

Реализация инструмента для непрерывной интеграции для IBM z/OS с поддержкой средства управления конфигурациями IBM SCLM

Щербаков Александр Владимирович
544 группа

Научный руководитель: д.ф.-м.н., проф. Нестеров В.М.

Рецензент: спец., рук.гр. Хлебников П.Н.

Причины

- Ручное тестирование на IBM z/OS занимает очень большое время
 - Особенно это сказывается на разработчиках
- Для непрерывной интеграции для IBM z/OS не существует открытой инструментальной поддержки
- SCLM – одно из популярнейших средств управления изменениями для IBM z/OS

Цель и задачи

- **Цель – создание открытого инструмента для непрерывной интеграции для IBM z/OS с поддержкой SCLM**
- **Задачи:**
 - **Разработка инструмента**
 - **Внедрение инструмента и непрерывной интеграции в процесс разработки ПО для мейнфреймов**
 - **Анализ результатов**

Непрерывная интеграция и IBM z/OS

- **Непрерывная интеграция – практика регулярного (лучше – несколько раз в день) слияния рабочих копий в главную с регулярным тестированием главной ветви**
- **Варианты инструментальной поддержки при разработке ПО под IBM z/OS:**
 - **Использование готовых инструментов**
 - **Самостоятельная разработка инструмента**
 - **Расширение готового инструмента**

Готовые решения?

- Нет готовых инструментов, полностью решающих поставленные задачи:
 - Либо автоматизация, либо поддержка SCLM
 - Все решения проприетарны
- Рассмотренные средства автоматизации:
 - IBM Tivoli NetView
 - SEA ODDS
 - BMC MainView
 - VNAC
- Средства с поддержкой SCLM:
 - RDz
 - SCLM Developer Toolkit

Расширение существующего инструмента

- Рассматривались инструменты:
 - Jenkins CI
 - GitLab CI
 - Go (Continuous Delivery)
- Для расширения был выбран Jenkins CI:
 - Функциональность
 - Открытость
 - Гибкость
 - Расширяемость
 - Документация

Решение №1: Serv

- Самостоятельная разработка
- Планировщик регулярного запуска запуска пакетных заданий (подобно Cron)
- Написан на С и Rexx
- Работает полностью в среде IBM z/OS
- Поддержка команд MVS: STOP и MODIFY
- Ведётся журнал запуска пакетных заданий

Пример журнала Serv

```
Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY R$SERV$M J0167742 DSID 101 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> SCROLL ==> CSR
***** TOP OF DATA *****
JCLLIB = 'USER1.PUB.SERV.TEST.JCL'
Add a JOB: WAIT5 000:00:00 000:00:13
Add a JOB: WAIT10 000:00:01 000:00:22
Add a JOB: WAIT33 000:00:02 000:00:41
Add a JOB: WAIT3 000:00:00 000:00:57
[ 7 May 2015, 00:01:21]: submit Job: WAIT5
Will refresh parms
[ 7 May 2015, 00:01:40]: finished Job: WAIT5
Add a JOB: WAIT5 000:00:00 000:00:13
Add a JOB: WAIT10 000:00:01 000:00:22
Add a JOB: WAIT33 000:00:02 000:00:41
Add a JOB: WAIT3 000:00:00 000:00:57
[ 7 May 2015, 00:01:41]: submit Job: WAIT5
Will end work
[ 7 May 2015, 00:01:50]: finished Job: WAIT5
Work finished
***** BOTTOM OF DATA *****

*ISFPCU4
```

Построение списка задач

Запуск 1го задания

Реакция на команду MODIFY

Конец 1го задания

Реакция на команду STOP

Решение №2: IBM zOS Connector

- Расширение для Jenkins CI
- Написан на Java и Jelly
- Поддержка SCLM в качестве SCM (системы управления изменениями)
- Запуск пакетных заданий в IBM z/OS:
 - Ожидание завершения работы пакетного задания
 - Сохранение журналов работы пакетных заданий в Jenkins CI и удаление их из z/OS

Jenkins CI: настройка SCLM

Source Code Management

- None
- CVS
- CVS Projectset
- SCLM

Server	LPAR_Name_Or_IP	Сервер (LPAR)
Port	21	FTP порт
Username	UserID	Пользователь
Password	Пароль
Project	PROJECT	Проект в SCLM
Alternate	ALTER	Дополнительное имя проекта
Group	GROUP	Группа в проетке
Types	HLASM	Типы отслеживаемых типов

- Use custom job header?

Job header	<pre>//CUSTOM JOB (ACCOUNT),'JENKINS', // MSGCLASS=A,CLASS=A,NOTIFY=&SYSUID</pre>	Особый заголовок пакетного задания для работы в IBM z/OS
------------	--	---

- Custom SCLM job step?

