实时数据应用场景中

人· 流程· 组织的变化

程显峰

为什么需要实时数据？

数据时效性

半衰期

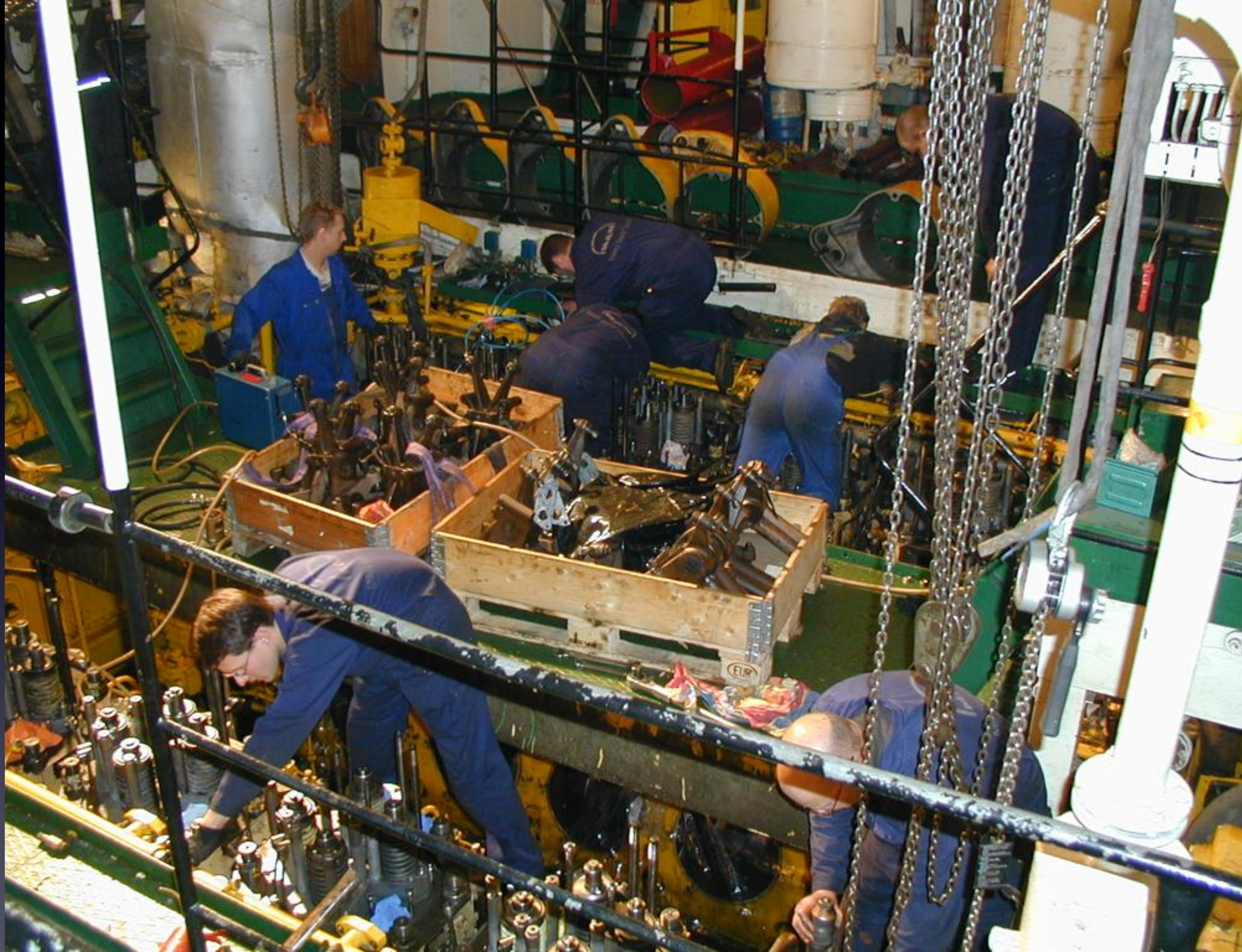
理想的实时数据系统

低延迟

高可用

马力足

但现实很残忍



如何全面评价实时系统

自动化

维护成本低

没有废弃的系统

如何细化到指标呢

时间

- 容易理解
- 容易测量
- 没有歧义

时间都浪费在哪里呢

```
def AddFavorite(u_id, i_id):
    db.insert('ImageFavorite', user_id =
        u_id, img_id = i_id)
    #同时更新image里的'favo_num'字段
    db.query("UPDATE image set favo_num=
        favo_num+1 WHERE id=$id", vars=dict
        (id=i_id))

#取消喜欢
def CancelFavorite(u_id, i_id):
    db.delete('ImageFavorite', vars = dict(
        img_id=i_id, user_id = u_id), where
        = 'img_id = $img_id and user_id =
        $user_id')
    #同时删除image里的'favo_num'字段
    db.query("UPDATE image set favo_num=
        favo_num-1 WHERE id=$id", vars=dict
        (id=i_id))

#是否喜欢当前图片
def IsFavorite(u_id, i_id):
    user_id = u_id
    img_id = i_id
    return db.select(
        'ImageFavorite',
        vars = dict(img_id=img_id, user_id=
            user_id),
        what = 'count(id) as c',
        where = 'img_id = $img_id and
            user_id = $user_id')[0].c

#得到用户喜欢的图片id
def GetFavImageByUserId(u_id):
    return db.select('ImageFavorite', vars=
        dict(user_id=u_id), where='user_id
        = $user_id')

#图片被多少用户喜欢
def GetFavImageByImageId(img_id):
    return db.select('ImageFavorite', vars=
        dict(img_id=img_id), order='id
        DESC', where='img_id = $img_id')

#用户喜欢了多少图片
def GetUserFavCount(user_id):
    return int(db.select('ImageFavorite',
```

```
#!/usr/bin/env python
# coding: utf-8

import web
import config
import md5
import os
import Image
import time
import datetime
