**# Переводим рутину ручного тестирования 1C на рельсы Jenkins-а и ADD. Легко!**

Вы все еще тестируете свои конфигурации 1С вручную? Да вы просто тратите жизнь впустую! Регулярные читатели инфостарта должны уже слышать ( и не по наслышке знать) про Vanessa Behavior и его новых отпрысков – ADD и Vannessa Automation.

Оба фреймворка – это замечательное воплощение идей удобного тестирования функциональности на 1С. Мы составляем cценарные тесты(или «фичи») на специальном языке gherkin, описывающем поведение пользователя в интерфейсе 1С Предприятия, а затем вручную прогоняем тесты на запускалке – внешней обработке 1С и узнаем, что у нас работает, а что – не очень. Если вы еще не пробовали рай BDD тестирования, то данный туториал будет максимально полезен: мы сразу убъем двух зайцев – на практическом примере узнаем, что это такое и научимся его правильно готовить.

Под правильной готовкой мы будем понимать не запуск тестов вручную (желающим в руки достаточно «плотный» материала про Ванессу), а создание переиспользуемого пайплайна тестирования в Jenkins. Пайплайн, который будет сам автоматически по расписанию запускать тесты. Пайплайн, который не будет ломать вам рабочие базы. Пайплайн, который даст удобный allure отчет. Наконец, пайплайн, который принесет уверенность в завтрашнем дне!

Звучит хорошо, не правда ли? Но сбавим градус пафоса, господа… и перейдем от теории сразу к практике. Все действия будут выполняться под Windows.

На картинках конечный результат будет выглядеть вот так:

```СКРИНШОТЫ```

**## 1. Установка GIT**

GIT - наверное уже известная всем система контроля верси кода, которая все больше вхожит в жизнь 1С-ков. Она нам потребуется для того, чтобы дженкинс могу работать со скриптами нашего пайплайна, которые расположены в экспериментальном репозитории https://github.com/ripreal/erp\_features.git (данные репозиторий подойдет, чтобы выполнить туториал, но для разворачивания на продакшене рекомендуется завести свой)

1. Скачиваем последний дистрибутив GIT for Windows и устанавливаем

**## 2. Установка и настройка Jenkins-а.**

Jenkins – бесплатная среда для автоматического запуска всех скриптов нашего пайплайна по расписанию. Установка и первичная настройка дженкинса не принесет никаких проблем.

1. Скачиваем дистрибутив JRE 1.8 и устанавливаем

2. Скачиваем последний дистрибутив Jenkins (на момент статьи это 2.141) и устанавливаем. Все настройки оставляем по-умолчанию.

3. Меняем стандартную кодировку дженкинса на UTF-8. Это важный этап, чтобы в веб-интерфейсе дженкинса все русские символы отображались корректно. Для этого добавляем параметр -Dfile.encoding=UTF8 в тег <arguments> в файле Jenkins.xml, расположенном в корневом каталоге установки дженкинса. Итоговая строка должна выглядеть примерно так:

```<arguments>-Xrs -Xmx256m -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle -Dfile.encoding=UTF8 -jar "%BASE%\jenkins.war" --httpPort=8991 --webroot="%BASE%\war"</arguments>```

4. Перезапускаем службу Jenkins в диспетчере задач Windows и проверяем, что все прошло хорошо, открывыв веб-интерфейс дженкинса по адресу http://localhost:8991.

![Начальный экран](media/Jenkins\_main.png)

**## 3. Настройка shared-libraries**

Эта удобная функция позволяет писать и складывать скрипты в отдельные библиотеки для их переиспользования в дальнейшем. Мы будем использовать эти библиотеки постоянно.

1. В веб-интерфейсе дженкинса переходим в меню Manage Jenkins => Global Tool Configuration => Global Pipeline Libraries

2.Нажимем Add и заполняем поля:

\* Name: shared-libraries

\* Default version: master

\* Retrieval method: modern SCM

\* Source Code Management: Git (не GitHub)

\* Poject Repository: https://github.com/ripreal/erp\_features.git

**## 4. Настройка Allure**

Аллюр позволит генерировать красивые отчеты прямо в дженкинсе по результатам тестирования в ADD.

1. Устаналиваем плагин allure. В веб-интерфейсе дженкинса переходим в меню Manage Jenkins => Manage plugins => Available, ищем в списке Allure и устанавливаем его

2. Устанавливаем сам дистрибутив allure. В веб-интерфейсе переходим Manage Jenkins => Global Tool Configuration => Allure Commandline installations => Add Allure Commandline. Заполняем появившиеся поля следующим образом

\* Name: allure

\* Label: allure

\* Download URL for binary archive: https://dl.bintray.com/qameta/maven/io/qameta/allure/allure-commandline/2.11.0/allure-commandline-2.11.0.zip

\* Subdirectory of extracted archive: allure

**## 5. Настройка окружения ADD**

Переходим к установке непосредственно самих утилит, нужных для работы вспомогательных административных скриптов и самого инструмента ADD.

1. Скачиваем последний дистрибутив OneScript и устанавливаем.

2. Устанавливаем библиотеку vannessa-runner для OneScript. Для этого выполняем команду:

```opm install vanessa-runner```

3. Также у вас уже должны быть установлены sclcmd (поставляется вместе с MS SQL Server) и powershell (с включенной политикой беспрепятственного запуска скриптов)

**## 6. Создание и настройка пайплайна в Jenkins**

Теперь самое интересное - создать и настроить пайплайн, который и будет запускать по расписанию весь процесс сборки, начиная с копирования эталонных баз и заканчивая их тестированием и очисткой.

1. В Веб-интерфейсе дженкинса переходим в меню New Item, заполняем проозвольное имя в поле Enter an item name (я выбрал erp\_features), выбираем тип скрипта - pipeline и нажимаем ОК

2. В новом окне открывается конфиг пайплайна. Единственная группа параметров, которую нам нужно в нем сейчас заполнить - это pipeline, в котором мы параметры репозитория, откуда скачивать код нашего пайплайна, включая Jenkinsfile - главный скрипт, входная точка всей сборки. Для этого скроллим в низ конфига и в группе pipeline заполняем следующие поля:

\* Definition -

Для написания самого скрипта мы будем испл В связке с несложным языком groovy