

Использование методологии и инструментов CI/CD

при разработке, тестировании и сопровождении проектов на базе платформы SIMATIC WinCC Open Architecture

SIEMENS

Рассматриваются вопросы использования методологии и инструментов непрерывной интеграции и доставки (CI/CD) при разработке, тестировании и сопровождении прикладных проектов на базе SCADA-системы SIMATIC WinCC Open Architecture (WinCC OA). Приводимые положения иллюстрируются на примере разработки проекта системы мониторинга состояния устройств релейной защиты и автоматики серии SIPROTEC 5.

ООО «Сименс», г. Москва

Значительное увеличение объемов обрабатываемой информации, усложнение архитектуры и решаемых задач, повышение требований к качеству и надежности являются характерными тенденциями развития промышленных систем сбора, обработки данных и управления [1]. Помимо роста объемов информации и трансформации структурно-функционального облика меняется динамика внедрения и в принципе жизненный цикл таких систем — ужесточаются требования к срокам реализации проектов, усиливается необходимость в непрерывном развитии внедренных систем, их адаптации к меняющимся условиям и требованиям в процессе эксплуатации. Указанные обстоятельства предопределяют потребность в использовании апробированных практик и инструментов «гибкой» разработки ПО при создании и последующем развитии промышленных информационных систем.

В настоящей статье описан опыт департамента «Корпоративные технологии» ООО «Сименс» в организации процесса разработки программных приложений на базе SCADA-платформы SIMATIC WinCC Open Architecture (WinCC OA) с использованием методологии и инструментов

непрерывной интеграции и доставки (CI/CD).

Методология и инструменты для построения Agile-процесса в мире SCADA-систем

Платформа WinCC OA предоставляет множество готовых компонентов, драйверов и библиотек функций, а также включает ряд типичных для SCADA-систем мастеров для конфигурирования и параметрирования системы без написания пользовательского программного кода [2]. Это, впрочем, не отменяет необходимости в программной разработке для реализации прикладной логики, алгоритмов анализа данных, пользовательского интерфейса — особенно для систем, к которым предъявляются специализированные требования по функциональности, визуализации, безопасности и т.д. Практика разработки коммерческого ПО предоставляет проверенные подходы и инструменты для удобной организации процесса и повышения качества конечного продукта. Такие подходы и инструменты востребованы и при создании проектов на базе WinCC OA. Ряд привычных для разработчиков ПО инструментов уже штатно имеется в составе WinCC OA (средства для мо-

дульного тестирования, отладчик, инструментарий для версионирования в среде разработки GED1 и др.), при этом для построения сквозного Agile-процесса представляется целесообразным использование апробированных методологий и инструментов.

Для определения специфики организации Agile-процесса для WinCC OA-проектов рассмотрим общий подход к гибкой разработке ПО.

Следует выделить два аспекта проектов создания ПО: организационный и технический. Организационный связан с планированием и постановкой задач и коммуникацией в команде. Технический заключается в стандартизованных практиках программной инженерии: от инженерии требований до сопровождения в ходе эксплуатации.

Организационные аспекты

В работе над крупными проектами программной разработки участвует несколько специалистов с разными интересами в отношении организации работы с ПО. Перечислим основные роли:

- заказчик — владелец целевой системы — заинтересован в стабильной работе и обновлении функциональности без перебоев в работе всей системы;

- заказчик-оператор заинтересован в простом и управляемом процессе обновления версий;
- разработчик — управляющий проектом — заинтересован в удобном управлении задачами и в инструментах отслеживания статуса разработки;
- разработчик — ведущий программист — заинтересован в инструментах сборки всего проекта, в быстром и автоматизированном развертывании системы у заказчика;
- QA-инженер заинтересован в инструментах отслеживания статуса тестирования приложения;

- разработчики программных компонентов заинтересованы в работе актуальной версии кода и автоматизации рутинных процедур.

С организационной точки зрения для управления проектом требуется единая платформа. В департаменте «Корпоративные технологии» ООО «Сименс» используются продукты компании Atlassian, поставляющей приложение Jira для управления задачами (рис. 1) и приложение Confluence для документирования. Эти инструменты гибко настраиваются, интегрированы между собой; к ним

доступно множество дополнительных модулей (плагинов): построение диаграмм, почасовое планирование и отслеживание и др. Кроме того, используются open source инструменты разработки для разных языков программирования и платформ (рис. 2).