Jenkins CI: Пакетное задание

Submit zOS Job

Server	LPAR_Name_Or_IP	Сервер (LPAR)
Port	21	FTP порт
Username	UserID	Пользователь
Password	Пароль
Job	<pre>//WAIT5 JOB //WAIT05 EXEC PGM=WAIT,PARM=5</pre>	Пакетное задание для запуска

Wait for completion? **Настройка времени ожидания и**

Time to wait (in minutes) **удаления журнала из IBM z/OS**

0 = wait forever

Delete job log from Spool?

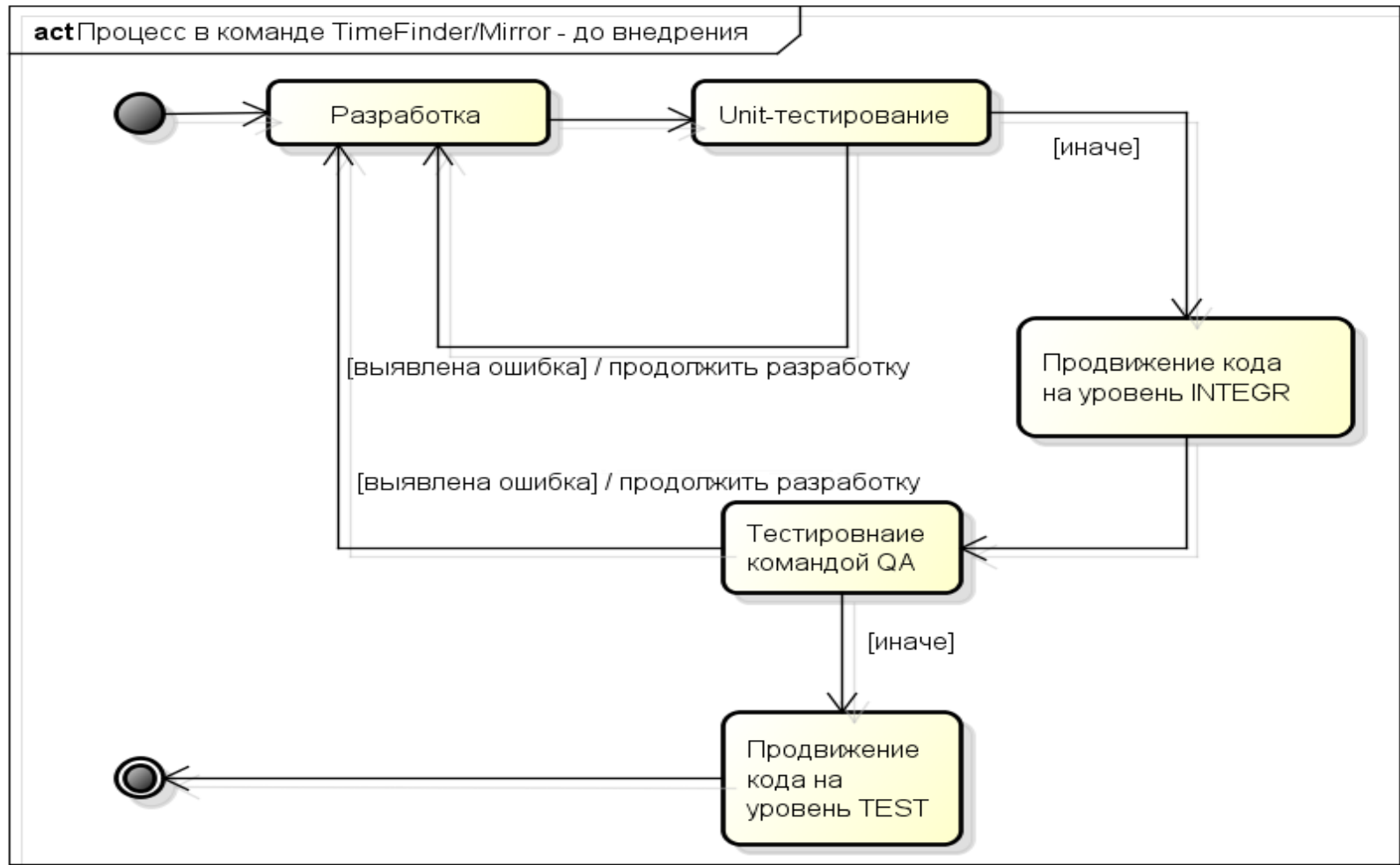
Внедрение

- В компании EMC, в команде Mainframe Enablers – TimeFinder/Mirror
- При внедрении Jenkins CI вместе с расширением IBM zOS Connector произведено перестроение процесса на использование непрерывной интеграции

