Let’s automate!

Artūras Šmorgun, VilniusPHP#6

Me

Software Engineer

Inviqa Group

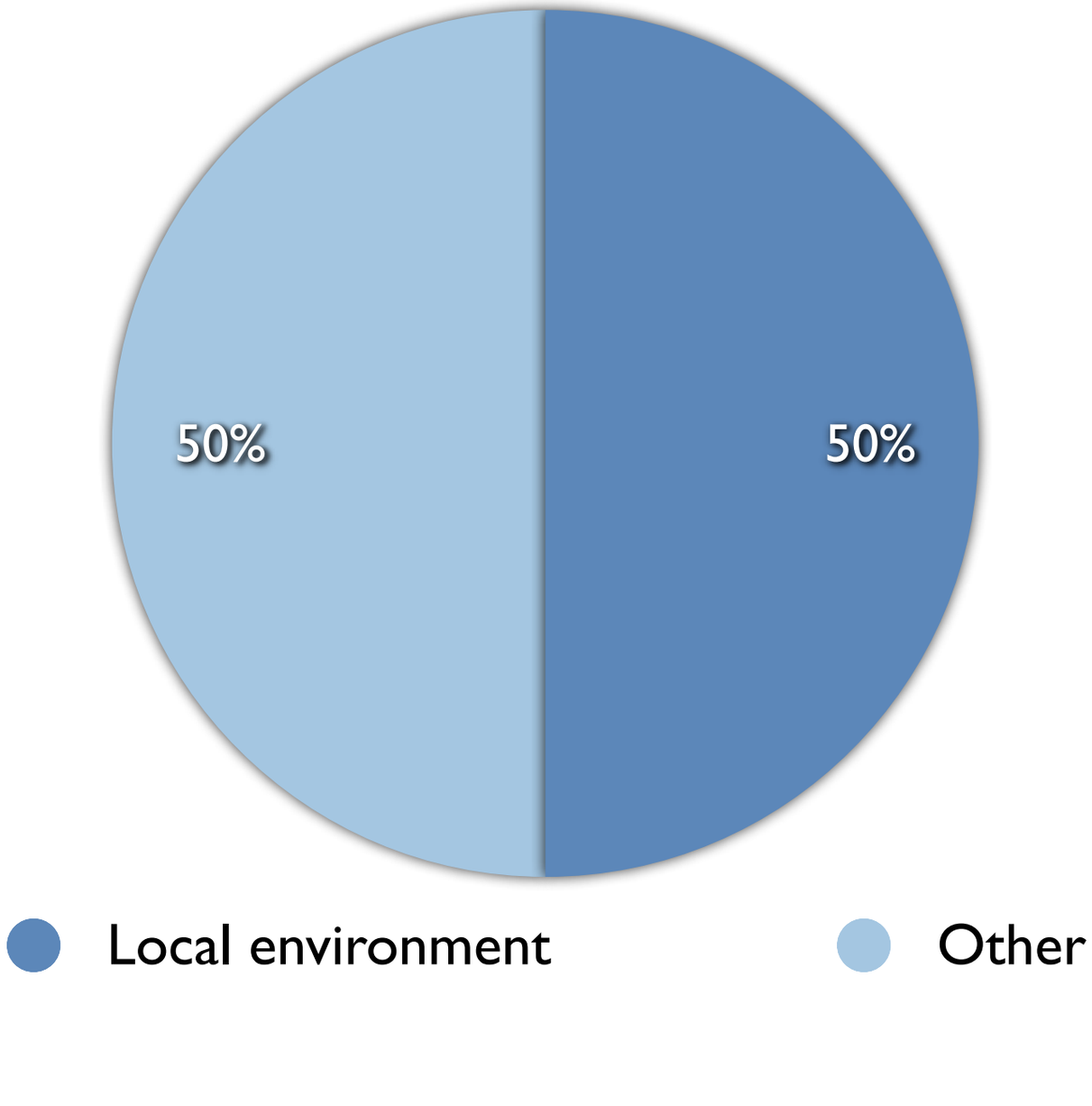
(Inviqa + Session Digital + SensioLabs UK)

Common environment

Git, Apache, PHP, MySQL...

Local

Used by 50% attendees



Advantages

Fast

Easy

Comfortable

Disadvantages

Not flexible

Very different from production (usually)