Технические аспекты

С программной точки зрения приложение на WinCC OA представляет собой набор конфигурационных файлов, файлов с описанием панелей, файлов с описанием экранных форм

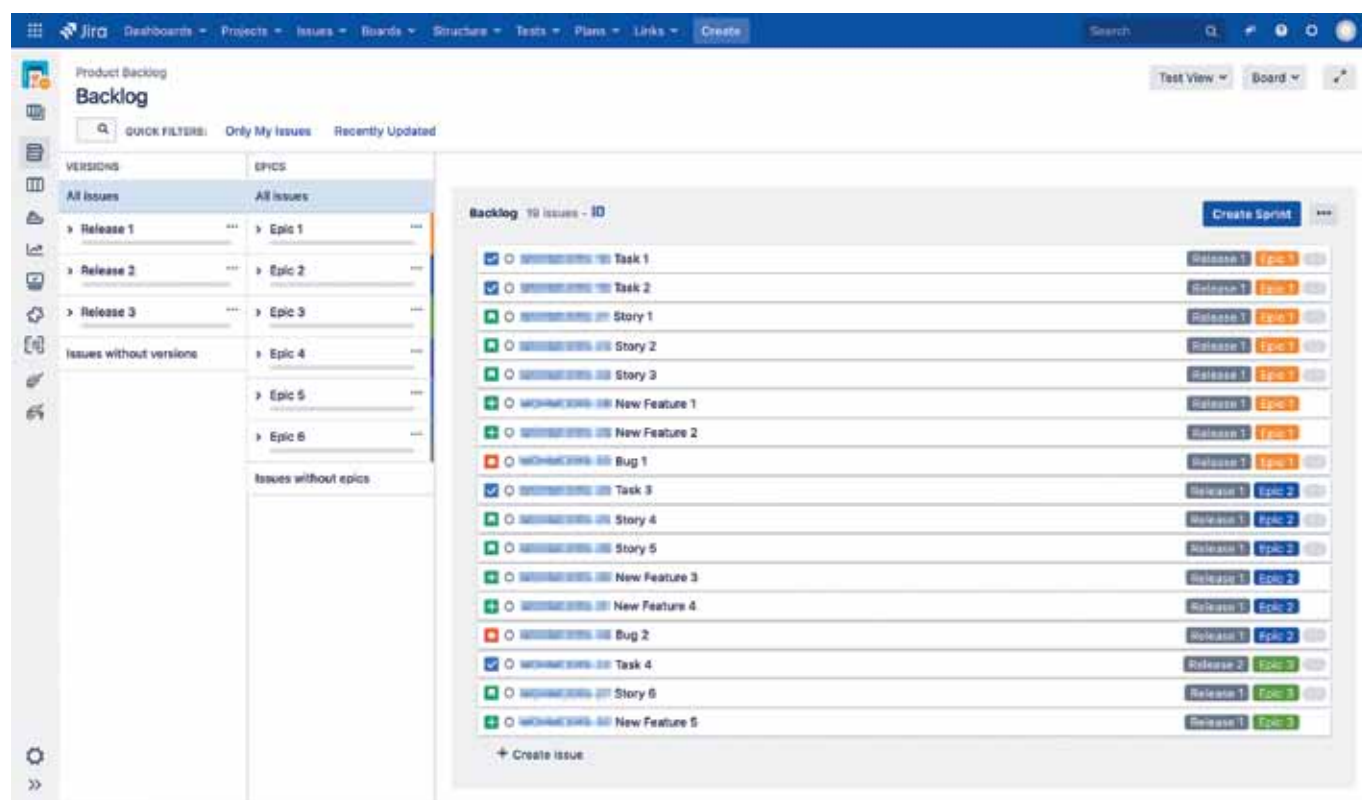


Рис. 1. Интерфейс Jira (источник: atlassian.com)