Процесс до внедрения

- Разработка построена на 2-х недельных итерациях
- Во время итерации изменения проходили 2 этапа тестирования:
 - Ручное unit-тестирование
 - Обязательное тестирование командой QA (тестировщики)
- Продвижение кода 1 раз в итерацию

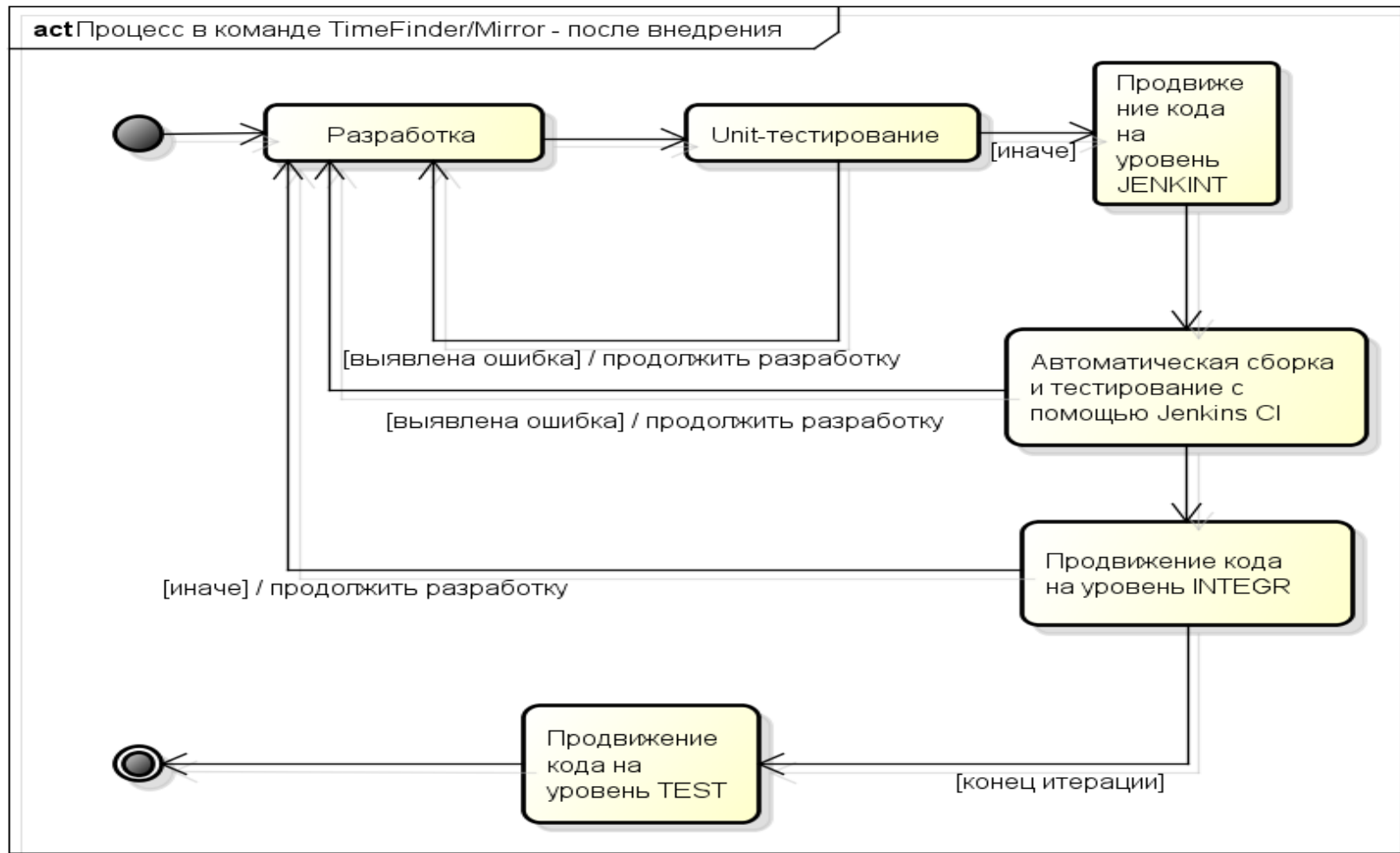
Процесс до внедрения



Результаты внедрения

- **Переход на разработку через тестирование**
 - **Перед разработкой производится создание контрольных тестов для принятия изменения**
- **Полностью автоматизированное тестирование разработчиками**
 - **Время тестирования уменьшилось в 1,5–2 раза**
- **Несколько продвижений кода в итерацию**
 - **Специальная SCLM-группа – JENKINT**
- **Исключение тестирования отделом QA из итерации**