import cgi
import random
import hashlib

from web import form

from app.models import applicants
from app.models import users

from app.helpers import session, utils
from app.helpers import email_templates
from config import view, site_name,
    encryption_key


siteName = site_name

user = session.get_session()

# password_form = form.Form(
#     form.Password('password',
#         form.notnull,
#         form.Validator('至少6个字符',
#         lambda x: users.is_valid_password
(x)),
#         description='你的新密码:'),
#     form.Button('submit', type='submit',
value='Change password')
# )

# nickname_form = form.Form(
#     form.Textbox('nickname',
#         form.notnull,
#         description='你的新昵称:'),
#     form.Button('submit', type='submit',
value='Change your nickname')
```

```
$def with(img, page
is_favorite, comment
$var page_title: $im
<div class="span7">
    $if pager.left:
        <a href="/ph
        #8249; 上一张
    $(pager.number +
    $if pager.right:
        <a href="/ph
        下一张 &#8250
    <br>

    <img src="$img.p
    " style="margin-

    <div id="imgInfo
        <h4 id="imag
        h4>
        $if img.imag
            <p id="i
            imageDes
        $else:
            $if img.
                <p
                点击
    </div>
    <div id="imgOthe
        <span>上传于
        |
        <span>$img.c
        <span>$img.v
        <span>
            $if img.
                <a h
                >
                </a>
            $else:
                <i i
                人喜
        </span> |
        $if img.user
            <a href=
            del">删除
```

