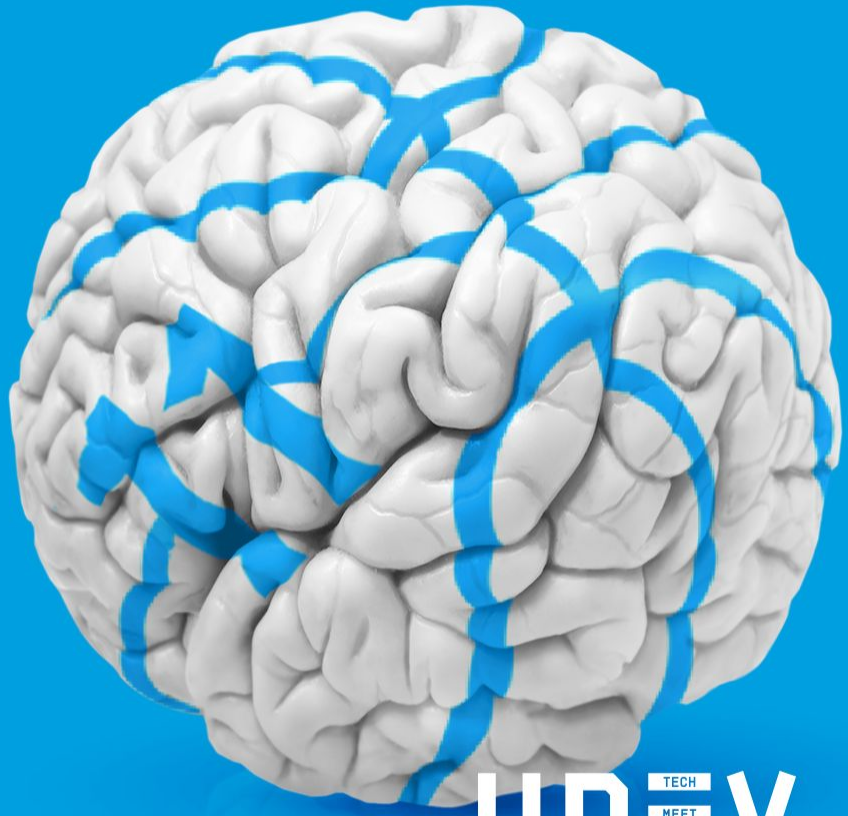

PLARIUM

AUTOMATION BUILDS WITH CAKE

ЧТО, ЗАЧЕМ И КАКИМ ОБРАЗОМ



Артем Дурнев
Unity Software Architect

UDEV TECH MEET UPS

СОДЕРЖАНИЕ

1 Что такое **Cake.** Описание и ключевые преимущества

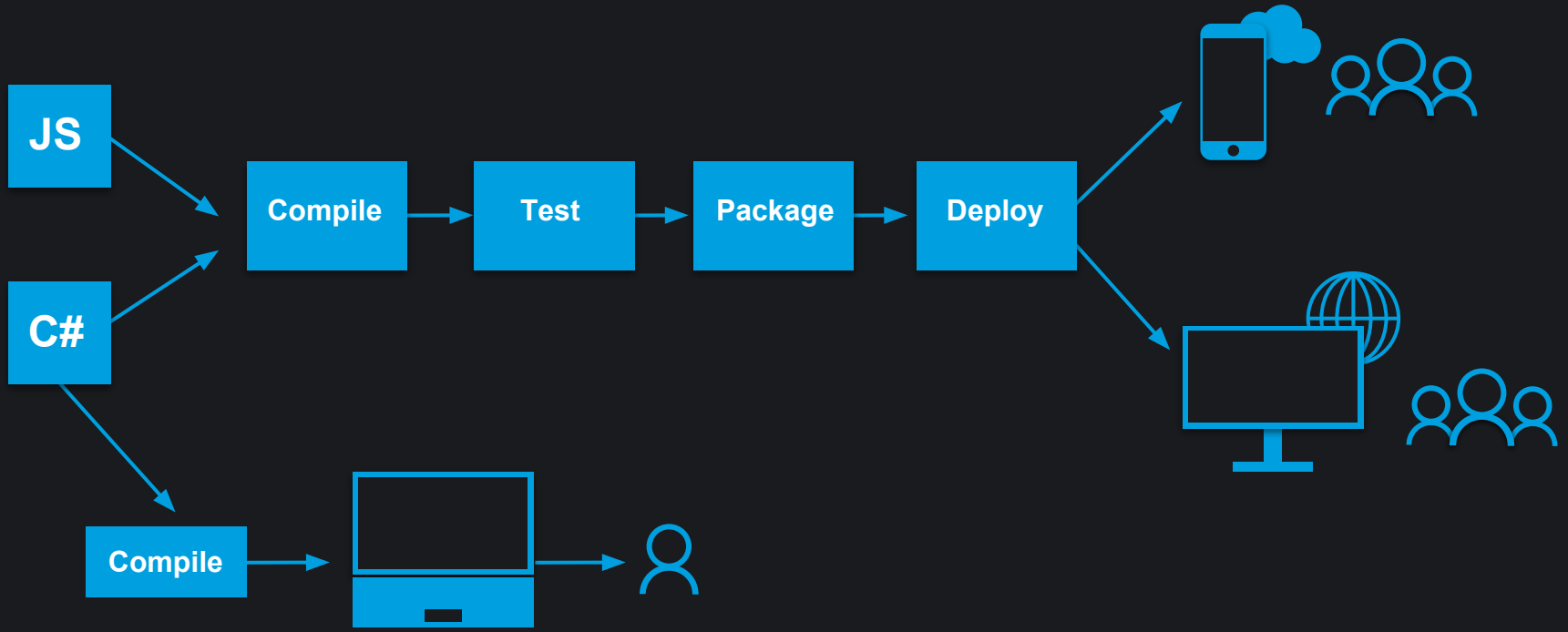
2 Основы работы с **Cake**

3 Разбор примеров использования **Cake** в наших проектах

CAKE. ЧТО ЭТО?

ОПИСАНИЕ И КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА

ПРОБЛЕМА



ПРОБЛЕМА

- Несколько языков разработки
- Множество стадий билд процесса
- Различное поведение в зависимости от окружения

ПРОБЛЕМА – COMPILE

- Чекаут из разных репозиториев
- Восстановление пакетов npm / nuGet
- Разрешение зависимостей
- Версионирование
- Обфускация кода
- Ветвления логики компиляции

ПРОБЛЕМА – TEST

- Unit тестирование
- Определение покрытия кода тестами
- Статический анализ кода
- UI тесты

ПРОБЛЕМА – PACKAGE

- Подпись кода
- Создание релиз ноутов
- Генерация документации

ПРОБЛЕМА – DEPLOY

- Zip
- Миграция баз данных
- Работа с FTP
- Работа с пулами приложений
- Нотификации про выполнение операции

АЛЬТЕРНАТИВЫ

- Попросить вылить того, кто уже выливал
- CI сервер
- Написать свой скрипт

АЛЬТЕРНАТИВЫ – РУКАМИ

- Каждый раз по новому
- «У меня все работает»
- Нет никакой периодичности
- Занимает много времени
- Отпуски и больничные
- Через пол года уже никто ничего не помнит

АЛЬТЕРНАТИВЫ – CI СЕРВЕР

- Сложно дебажить процесс билда
- Конфигурация находится не под version-контролем
- Нет связи между решением и определением процесса билда
- Для некоторых CI систем новый синтаксис

АЛЬТЕРНАТИВЫ – СВОЙ СКРИПТ

- PowerShell – мощный язык, но им нужно владеть
- Проблемы с мультиплатформенность (ее нет)
- Огромное количество подводных камней

КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА

Cake (C# Make) is a cross-platform build automation system with a C# DSL for tasks such as compiling code, copying files and folders, running unit tests, compressing files and building NuGet packages.

КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА

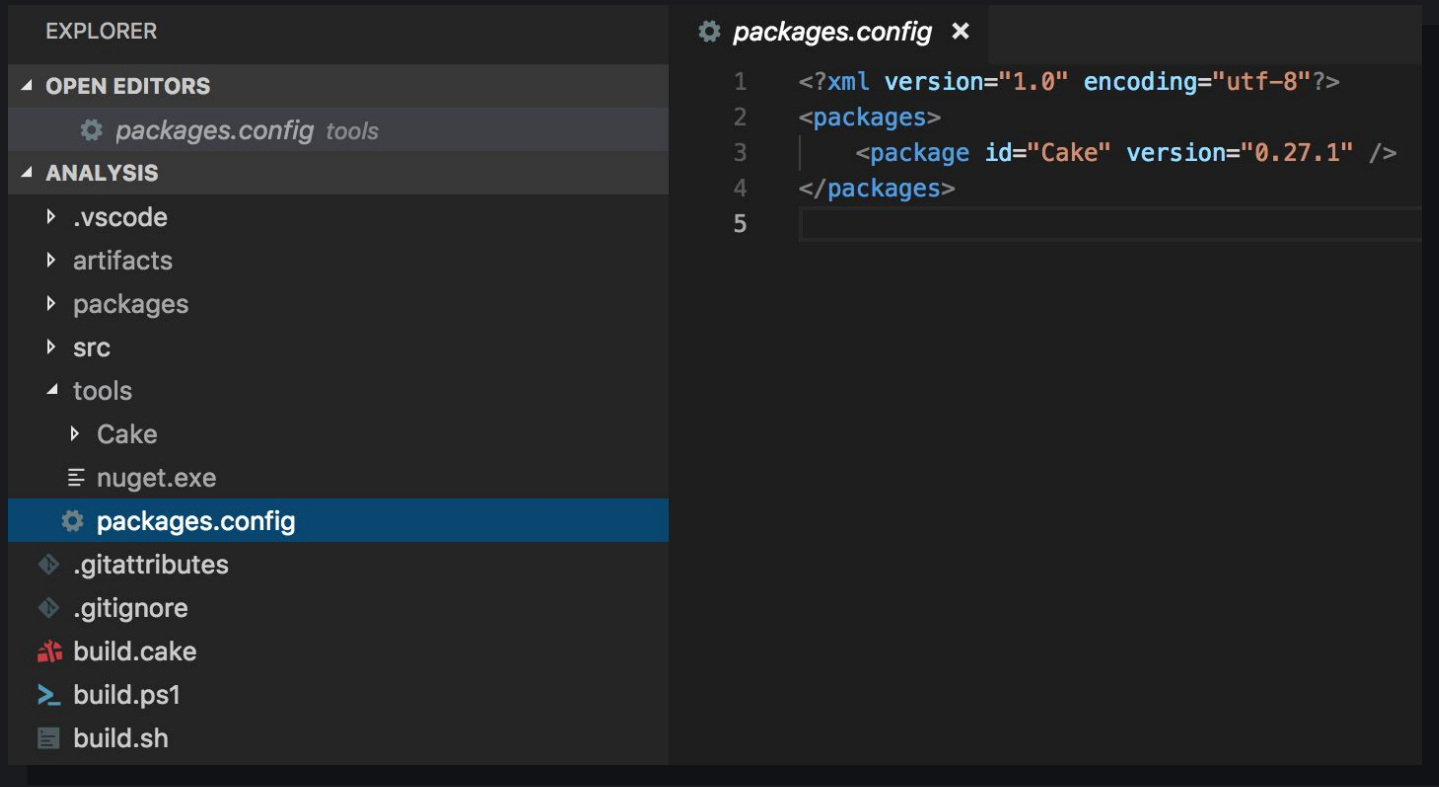
- Полный контроль над процессом билда
- Мультиплатформенность – поддерживается запуск на Windows, Linux, macOS
- Для написания кода используется C# с всеми его преимуществами
- Легкость расширения функциональности через nuGet
- Open Source. Разработка ведется на GitHub
- Есть каналы для общения с мейнтейнерами в Gitter
- Огромная экосистема. 112 расширений для любых потребностей, среди которых MSBuild, Git, SemVer2.0.0, etc

CAKE. ОСНОВЫ РАБОТЫ

СТРУКТУРА ФАЙЛОВ ДЛЯ CAKE

- *build.ps1/build.sh*
  - Bootstrapper скрипты, которые обеспечивают установку зависимостей для Cake и правильный запуск
  - Написаны на PowerShell для Win и Bash для Linux/Mac
  - Обеспечивают валидацию параметров и запуск нужного таска на выполнение
  - Может быть изменен под нужды проекта
- *build.cake*
  - Основной скрипт, написанный на C# с некоторыми особенностями
  - Содержит основную логику
  - Имя может отличаться от стандартного
- *tools/packages.config* (не обязательно)
  - Говорит bootstrapper скрипту, какие nuGet пакеты и какой версии устанавливать перед запуском

СТРУКТУРА ФАЙЛОВ ДЛЯ CAKE



IDE

- **Visual Studio**
  - Нужно установить расширение
  - Поддержка запуска тасков на выполнение, с выводом во внутреннюю консоль
  - Поддержка Debug
  - Нет intellisense, но есть подсветка синтаксиса

- **VS Code**
  - Нужно установить расширение
  - Поддержка запуска тасков на выполнение, с выводом во внутреннюю консоль
  - Поддержка Debug
  - Подсветка синтаксиса и intellisense
  - Дополнительные команды и сниппеты
  - ИМХО работает не всегда идеально, особенно на Mac