Team members' environments are different

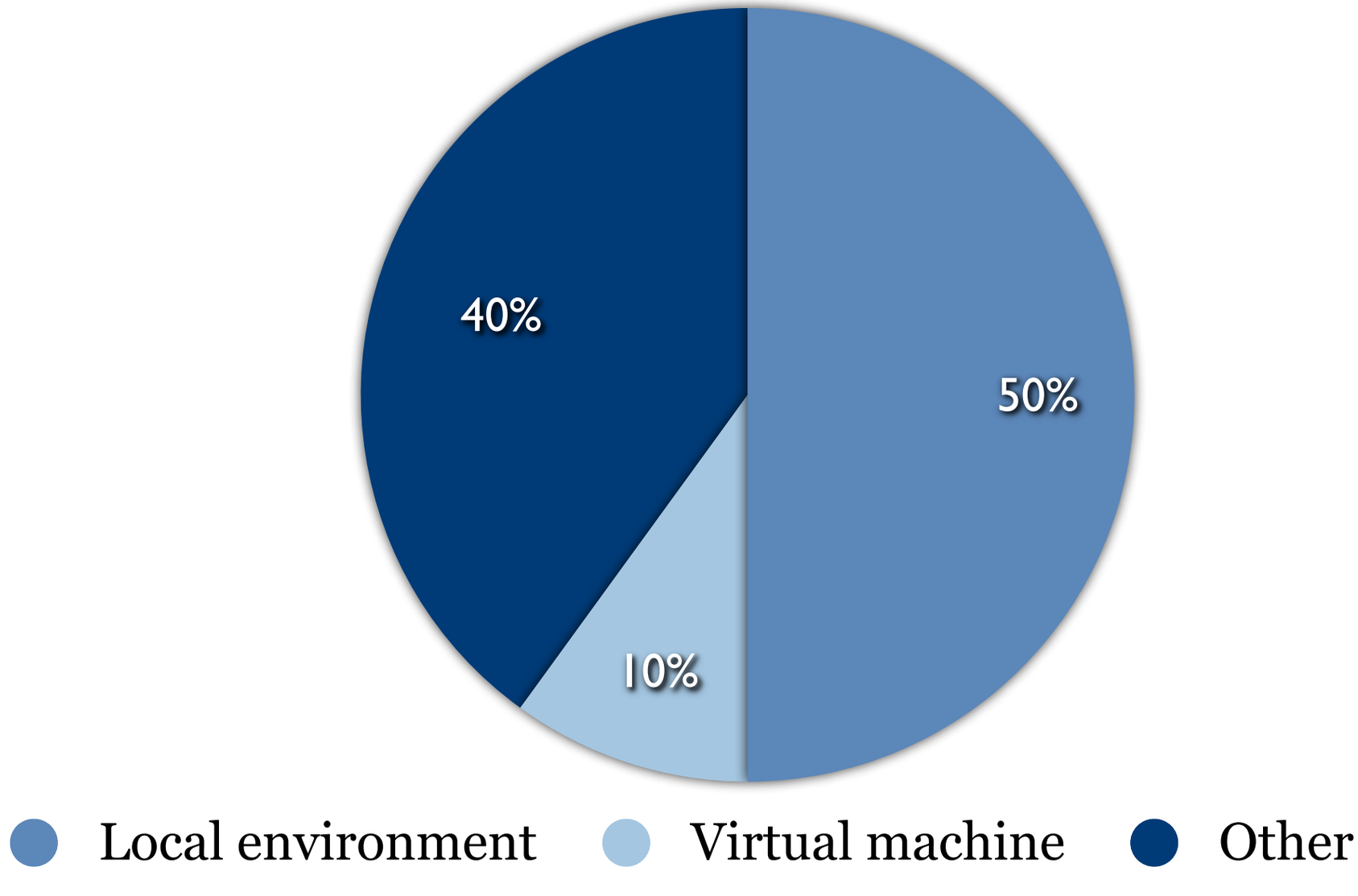
What have we got?

Fast, easy, comfortable environment.

What if it would be more similar to production?

“Golden” VM

Used by 10% attendees



Advantages

More similar to production (usually)

More flexible

Works same way on different OS

Team members use same environment

Virtualbox

Oracle product

Open source

Free

Works under Windows, Linux, Mac, Solaris

Can host Windows, Linux, Mac, Solaris, ...

Installation

Google: “virtualbox download”

Download

Install (next > next > next)

Start

Creating VM

New machine with new virtual disk

Install OS

Install Guest Additions

Configure shares

Install Apache, PHP, MySQL

Done :)

Disadvantages

More steps to reproduce

Uses more resources

Slower (sometimes)

Usually is copied between projects

Not easy to distribute

What have we got?

Fast, easy, comfortable environment,

similar to production and flexible.

What if it would be easier to distribute?

Vagrant

Software for creating and configuring virtual development environments.

What is it?

Console tool

Written in Ruby

With integrated VirtualBox integration

Vmware integration (via plugin)

Cloud integration (via plugins)

Advantages

Automated environment distribution

Ready to use templates in the internet

Environment setup is code

Easier to manage several projects

Easier to create complex environment

How it works?

It uses Baseboxes

to create Virtual Machines

via configuration in Vagrantfile

Installation

Google: “vagrant download”

Download

Install (next > next > next)

Done :)

Commands

```
vagrant init

vagrant up  && vagrant halt

vagrant suspend  && vagrant resume

vagrant restart

vagrant destroy
```

Configuration

`vagrant provision`

`--no-provision`

Shell provisioning

What have we got?