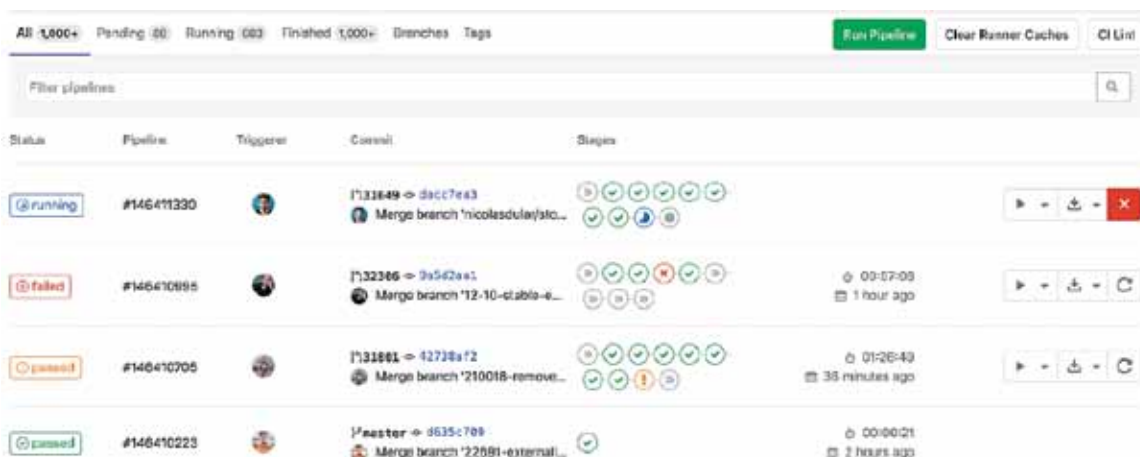


Рис. 2. Интерфейс CI-сервера (источник: code.siemens.com)

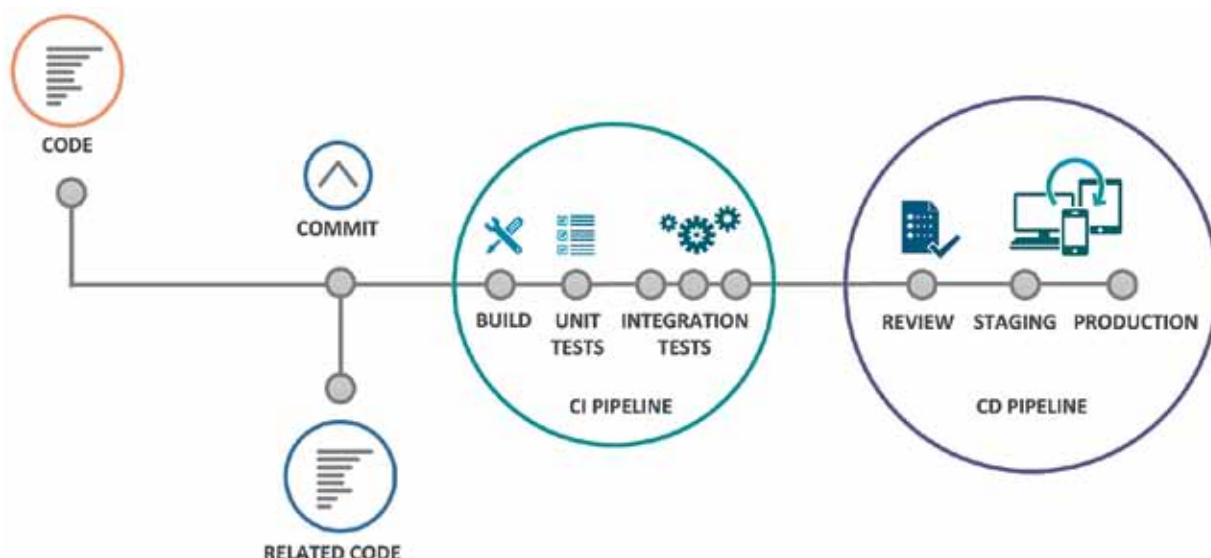


Рис. 3. Логика построения конвейера CI/CD (источник: Siemens)

и прочих объектов, файлов библиотек и сценариев, а также файлов конфигурационных и исторических баз данных. Необходимость в особых подходах к разработке, тестированию и сопровождению ПО определяется следующими требованиями:

- ▶ одновременный труд нескольких разработчиков требует синхронизации результатов (ежедневно);
- ▶ ведущий разработчик хочет иметь возможность отслеживать вносимые в код изменения;
- ▶ нужна история версий кода;
- ▶ требуется частое обновление версий ПО;
- ▶ нужно отслеживать покрытие кода тестами (юнит-тестами, интеграционными тестами);

▶ разработка часто ведется на локальных (или виртуальных) машинах, и требуется автоматизация загрузки проекта в тестовое окружение, а потом и в продуктивное окружение с реальным оборудованием;

▶ нужна возможность вернуться к предыдущей стабильной версии приложения в случае необходимости.

