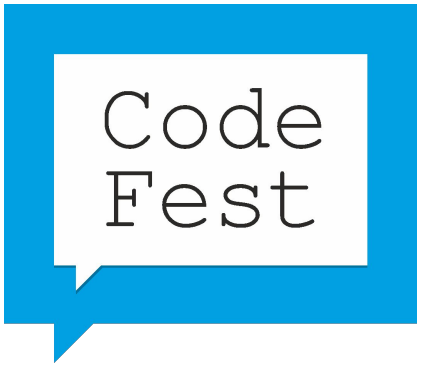


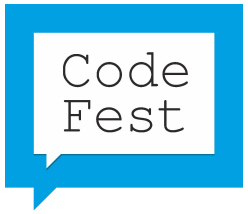
Эффективные методики автоматизированного тестирования в условиях непрерывной интеграции

Сергей Андреев,
JetBrains

Code Fest

О себе
(это очень важный слайд)

JetBRAINS
Develop with pleasure!

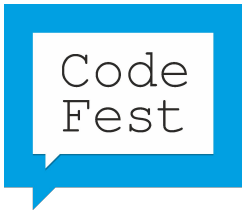


Программы сами себя не напишут (с).

Проблемы:

- где рванёт после коммита?
- совместимы ли коммиты с жизнью приложения?
- тестерам постоянно нужна новая жертва
- как-то это все затягивается....

JetBRAINS
Develop with pleasure!



CI - введение

Непрерывная интеграция (Continuous Integration (CI)) – практика разработки ПО, когда все члены команды интегрируют результаты своей работы с некоторой частотой, обычно каждый участник интегрирует хотя бы раз в день, что приводит ко множетвенным интеграциям в течение дня.

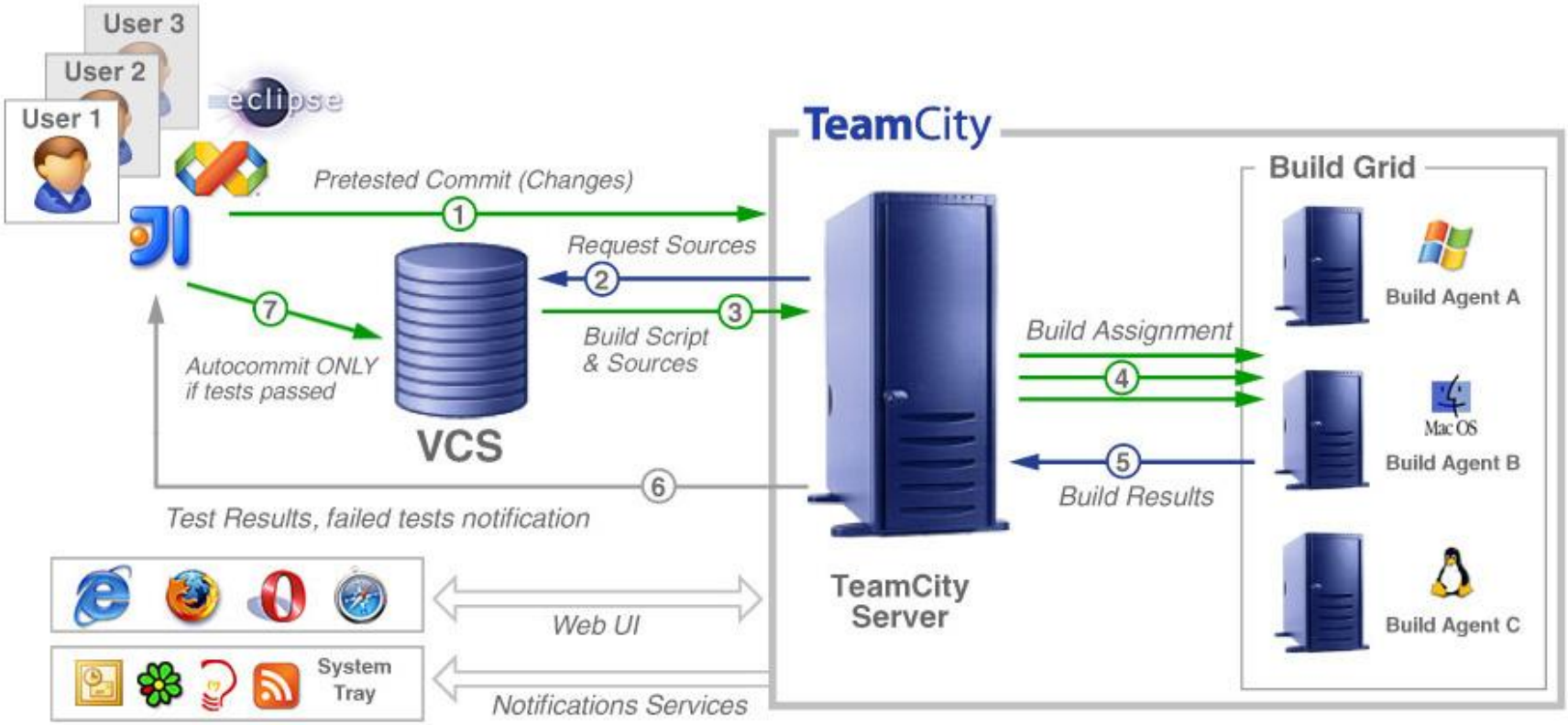
Вы хочите ссылок? Их есть у меня!

