблемы сразу же после их исправления.
- Снижение возможности дорогостоящих отказов, возникающих на завершающей стадии проекта, уже после внедрения и полного запуска системы в рабочем режиме.

С позиции руководства проекта Agile предоставляет:

- предсказуемость сроков выпуска проекта;
- повышение качества поставляемой продукции;
- наибольшее совпадение с требованиями заказчиков при минимальном числе попыток;
- ускорение разработки на фоне снижения рисков и затрат.

Таким образом, раннее тестирование, предусмотренное в рамках методологии Agile, дает возможность устранить часть проблем заранее. Это сокращает объемы более трудоемкого тестирования и отладки на реальной аппаратуре уже в процессе ее работы, когда для устранения неполадок придется приостанавливать производство.

О средствах тестирования

Совсем недавно для тестирования программных продуктов не существовало никаких специальных инструментов, все проверки выполнялись вручную либо силами программистов, либо пользователями. Ни о каком системном подходе в обеспечении качества не могло быть и речи.

Затем стали появляться первые средства автоматизации тестирования, поскольку практический опыт показал низкую эффективность ручного тестирования, так как оно требует много времени и трудозатрат. Такие первые инструменты автоматизации, по сути, являлись библиотеками, которые можно было применять для написания тестов, а это, в свою очередь, требовало от тестировщика уровня программирования, сопоставимого с уровнем разработчика. Но прогресс не стоит на месте, и современные инструменты автоматизации тестирования требуют от человека минимума действий для написания тестов.

Условно инструменты автоматизации тестирования можно классифицировать на средства функционального тестирования и средства нагрузочного тестирования. Первые предназначены для проверки соответствия программного обеспечения предъявляемым к нему требованиям. Вторые служат для оценки производительности программных продуктов.

Рынок инструментов функционального тестирования на сегодня предоставляет множество решений. К лидирующим компаниям можно отнести:

- HP (WinRunner, QuickTest Professional);
- Borland (SilkTest);
- IBM (Functional Tester, Robot);
- AutomatedQA (TestComplete).

Следует отметить, что часть из этих средств, представляющих собой среды для разработки приложений, используют «промышленные» языки программирования. (Так, например, для разработки скриптов VB использован язык QTP, а Functional Tester исполнен в среде Eclipse, что дает возможность написания Java-скриптов.) Другие же применяют адаптированные или собственные, специально разработанные языки. (В частности, TestComplete использует язык 4Test, а Robot — язык SQABasic.) Многие из этих инструментов предназначены либо для тестирования веб-приложений либо консольных, запрограммированных в .Net или Java [7].

Что касается нагрузочного тестирования, то здесь механизм функционирования инструментов более сложный. Выполняется перехват трафика между сервером и тестируемой программой, затем эти данные представляются в виде, удобном для работы и анализа.

Поскольку в силу сложности и многофункциональности современного программного обеспечения выполнения лишь одного функционального тестирования явно недостаточно, практически все производители средств автоматизации тестирования выпускают инструменты автоматизации нагрузочного тестирования. Так, например, HP предлагает LoadRunner, а IBM — Performance Tester и Robot.

Также существуют и широко востребованы среды поддержки тестирования. Они служат для хранения тестовых сценариев, учета требований, дают возможность управлять процессом выполнения тестов, вести учет обнаруженных дефектов, выполнять анализ проведенных работ.

Безусловным лидером здесь является продукт HP Quality Center, который интегрируется со средствами функционального (HP QuickTest Professional) и нагрузочного (LoadRunner) тестирования. Он же обеспечивает полное управление процессом тестирования.

При выборе конкретных инструментов тестирования одним из ключевых моментов является возможность интеграции этих инструментов с программным обеспечением, используемым в данной компании. Например, если для разработки программного обеспечения и автоматизации в компании уже используются продукты IBM, то в качестве инструмента тестирования выбирать TestComplete, возможно, не так уж и рационально.

Для инструментов функционального тестирования важным фактором является поддержка той или иной среды разработки, автоматическая регистрация обнаруженных дефектов, наличие сценариев восстановления, возможность предоставления отчетности о тестировании. Для нагрузочного тестирования необходимы поддержка протоколов, которые используются тестируемой программой, возможность гибкой настройки тестовых сценариев, наличие встроенных средств мониторинга серверных аспектов, наличие средств анализа результатов и предоставления отчетности.

Следует учитывать также стоимость инструментов тестирования. Если запланировано однократное тестирование, то покупка дорогостоящих инструментов нецелесообразна. Альтернативным решением будет аренда лицензий (покупка временных лицензий). Это, как правило, обойдется гораздо дешевле.

Также нужно обращать внимание и на фактор исполнителей, то есть стиль работы и опыт конкретных инженеров службы обеспечения качества.

Заключение

Итак, основные вехи намечены, материал для размышлений предоставлен. Подведем итоги изложенного.

В результате установки и настройки на исследуемом компьютере свободно распространяемой и вместе с тем надежной ОС Ubuntu тестировщик получает базу для проведения дальнейших работ.

Выполняется инсталляция программной среды окружения. В зависимости от целей использования компьютера инсталлируется та или иная SCADA-система. Рынок программных продуктов для промышленных применений предлагает множество подходящих решений, как коммерческих, так и свободно распространяемых. Это же можно сказать и об имеющемся многообразии доступных средств тестирования, которые будут выбраны и задействованы службой обеспечения качества.

Предлагаемая методология тестирования, Agile, позволяет значительно повысить качество выпускаемой продукции, сократить сроки реализации проекта и в целом поднять рейтинг продукта на рынке, избилующем многообразием предложений.

Не стоит, однако, вдаваться в крайности и забывать о том, что покрыть всю программу тестовыми модулями практически невозможно и экономически нецелесообразно. Мастерство тест-инженеров как раз и заключается в выборе оптимального курса тестирования, составлении грамотных тест-планов, следовании этим планам, дроблении сложных задач, сосредоточении на критических и первоочередных моментах, распараллеливании выполнения задач, а также налаженной постоянной обратной связи с командой разработчиков.

Положительные результаты, обеспеченные на этапе предварительного тестирования, дают возможность уверенно заявить о том, что проверенное оборудование, возвращенное на объектах, гарантирует большую надежность работы. При этом оптимизация затрат времени и сил полностью окупается при эксплуатации на объекте, поскольку существенно снижается процент отказов и аварий, что весьма важно в промышленной сфере. ■

Литература

1. www.ubuntu.com
2. www.canonical.com
3. www.ubuntuforums.org
4. www.unetbootin.sourceforge.net
5. www.oscada.org
6. Хендерсон П., Греннинг Дж. Применение итерационных методик для тестирования встраиваемого ПО. Современные технологии автоматизации // СТА-Пресс. 2012. № 4.
7. Можаяв П. Средства автоматизированного тестирования // Открытые системы. 2009. № 3.