TASKS

- Основной юнит выполнения Cake
- Могут иметь зависимости, критерии и специфическую обработку ошибок

build.cake

```
run task | debug task
1  Task("Default")
2    .Does(() => {
3      Information("Hello Plarium!");
4    });
5
6  RunTarget("Default");
```

TERMINAL 1: zsh

```
→  cake-example ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/cake-example/tools/packa
ges.config are already installed.

========================================
Default
========================================
Hello Plarium!

Task                          Duration
--------------------------------------------------
Default                       00:00:00.0069200
--------------------------------------------------
Total:                        00:00:00.0069200
→  cake-example
```

TASKS

- Указывает какой таск будет запускаться
- Всегда должен быть в конце
- У тасков нет параметров

build.cake

```
run task | debug task
1  Task("Default")
2    .Does(() => {
3      Information("Hello Plarium!");
4    });
5
6  RunTarget("Default");
```

ЗАВИСИМОСТИ

- Таски могут быть зависимы друг от друга
- Зависимостей может быть множество
- Каждый таск выполняется только 1 раз

ЗАВИСИМОСТИ

build.cake

```
run task | debug task
Task("Clean")
  .Does(() => {
    Information("Cleaning...");
  });
run task | debug task
Task("Checkout")
  .IsDependentOn("Clean")
  .Does(() => {
    Information("Checkouting...");
  });

run task | debug task
Task("Compile")
  .IsDependentOn("Checkout")
  .IsDependentOn("Clean")
  .Does(() => {
    Information("Compiling...");
  });

run task | debug task
Task("Test")
  .IsDependentOn("Compile")
  .Does(() => {
    Information("Testing...");
  });

RunTarget("Test");
```

TERMINAL 1: zsh

```
→  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advice/cake-exampl
es#2/tools/packages.config are already installed.

========================================
Clean
========================================
Cleaning...

========================================
Checkout
========================================
Checkouting...

========================================
Compile
========================================
Compiling...

========================================
Test
========================================
Testing...

Task                          Duration
--------------------------------------------------
Clean                         00:00:00.0056814
Checkout                      00:00:00.0001497
Compile                       00:00:00.0001155
Test                          00:00:00.0001136
--------------------------------------------------
Total:                        00:00:00.0060602
→  cake-examples#2
```

КРИТЕРИИ

- Необходимы для контроля выполнения билд скрипта
- Критерии вычисляются на момент создания задачи
- Если нужно отложить вычисление до момента выполнения задачи – стоит использовать лямбды
- Обычно используются для изменения логики в зависимости от окружения, бранча, прочее

КРИТЕРИИ

build.cake

```
var isLocalBuild = BuildSystem.IsLocalBuild;
var isRecoveryMode = false;

run task | debug task
Task("Checkout")
    .WithCriteria(!isLocalBuild)
    .Does(() =>
    {
      Information("Checkouting...");
    });

run task | debug task
Task("Migration")
  .Does(() =>
  {
    Information("Migration...");

    isRecoveryMode = true;
  });

run task | debug task
Task("Recover")
  .WithCriteria(() => isRecoveryMode)
  .Does(() =>
  {
    Information("Recovering...");
  });

run task | debug task
Task("Deploy")
  .IsDependentOn("Checkout")
  .IsDependentOn("Migration")
  .IsDependentOn("Recover")
  .Does(() =>
  {
    Information("Deploying...");
  });

RunTarget("Deploy");
```

TERMINAL 1: zsh

```
➜  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advic
e/cake-examples#2/tools/packages.config are already instal
led.

========================================
Migration
========================================
Migration...

========================================
Recover
========================================
Recovering...

========================================
Deploy
========================================
Deploying...

Task                          Duration
--------------------------------------------------
Checkout                      Skipped
Migration                     00:00:00.0053332
Recover                       00:00:00.0002034
Deploy                        00:00:00.0001947
--------------------------------------------------
Total:                        00:00:00.0057313
➜  cake-examples#2
```

РАБОТА С ОШИБКАМИ

- Выполнение скрипта прерывается при возникновении необработанного исключения

- Для тасков существует несколько вариантов обработки исключений:
  - `OnError` – колбек для обработки ошибки, продолжает выполнение
  - `ContinueOnError` – продолжает выполнение скрипта
  - `ReportError` – колбек для уведомления про исключение, останавливает выполнение скрипта

РАБОТА С ОШИБКАМИ

build.cake

```
run task | debug task
1  Task("Abort")
2      .Does(() =>
3      {
4          var criteria = false;
5          if (criteria.ToString().Length > 4)
6              throw new Exception("Sorry mam...");
7      });
8
9  RunTarget("Abort");
```