Конвейер CI/CD

Подход, отвечающий перечисленным выше требованиям к разработке, тестированию и сопровождению коммерческого ПО, известен как Continuous Integration & Continuous Delivery (непрерывная интеграция и непрерывная доставка, сокращенно — CI/CD).

Задача CI/CD заключается в организации автоматизированного процесса сборки, доставки и тестирования по типу конвейера (pipeline) в целях исключения влияния человеческого фактора и дестабилизации процесса разработки.

Логика построения конвейера CI/CD для WinCC OA состоит в следующем (рис. 3):

- ▶ разработчик пишет код, например процедуру обработки события по изменению значений точки данных от callback-функции dpConnect;
- ▶ написанный фрагмент кода передается (COMMIT → PUSH) в общую кодовую базу;
- ▶ приложение собирается (BUILD) — в случае с WinCC OA это может быть

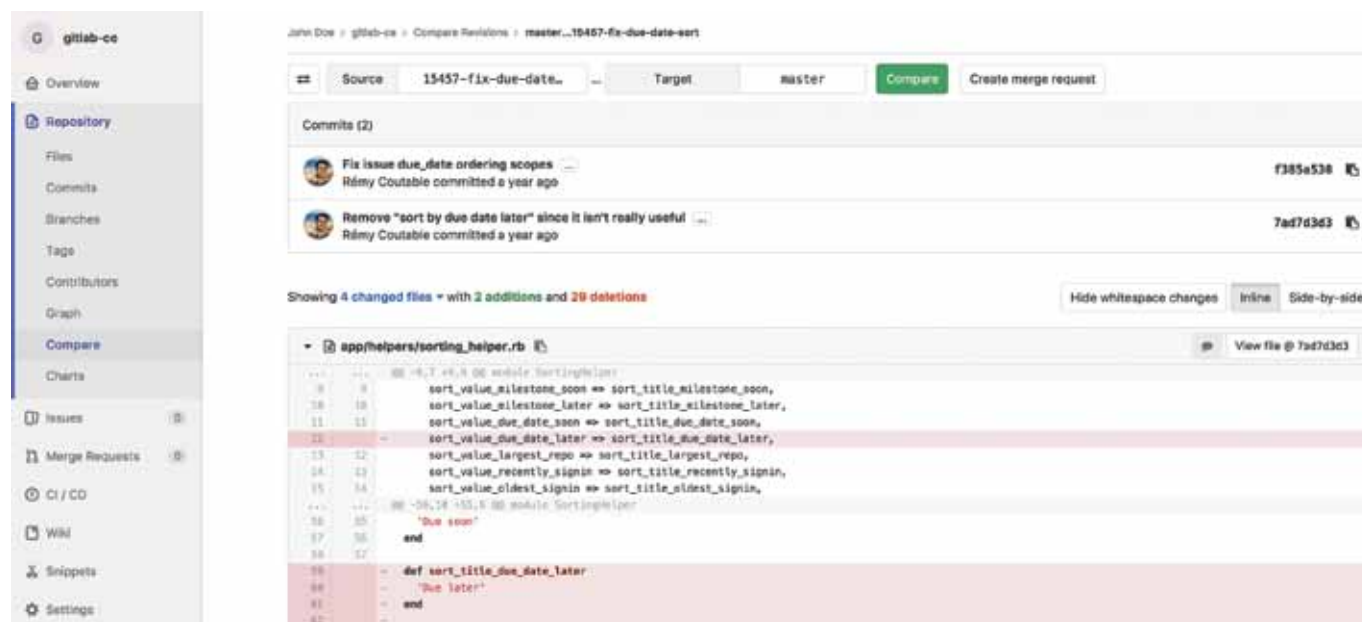


Рис. 4. Интерфейс code.siemens.com

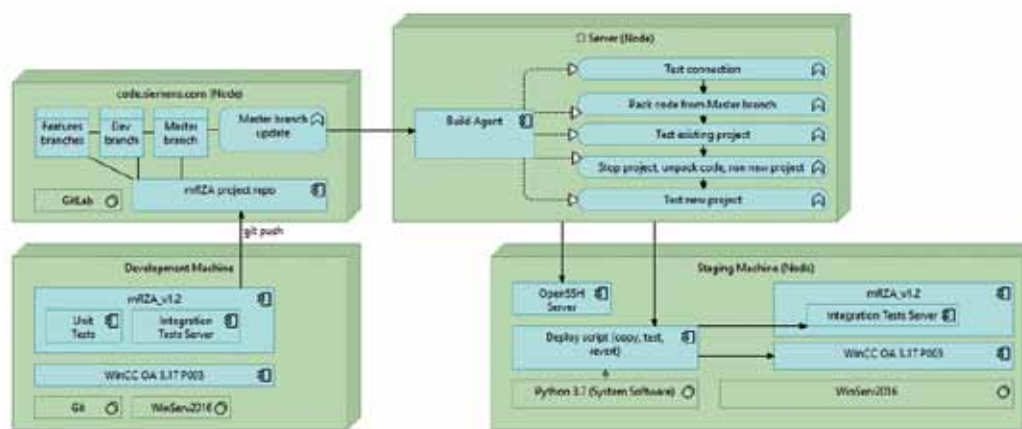


Рис. 5. Логика построения конвейера CI/CD в реальном проекте разработки (источник: Siemens)

сборка библиотек или C#-приложений);

- ▶ проводится тестирование отдельных функций/модулей (UNIT TESTS);

- ▶ проводится тестирование компонентов и их взаимодействия (INTEGRATION TESTS);

- ▶ результаты тестирования и переданный код проходят ревизию (REVIEW);

- ▶ далее приложение развертывается на стенде (STAGING) для оценки качества;

- ▶ в случае прохождения всех проверок приложение развертывается в реальном окружении (PRODUCTION).

Такой стандартный конвейер доставки ПО применим как при разработке, так и при сопровождении ПО. Для каждого этапа этого конвейера есть проверенные инструменты, и они могут быть применены в проектах на базе WinCC OA.

Пример использования конвейера CI/CD при реализации проекта WinCC OA

Конвейер CI/CD был организован для проекта разработки системы мониторинга состояния устройств релейной защиты и автоматики (РЗА) серии SIPROTEC 5.

Особенности целевой системы:

- ▶ получение данных как по протоколу IEC 61850, так и по проприетарному протоколу устройств SIPROTEC 5 с динамическим созданием модели данных устройства;