Процесс после внедрения

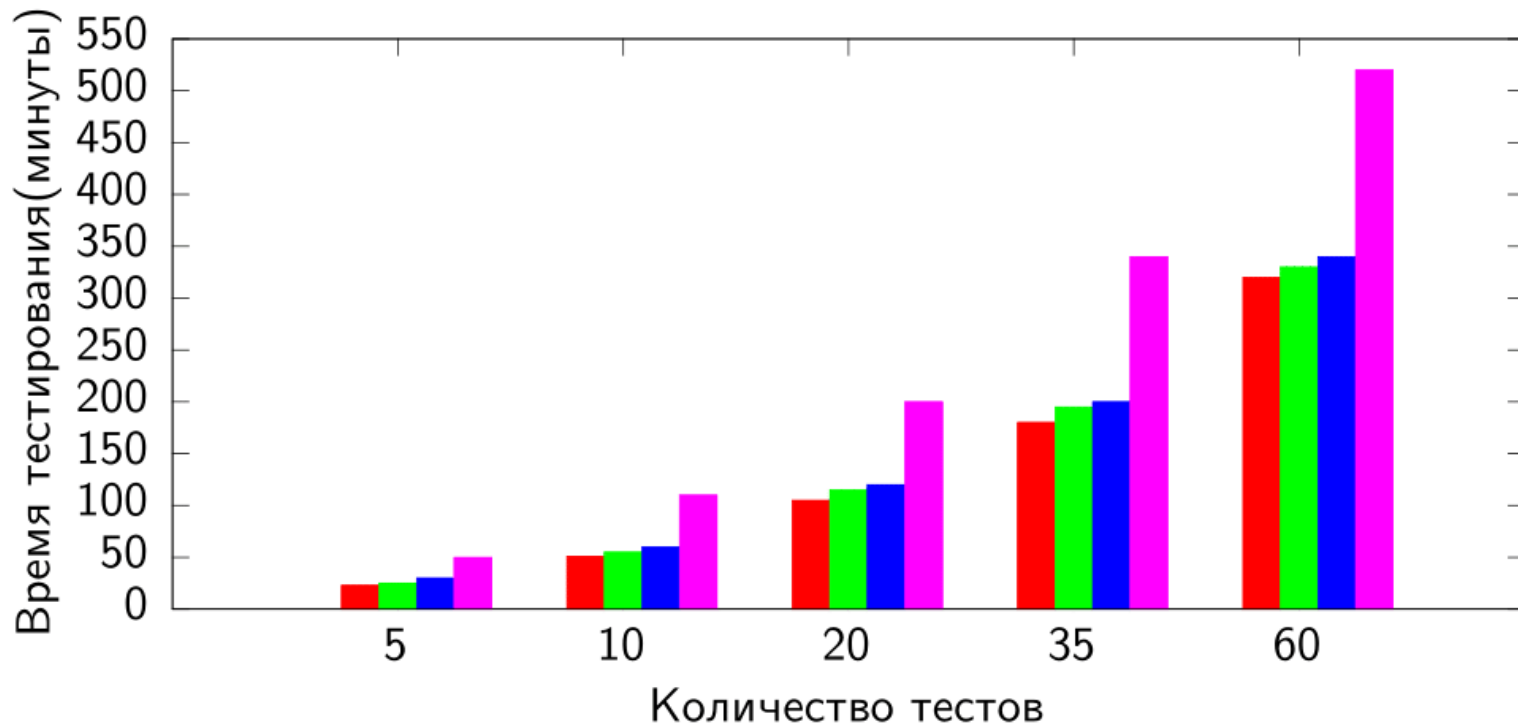


Время тестирования (минуты)

Тип тестирования	Количество тестов в наборе				
	5	10	20	35	60
Ручное тестирование	50	110	200	340	520
QAUTO (тестировщиками)	30	60	120	200	340
Jenkins CI	25	55	115	195	330
“Чистое” время работы тестов	23	51	105	180	320

Время тестирования (минуты)

Чистое время █
Jenkins CI █
QAUTO █
Ручное тестирование █



Результаты

- Созданы 2 открытых инструмента для непрерывной интеграции для IBM z/OS с поддержкой SCLM:
 - [IBM zOS Connector](#) – расширение Jenkins CI
 - [Serv](#)
- Произведено внедрение непрерывной интеграции в процесс разработки ПО для мейнфреймов с помощью IBM zOS Connector
- Произведён анализ результатов внедрения