TERMINAL 1: zsh

```
→  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advice/cake-examples#2/tools/packages.config are already installed.

========================================
Abort
========================================
An error occurred when executing task 'Abort'.
Error: One or more errors occurred.
        Sorry mam...
→  cake-examples#2
```

РАБОТА С ОШИБКАМИ

build.cake

run task | debug task

```
1  Task("ErrorProcessing")
2      .Does(() =>
3      {
4          var criteria = false;
5          if (criteria.ToString().Length > 4)
6              throw new Exception("Sorry mam...");
7      })
8      .OnError(exception =>
9      {
10         Information("It's ok.");
11     });
12
13 RunTarget("ErrorProcessing");
```

TERMINAL 1: zsh

```
→  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advice/cake-examples#2/tools/packages.config are already installed.

========================================
ErrorProcessing
========================================
An error occurred when executing task 'ErrorProcessing'.
It's ok.

Task                          Duration
--------------------------------------------------
ErrorProcessing               00:00:00.0093932
--------------------------------------------------
Total:                        00:00:00.0093932
→  cake-examples#2
```

РАБОТА С ОШИБКАМИ

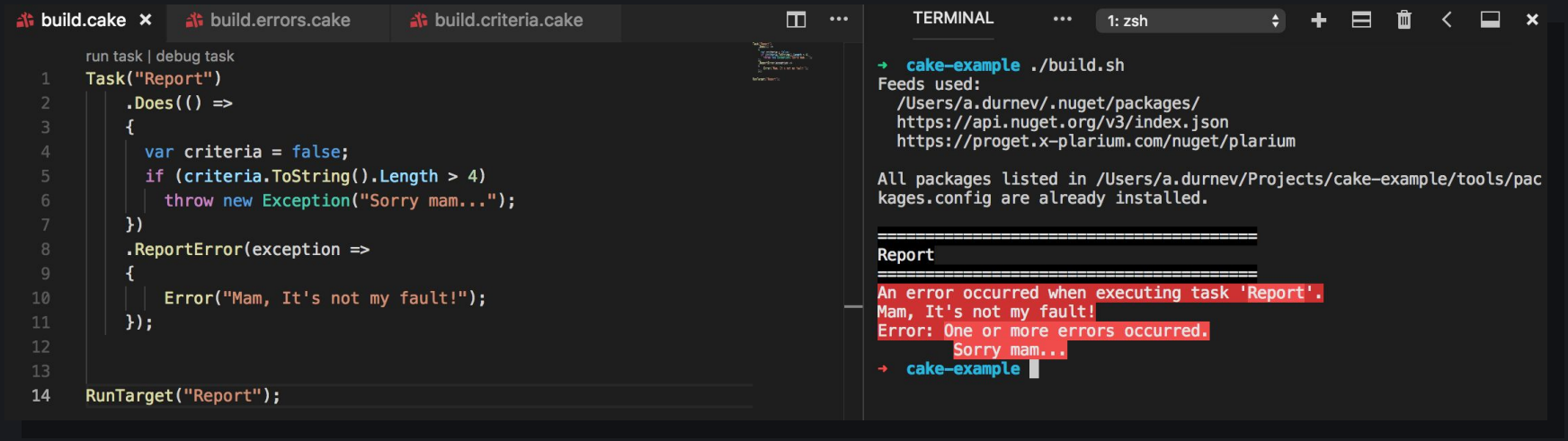
build.cake

```
run task | debug task
1  Task("ContinueOnError")
2      .ContinueOnError()
3      .Does(() =>
4      {
5          var criteria = false;
6          if (criteria.ToString().Length > 4)
7              throw new Exception("Sorry mam... mam?");
8      });
9
10 RunTarget("ContinueOnError");
```

TERMINAL 1: zsh

```
→  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advice/cake-examples#2/tools/packages.config are already installed.

========================================
ContinueOnError
========================================
An error occurred when executing task 'ContinueOnError'.

Task                          Duration
--------------------------------------------------
ContinueOnError               00:00:00.0076059
--------------------------------------------------
Total:                        00:00:00.0076059
→  cake-examples#2
```

РАБОТА С ОШИБКАМИ



PRE/POST TASKS

- `Setup` – существует для выполнения перед первым таском
- `Teardown` – существует для выполнения после последнего таска

build.cake

```
1  Setup(ctx =>
2  {
3    Information("Prepearing...");
4  });
5
6  Teardown(ctx =>
7  {
8    Information("Finishing...");
9  });
10
   run task | debug task
11 Task("Default")
12   .Does(() => {
13     Information("Hello Plarium!");
14   });
15
16 RunTarget("Default");
```