- ▶ использование актуальной версии WinCC OA (3.17) с поддержкой InfluxDB в качестве штатной СУБД [3], что открывает возможности дальнейшей локализации ПО;

- ▶ применение имитационного моделирования для анализа аварийных событий в энергосистеме.

В разработке приложения участвует от 3 до 6 разработчиков одновременно, сборка приложения для тестирования осуществляется на отдельном сервере с последующим переносом стабильной версии на лабораторный стенд с подключенными устройствами РЗА. Заказчику приложение передается в составе программно-аппаратного комплекса, включающего сервера и систему хранения данных, прошедшего заводские испытания. На энергообъекте система проходит наладку и полноценные испытания, требующие внесения обновлений в ПО.

Для разработчиков создается пул виртуальных машин с Windows Server, на которых разворачивается WinCC OA и проект приложения, а также клиент Git для связи с корпоративным репозиторием программного кода code.siemens.com. Репозиторий построен на базе Gitlab и предоставляет удобный интерфейс для контроля версий и управления процессом разработки, включая wiki, трекер запросов на слияние кода (мерж-реквесты) и конфигурацию конвейеров сборки (рис. 4).

Логика построения конвейера CI/CD в реальном проекте разработки показана на рис. 5.

Обновление ветви master, права на которое имеет только ведущий программист, запускает процесс сборки и доставки проекта, построенный на CI-сервере.

Процесс проходит в несколько этапов:

- ▶ проверка SSH-соединения с машиной на стенде (STAGING);

- ▶ упаковка последней версии приложения в архив;

- ▶ автоматическое тестирование текущей версии приложения, позволяющее убедиться в работоспособности окружения;

- ▶ остановка текущего приложения, распаковка новой версии, синхронизация баз данных, запуск новой версии приложения;

- ▶ автоматическое тестирование новой версии приложения. В случае отрицательных результатов тестирования новой версии автоматически производится откат на предыдущую, работоспособную, версию.