http://en.wikipedia.org/wiki/Continuous_Integration

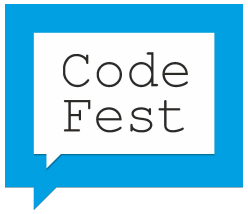
http://www.martinfowler.com/articles/continuousIntegration.html

Code Fest

CI с иллюстрациями



JetBRAINS
Develop with pleasure!



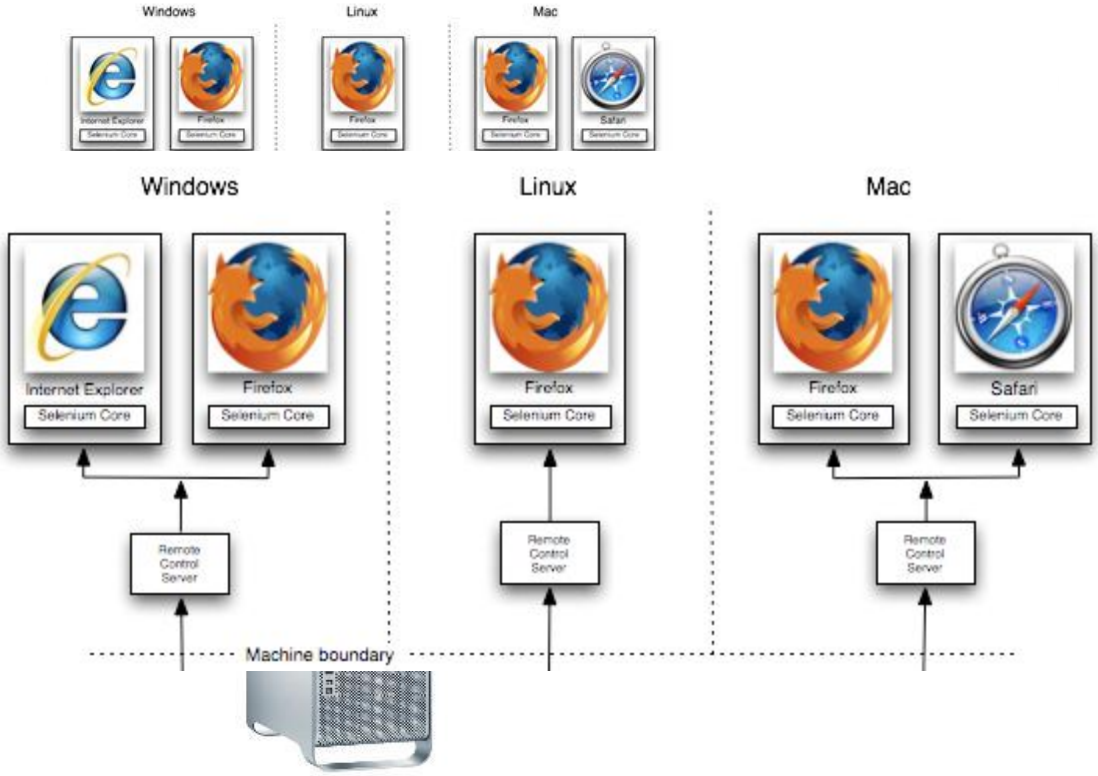
В итоге:

В итоге:

1. «Безопасность» изменений
2. Работоспособность приложения после изменений
3. Свежий билд для работы
4. ???
5. Экономия времени

Code Fest

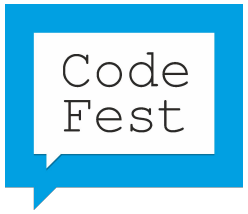
Как же без Selenium?




```
<jvmarg value="-Dhost=${seleniumhub.hostname}" />
<jvmarg value="-Durl=http://${env.HOSTNAME}:${web.port}" />
<jvmarg value="-Dbrowser=*${browser}" />
```

JetBRAINS Develop with pleasure!

Additional Ant command line parameters: -Dbrowser.name=googlechrome -Dseleniumhub.hostname=



Code Coverage

COVERAGE SUMMARY FOR PACKAGE [jetbrains.[redacted]]

name	class, %	method, %	block, %	line, %
jetbrains.[redacted]	60% (47/78)	56% (145/257)	40% (2681/6742)	45% (623.6/1371)

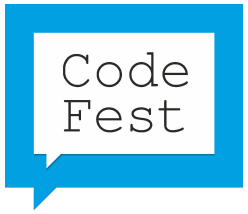
COVERAGE BREAKDOWN BY SOURCE FILE

name	class, %	method, %	block, %	line, %
[redacted].java	0% (0/1)	0% (0/4)	0% (0/134)	0% (0/41)
[redacted].java	0% (0/1)	0% (0/3)	0% (0/193)	0% (0/33)
[redacted].java	0% (0/15)	0% (0/35)	0% (0/610)	0% (0/92)
[redacted].java	0% (0/1)	0% (0/3)	0% (0/56)	0% (0/20)
[redacted].java	50% (1/2)	20% (1/5)	1% (4/411)	8% (2/26)
[redacted].java	50% (1/2)	20% (1/5)	1% (4/285)	8% (2/25)
[redacted].java	100% (1/1)	50% (3/6)	18% (75/426)	21% (16/76)
[redacted].java	100% (1/1)	67% (6/9)	19% (58/302)	24% (14.2/59)
[redacted].java	44% (7/16)	33% (12/36)	22% (113/509)	24% (22/90)
[redacted].java	100% (1/1)	67% (2/3)	40% (183/454)	43% (57/133)
[redacted].java	67% (4/6)	59% (19/32)	41% (192/474)	38% (39.7/104)
[redacted].java	100% (1/1)	100% (2/2)	42% (18/43)	46% (6/13)
[redacted].java	100% (2/2)	100% (4/4)	43% (81/187)	45% (14/31)
[redacted].java	100% (1/1)	100% (2/2)	44% (15/34)	60% (6.6/11)
[redacted].java	100% (2/2)	79% (19/24)	46% (311/669)	47% (73.5/155)
[redacted].java	100% (1/1)	50% (1/2)	50% (3/6)	50% (1/2)
[redacted].java	100% (1/1)	100% (4/4)	67% (42/63)	64% (9/14)
[redacted].java	100% (1/1)	50% (1/2)	70% (7/10)	33% (1/3)
[redacted].java	100% (2/2)	89% (8/9)	76% (81/106)	60% (18/30)
[redacted].java	100% (4/4)	100% (16/16)	77% (203/264)	82% (50.2/61)
[redacted].java	100% (1/1)	100% (2/2)	78% (73/94)	80% (17.7/22)
[redacted].java	100% (1/1)	67% (2/3)	79% (19/24)	67% (8/12)
[redacted].java	100% (2/2)	86% (6/7)	82% (50/61)	76% (9.8/13)
[redacted].java	100% (1/1)	67% (2/3)	82% (14/17)	88% (7/8)
[redacted].java	100% (1/1)	67% (4/6)	83% (25/30)	80% (8/10)
[redacted].java	100% (1/1)	100% (3/3)	84% (751/898)	83% (166/201)
[redacted].java	100% (1/1)	80% (4/5)	84% (16/19)	86% (6/7)