TERMINAL 1: zsh

```
→  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/te
h-advice/cake-examples#2/tools/packages.config are
already installed.

-------------------------------------------------
Setup
-------------------------------------------------
Prepearing...

=================================================
Default
=================================================
Hello Plarium!

-------------------------------------------------
Teardown
-------------------------------------------------
Finishing...

Task                          Duration
-------------------------------------------------
Setup                         00:00:00.0033748
Default                       00:00:00.0035810
Teardown                      00:00:00.0004798
-------------------------------------------------
Total:                        00:00:00.0074356
→  cake-examples#2
```

АРГУМЕНТЫ

- Основной способ передачи настроек в скрипт
- Конвертируются из строкового представления в нужный вам тип
- Также можно обращаться к переменным окружения

АРГУМЕНТЫ

- Аргумент `target` – стандартный подход к вызову нужного вам таска
- Старайтесь указывать имя таска по умолчанию, чтоб упростить вызов скрипта

build.cake

```
1  var target = Argument("target", "Spartak");
2
   run task | debug task
3  Task("Spartak")
4      .Does(() => {
5          Information("Спартак здесь?");
6      });
7
   run task | debug task
8  Task("CSK")
9      .Does(() => {
10         Information("ЦСКА ЗДЕСЬ?");
11     });
12
13 RunTarget(target);
```

TERMINAL 1: zsh

```
➜  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/te
h-advice/cake-examples#2/tools/packages.config are
already installed.

========================================
Spartak
========================================
Спартак здесь?

Task                          Duration
----------------------------------------
Spartak                       00:00:00.0056903
----------------------------------------
Total:                        00:00:00.0056903
➜  cake-examples#2 ./build.sh -target=CSK
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/te
h-advice/cake-examples#2/tools/packages.config are
already installed.

========================================
CSK
========================================
ЦСКА ЗДЕСЬ?

Task                          Duration
----------------------------------------
CSK                           00:00:00.0060547
----------------------------------------
Total:                        00:00:00.0060547
➜  cake-examples#2
```

АРГУМЕНТЫ

build.cake

```
1   var target = Argument("target", "Run");
2
3   var word = Argument("word", string.Empty);
4   bool isGodExist = Argument<bool>("god");
5
    run task | debug task
6   Task("Run")
7       .Does(() => {
8         if (isGodExist)
9           Information($"And God said, {word} and there was {word}");
10      });
11
12  RunTarget(target);
```

TERMINAL 1: zsh

```
➜  cake-examples#2 ./build.sh --god=True --word=Light
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advic
e/cake-examples#2/tools/packages.config are already instal
led.

========================================
Run
========================================
And God said, Light and there was Light

Task                          Duration
--------------------------------------------------
Run                           00:00:00.0056641
--------------------------------------------------
Total:                        00:00:00.0056641
➜  cake-examples#2
```

АРГУМЕНТЫ

build.cake

```
1   var target = Argument("target", "Run");
2
3   var word = Argument("word", string.Empty);
4   bool isGodExist = Argument<bool>("god");
5
    run task | debug task
6   Task("Run")
7       .Does(() => {
8         if (isGodExist)
9           Information($"And God said, {word} and there was {word}");
10      });
11
12  RunTarget(target);
```

TERMINAL 1: zsh

```
➜  cake-examples#2 ./build.sh --god=True
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advic
e/cake-examples#2/tools/packages.config are already instal
led.

========================================
Run
========================================
And God said,  and there was

Task                          Duration
--------------------------------------------------
Run                           00:00:00.0058531
--------------------------------------------------
Total:                        00:00:00.0058531
➜  cake-examples#2 ./build.sh --word=Light
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advic
e/cake-examples#2/tools/packages.config are already instal
led.
Error: One or more errors occurred.
       Argument 'god' was not set.
➜  cake-examples#2
```

АРГУМЕНТЫ

build.cake

```
var target = Argument("target", "Run");

var word = EnvironmentVariable("word") ?? "Light";
bool isGodExist() => Argument<bool>("god");

run task | debug task
Task("Clear")
    .Does(() => {
      Information("Clearing...");
    });

run task | debug task
Task("Run")
    .Does(() => {
      if (isGodExist())
        Information($"And God said, {word} and there was {word}");
    });

RunTarget(target);
```