Fast, easy, comfortable environment,

similar to production and flexible.

Easy to distribute.

What if it would be easier to setup VM?

Chef

Configuration management tool.

Chef Solo

Version of Chef, perfect for managing single machine (per project, per environment).

What is it?

Console tool

Written in Ruby

Open source

by Opscode

Advantages

Configuration as a code

Easy to reproduce

Easy to store

Loads of ready to use recipies

How it works?

Uses Cookbooks & Recipes.

Vagrant starts chef provisioning.

Chef provisions defined recipes.

Recipes configures environment.

Installation (deb)

Add opscode repository

```
echo "deb http://apt.opscode.com/ `lsb_release -cs`-0.10
  main" | sudo tee /etc/apt/sources.list.d/opscode.list
```

Add GPG key

```
sudo mkdir -p /etc/apt/trusted.gpg.d
gpg --keyserver keys.gnupg.net --recv-keys 83EF826A
```

Install package

```
sudo apt-get update && sudo apt-get install chef
```

What have we got?

Fast, easy, comfortable environment,

similar to production and flexible.

Easy to distribute, with configuration

as a code.

What if it would be easier to manage cookbook dependencies?

Librarian

Framework for writing bundlers, which are tools that resolve, fetch, install, and isolate a project's dependencies.

Librarian Chef

Bundler for Chef-based infrastructure repositories.

Advantages

No need to maintain third party cookbooks

No need to know have all the dependencies

How it works?

It becomes source of public cookbooks

Custom cookbooks goes to “site-cookbooks”

Configuration in Cheffile

Lock in Cheffile.lock

Installation

Install RVM

```
\curl -L https://get.rvm.io | bash -s stable --ruby --
autolibs=enable --auto-dotfiles
```

Install package

```
sudo gem install librarian-chef
```

Commands

```
librarian-chef init

librarian-chef install [--clean] [--verbose]

librarian-chef show

librarian-chef outdated [--verbose]
```

What have we got?

Fast, easy, comfortable environment,

similar to production and flexible.

Easy to distribute, with configuration

as a code.

With cookbook dependency management.

What if I wouldn’t need to rely on unknown baseboxes?

Veewee

Tool for easily (and repeatedly) building virtual machine images.

Advantages

VM creation via code

No need to rely on baseboxes from the Internet

Custom baseboxes

How it works?

VM for KVM, Parallels, Virtualbox.

Using Templates

and Definitions.

Builds

and Validates.

can Export.

Installation

```
sudo gem install veewee
```

Commands

```
veewee vbox templates

veewee vbox define <boxname>

veewee vbox build <boxname>

veewee vbox verify <boxname>

veewee vbox export <boxname>
```

Disadvantages

Too complicates at times

Now always what you need

Bento

What is it?

Opscode product

Only for VirtualBox

Advantages

DRY modular baseboxes

How it works?

Encapsulates Veewee

Has several DRY Definitions prepared for use

Installation

Clone bento from github

```
git clone git://github.com/opscode/bento.git
```

Install package

```
cd bento && bundle install
```

Commands

```
bundle exec veewee vbox list

bundle exec veewee vbox build <def-name>

bundle exec veewee vbox build <def-name>

bundle exec veewee vbox export <def-name>
```

What have we got?

Fast, easy, comfortable environment,

similar to production and flexible.

Easy to distribute, with configuration as a code.

With cookbook dependency management,

custom built baseboxes.

What if deployment would be more reliable too?

Chef Server

A systems and cloud infrastructure automation framework, no matter the size of infrastructure.

Organization

One or more dev workstation

Single chef server

Many nodes

Workstation

Communicates with Chef Server

Knife

Recipe synchronization

Organization policy configuration

Shef Server

Hub for configuration data

Cookbooks

Policies

Registered nodes metadata

Node (Chef Client)

Maintained server

Chef-client for configuring

Ohai for system information

Capistrano

Utility and framework for executing commands in parallel on multiple remote machines, via SSH.

What is it?

Console tool

Open Source

Free

Written in Ruby

Advantages

Instant deployment to all machines

Versioning of deployments

Fast rollback

How it works?

Capfile - definition

Tasks - executes on servers

Servers - 1..N

Roles - may be unique per server

Installation

```
sudo gem install capistrano
```

Commands

```
capify
```

```
cap <taskname>
```

```
cap -vT
```

```
cap invoke COMMAND=”echo 'command'” SUDO=1
```

```
cap shell
```

What have we got?

Fast, easy, comfortable environment,

similar to production and flexible.

Easy to distribute, with configuration as a code.

With cookbook dependency management,

custom built baseboxes.

Automated deployment of infrastructure and code.

Links

Virtualbox

Vagrant

Veewee / Bento

Chef

Librarian

Capistrano

Questions?

Artūras Šmorgun

asmorgun@inviqa.com

@asarturas

Thank you!

Artūras Šmorgun

asmorgun@inviqa.com

@asarturas