Опыт построения конвейера CI/CD в реальном проекте разработки на WinCC OA

Опыт выявил следующие особенности при построении конвейера CI/CD для WinCC OA:

- ▶ имеются особенности запуска экземпляра:

- перед развертыванием нужно дождаться остановки проекта, можно использовать таймауты, проверку статуса остановки и т. д.;

- целесообразно использовать контейнеры с развертыванием каждый раз нового образа;

- ▶ желательно разделять код проекта, БД проекта и архив сигналов, в противном случае изменения в код вносятся вместе с изменениями в проект, и данные изменения в проект и БД трудно отслеживать;

- ▶ необходимо продумывать параметризацию конвейера CI/CD, чтобы была возможность собрать проект из произвольных веток и установить на произвольную машину;

► отдельно нужно продумывать модель патчей (на действующем объекте безопаснее обновить скрипт, чем проект целиком — не нужно будет повторно проводить весь объем тестов);

► нужна обратная идентификация истории изменений (коммита) из кода проекта, так как по экземпляру не сразу понятно, из чего он собран и что менялось в дальнейшем;

► имеются сложности с контролем целостности файлов проекта (определение того, каким образом сделана сборка — было ли ручное вмешательство в процесс);

► нужно использовать одну экосистему, например GitLab / Gitlab Runner в качестве CI-сервера вместо «зоопарка» средств.

Пути развития

Одним из возможных улучшений процесса сборки и доставки проектов WinCC OA является использование изолированных сред для запуска приложений — Docker-контейнеров, которые являются реализацией паттерна Immutable Infrastructure.

Такой подход имеет очевидные плюсы, в частности:

► повторяемое окружение (всё, что нужно для запуска определенной версии вашего проекта — операционная система, код, конфигурационные файлы, будет содержаться внутри Docker-образа, описанного определенным файлом. Среды, созданные

из одного образа, полностью идентичны);

► удобный запуск (готовый Docker-образ можно легко запустить на любом Docker-хосте);

► инфраструктура как код (не нужно вручную настраивать среду, устанавливать пакеты и зависимости — для этого есть Docker-файл, а также CI/CD-скрипты, которые декларативно описывают состояние системы, которого необходимо достичь);

► простое масштабирование (для того чтобы развернуть новый экземпляр проекта, необходимо всего лишь запустить новые Docker-контейнеры с помощью нескольких команд).

Процесс CI/CD с использованием Docker-контейнеров будет включать в себя создание из Docker-файлов образов проекта и сохранение их в реестре образов, обновление образов проекта, развертывание определенных версий приложения на Docker-хостах (серверах, на которых установлен Docker) через создание Docker-контейнеров из образов.

Наряду с преимуществами надо упомянуть и некоторые сложности этого подхода, связанные с аспектами хранения данных в WinCC OA. В частности, Docker-контейнер должен использоваться для запуска приложения (или проекта в нашем случае) в изолированной среде. Внутри контейнера не должно быть базы данных или самих данных — они должны находиться вне контейнера (либо

быть развернуты в другом контейнере, либо вне его).

Поддержка Docker-контейнеров запланирована в WinCC OA версии 3.18, выпуск которой ожидается весной 2021 года.

Выводы

Применение подхода CI/CD целесообразно при реализации проектов WinCC OA и позволяет автоматизировать процессы сборки, развертывания и тестирования на этапах разработки и сопровождения, сократить сроки реализации, свести к минимуму количество ошибок. На рынке имеются инструменты, позволяющие организовать конвейер CI/CD для проектов WinCC OA. Опыт практического применения подтверждает, что организация процесса непрерывной интеграции и доставки приложений WinCC OA обеспечивает повышение скорости разработки и качества конечного продукта.

Литература

1. Соловьёв С. Ю., Серов А. Ю., Кондрашкин А. А. Стандартизация при инжиниринге, сопровождении и модернизации крупномасштабных SCADA-проектов с SIMATIC WinCC Open Architecture // ИСУП. 2020. № 5.

2. Соловьёв С. Ю. Инжиниринг проектов на базе SCADA-системы SIMATIC WinCC OA // ИСУП. 2016. № 4.

3. Серов А. Ю., Соловьёв С. Ю. SIMATIC WinCC Open Architecture 3.17: не просто новая версия // ИСУП. 2020. № 4.

Ю. В. Бубнов, эксперт-исследователь,
Н. А. Рыжиков, инженер-исследователь,
департамент «Корпоративные технологии»,
С. Ю. Соловьёв, к. т. н., руководитель
Центра компетенций,
управление «Цифровое производство»,
ООО «Сименс», г. Москва,
тел.: +7 (495) 737-1737,
e-mail: icc.ru@siemens.com,
сайт: www.siemens.ru