主要非计划时间

- 处理上线部署
- 调试bug
- 沟通协调

为什么进度计划失效


BRITISH AIRWAYS

有本书专门讨论

- 系统管理员的时间管理
- TODO无用论
- 消除“我忘记了”
- LISA大会

DON'T

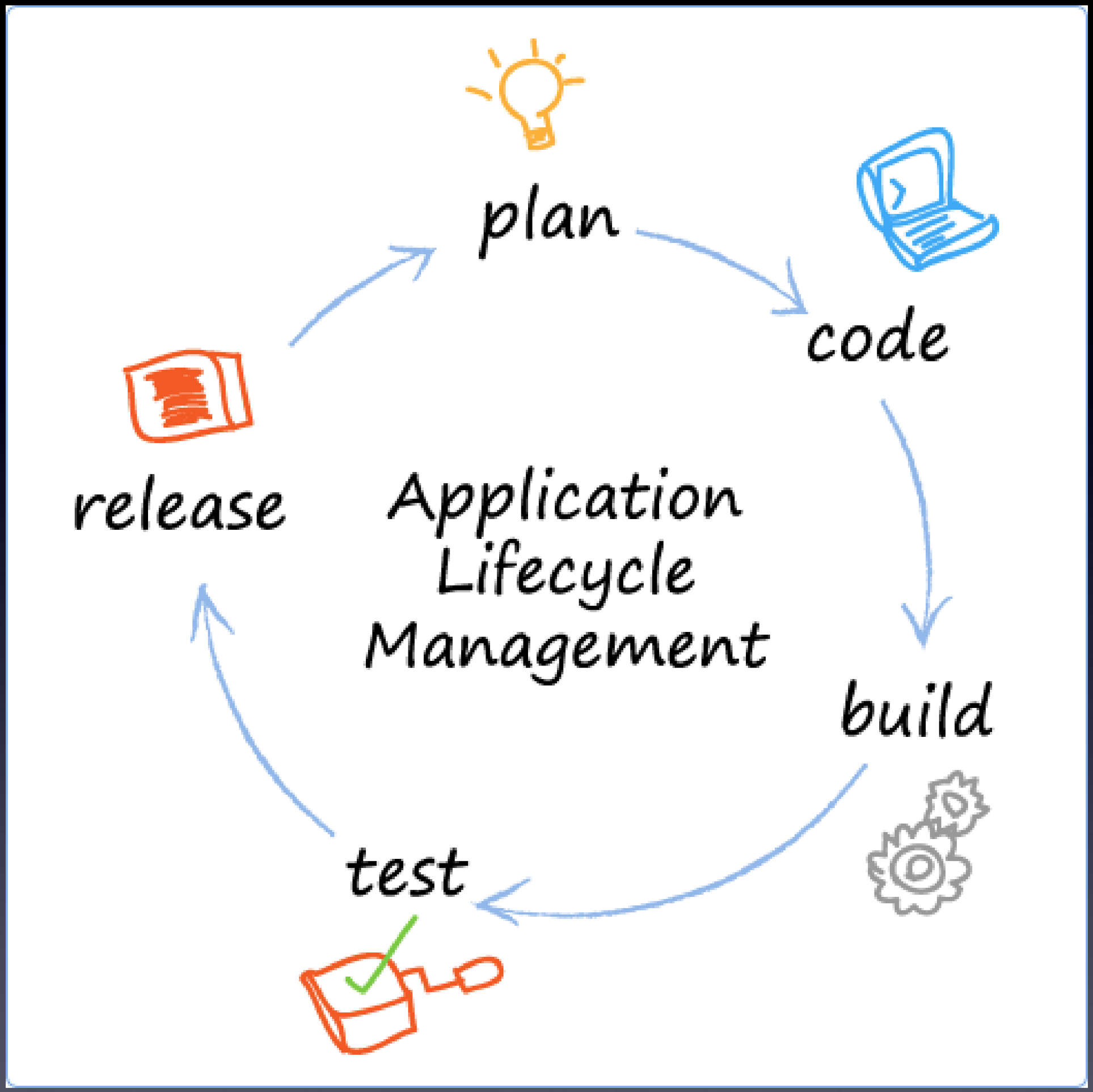
FORGET

关键衡量指标

- 回滚
- 定位bug
- 隔离

软件的生命周期

- 设计到交付
- 设计到交付只是一点点
- 只有运行起来的软件才有价值


plan
code
build
test
release
Application
Lifecycle
Management

只有运行起来的软件才有价值

技术能带来哪些改进

- 监控和性能分析
- 自动化
- 基础设施
- 调试困难
- DevOps


max: 230V, 3

技术评估方式转变

时间算个新的维度

节约时间

技术所带来负面影响

- move fast and break things
- 抛弃很多习惯

启动时间

裸机	虚拟机	容器
小时	分钟	秒

本性难移

习惯的力量

- 我们决策大部分不是依赖于思考而是习惯

个人的习惯

- 代码风格
- 上线操作
- 测试覆盖
- 调试手法
- ……

组织的习惯

习惯的适应场景

- 节省成本么？
- 好管理么？
- 办公室政治？

组织

- 高效
- 快速反应
- 自我反省

组织

- 泯灭个性的
- 有基础的协作规范的
- 强调细节上的一致

转型

各种不适应


配合

练习

需要多大的代价


LinkedIn


SWISS BANK CORPORATION
1 KILO
999.9
S.B.S

Erlang法则

- Erlang不光是一种编程语言
- 解决问题越快，并发数要求就越少

减少解决单个问题的时间

- 减少等待
- 人不是机器，不能高并发

案例

- Heroku
- 你的线上系统有版本控制么？

feature branches

develop

release branches

hotfixes

master

Time

Feature for future release

Major feature for next release

Severe bug fixed for production: hotfix **0.2**

Tag **0.1**

Incorporate bugfix in **develop**

Tag **0.2**

From this point on, "next release" means the release

Start of release branch for **1.0**

案例

- github上的机器人hubot
- openshift的合并机器人

HUBOT

(note: it's pronounced hew-bot)