Code Fest

Unit Tests

JetBRAINS
Develop with pleasure!



JMeter

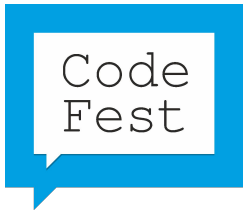
Вопрос залу: На сколько процентов JMeter лучше, чем ничего?

Ответ:
Как минимум в 2 раза больше.

Функциональное тестирование RESTful интерфейса

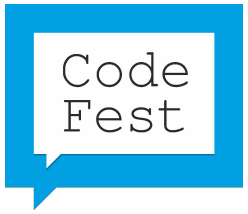
+ Легко создавать тесты
+ Навыков программирования не нужно
- Тесты неуклюжие

JetBRAINS
Develop with pleasure!



Внезапно...

DSL

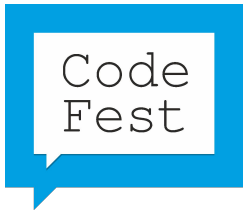


Хорошо, что DSL...

```
tst.openURL("http://www.google.com/webhp?complete=1&hl=en");

tst.typeSmth("Cheese", "q");

tst.waitForElementPresent("gac_m", 5000);
```

JetBRAINS
Develop with pleasure!

http://code.google.com/p/selenium/wiki/GettingStarted



DSL

Предметно-ориентированный язык программирования (англ. *domain-specific programming language, domain-specific language, DSL*) — язык программирования, специально разработанный для решения определённого круга задач, в отличие от *языков программирования общего назначения*, таких, как Си, или *языков моделирования общего назначения* наподобие UML, PostScript, SQL и др.

http://ru.wikipedia.org/wiki/Предметно-ориентированный_язык_программирования

Code Fest

DSL – JMeter Lang

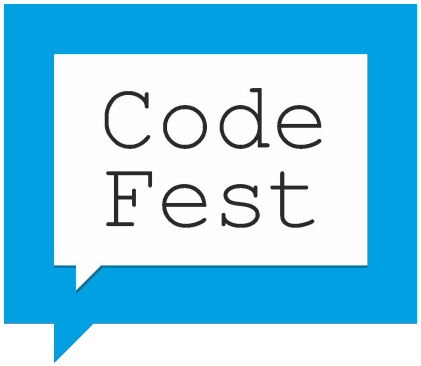
```
request Get all bundles for get  {
  parameters:
    << no parameters >>
  assert:
    code 200: OK
    text equals <?xml version="1.0" encoding="UTF-8" standalone...
}

request Remove group from bundle for delete /{bundleName}/group/{fieldValue} {
  parameters:
    path bundleName = newBundleName
    path fieldValue = "idea-developers"
  assert:
    code 200: OK
}

request <no name> for <no method> {
  parameters:
    << no parameters >>
  assert:
    << no asserts >>
}
```

JetBRAINS Develop with pleasure!

Code Fest

PolePosition

JetBRAINS Develop with pleasure!



Пожалуй хватит.
Спасибо за внимание!

Сергей Андреев,
JetBrains

sergey.andreev@jetbrains.com
smandreev@gmail.com