TERMINAL 1: zsh

```
→  cake-examples#2 ./build.sh -target=Clear
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advic
e/cake-examples#2/tools/packages.config are already instal
led.

========================================
Clear
========================================
Clearing...

Task                          Duration
--------------------------------------------------
Clear                         00:00:00.0054885
--------------------------------------------------
Total:                        00:00:00.0054885
→  cake-examples#2
```

ДИРЕКТИВЫ ПРЕПРОЦЕССОРА

- Директивы используются для загрузки в скрипт различных расширений, скриптов и nuGet библиотек
- Можно использовать пакеты как с nuGet сервера, так и локальные скрипты или либы

ДИРЕКТИВЫ ПРЕПРОЦЕССОРА

build.cake

```
#load nuget:[PlariumFeed]?package=Advisory.Cake
#tool nuget:?package=NUnit.ConsoleRunner

#load "MessageProvider.cake"

var target = Argument("target", "Fight");

run task | debug task
Task("Fight")
  .Does(() => {
      var provider = new MessageProvider();

      var message = new StringBuilder();
      message.AppendLine($"- {provider.KansiroSays()}");
      message.AppendLine($"- {provider.EnemyAnswer()}");

      Information(message.ToString());
  });

RunTarget(target);
```

MessageProvider.cake

```
class MessageProvider
{
  public string KansiroSays()
  {
    return "Omae wa mou shindeiru";
  }

  public string EnemyAnswer()
  {
    return "Nani?!?!?!";
  }
}
```

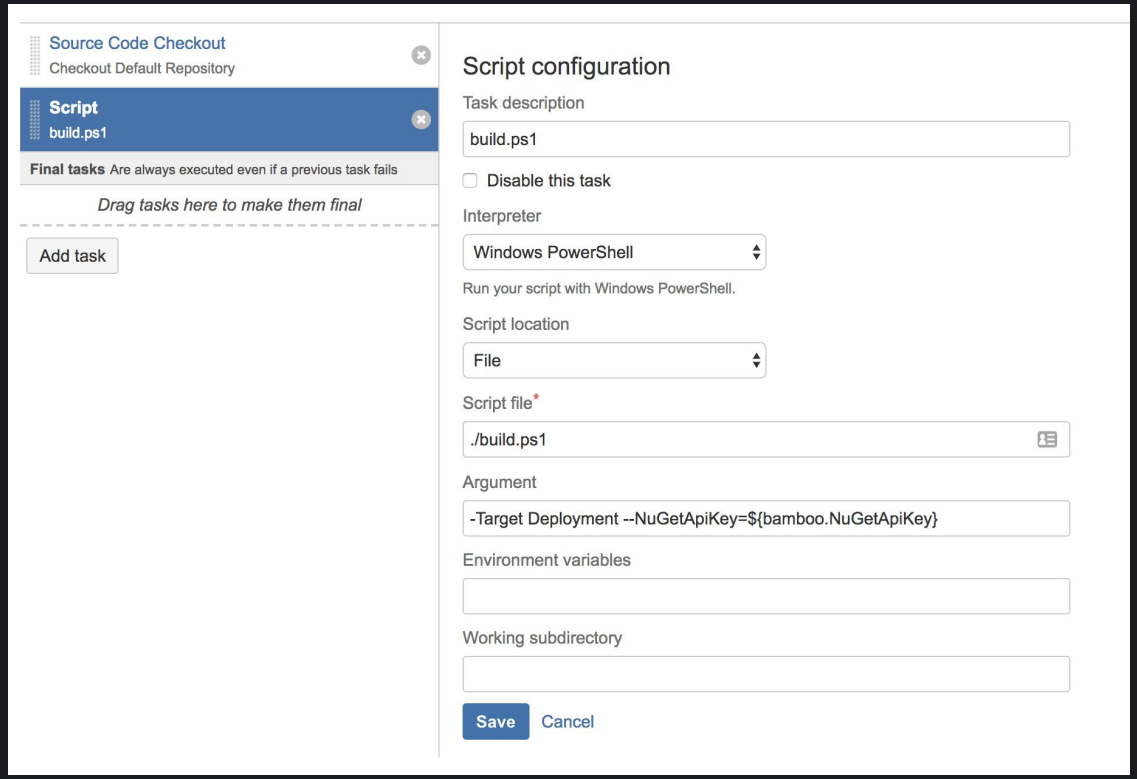
TERMINAL 1: zsh

```
➜  cake-examples#2 ./build.sh
Feeds used:
  /Users/a.durnev/.nuget/packages/
  https://api.nuget.org/v3/index.json
  https://proget.x-plarium.com/nuget/plarium

All packages listed in /Users/a.durnev/Projects/tech-advic
e/cake-examples#2/tools/packages.config are already instal
led.
The assembly 'Cake.Slack, Version=0.12.0.0, Culture=neutra
l, PublicKeyToken=null'
is referencing an older version of Cake.Core (0.26.0).
For best compatibility it should target Cake.Core version
0.28.0.

========================================
Fight
========================================
- Omae wa mou shindeiru
- Nani?!?!?!

Task                          Duration
--------------------------------------------------
Fight                         00:00:00.0034192
--------------------------------------------------
Total:                        00:00:00.0034192
➜  cake-examples#2
```

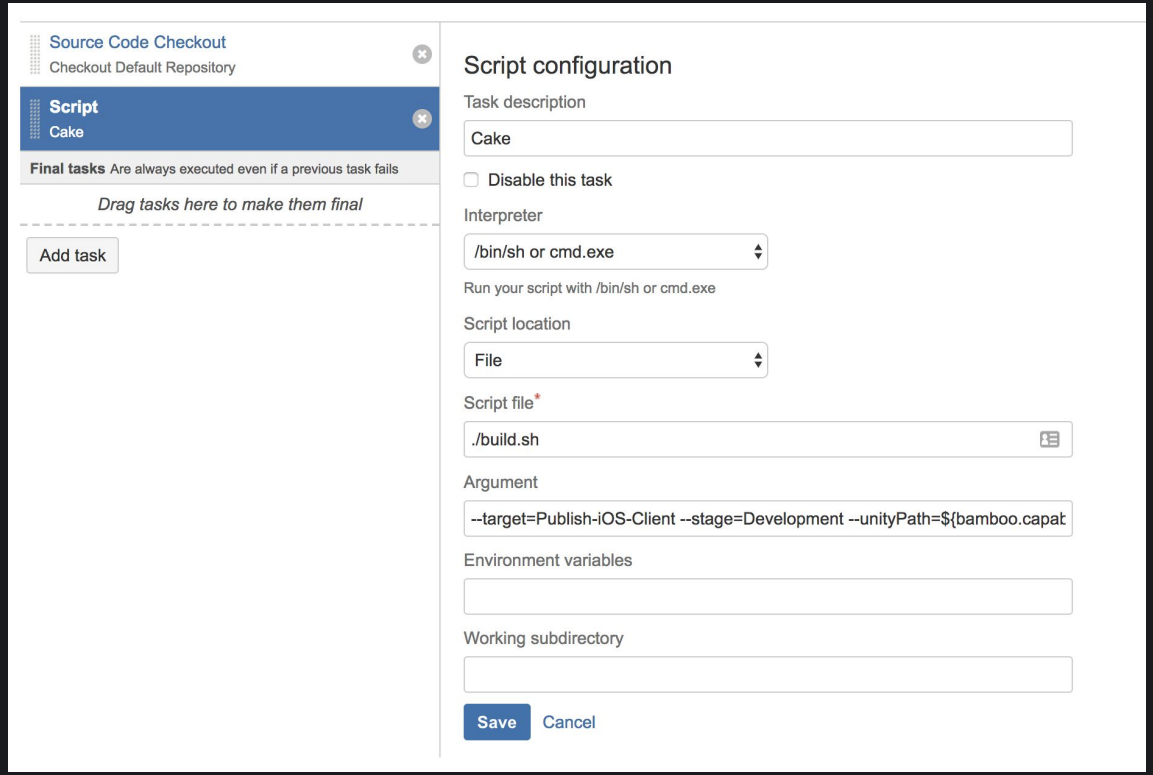
АРГУМЕНТЫ КОМАНДНОЙ СТРОКИ

- Существуют для передачи аргументов в билд скрипт
- Формат передачи аргументов в sh и ps1 bootstrap скрипты – могут отличаются
- Формат передачи аргументов также отличается в разных версиях bootstrap скриптов
- Из полезных аргументов можно выделить:
  - `-dryrun` – выполняет сборку скрипта и выводит очередь выполнения тасков
  - `-verbosity` – задает минимальный уровень логов (quiet, minimal, normal, verbose, diagnostic)
- Примеры вызова скриптов:
  - *./build.sh --script=custom-build.cake --target="Integration" --verbosity=normal*
  - *.\build.ps1 -Target Deployment --nuGetApiKey=proget:xXxXXxXxXXxX*

ИНТЕГРАЦИЯ С BAMBOO – WIN



ИНТЕГРАЦИЯ С BAMBOO – MAC



ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ CAKE В НАШИХ ПРОЕКТАХ

THANK YOU

TAKE THE WORLD