A CUSTOMIZABLE, KEGERATOR-POWERED LIFE EMBETTERMENT ROBOT

COMMISSIONED BY **GITHUB**

1. GIVEN NAME OF INVENTION
PLEASE INCLUDE ANY PRONUNCIATION NUANCES

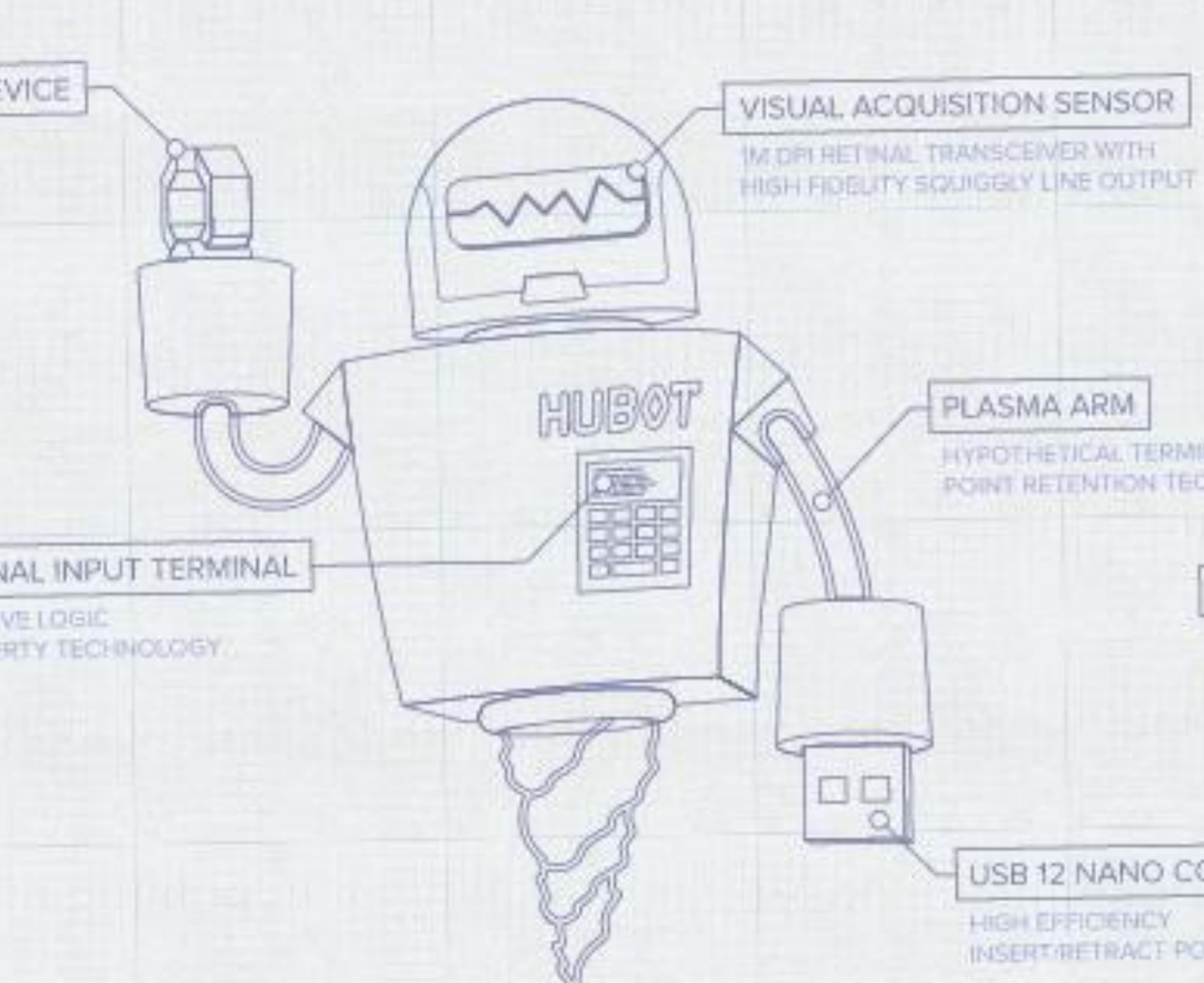
2. DESCRIPTION OF INVENTION
PLEASE BE AS CLEAR AND CONCISE AS POSSIBLE

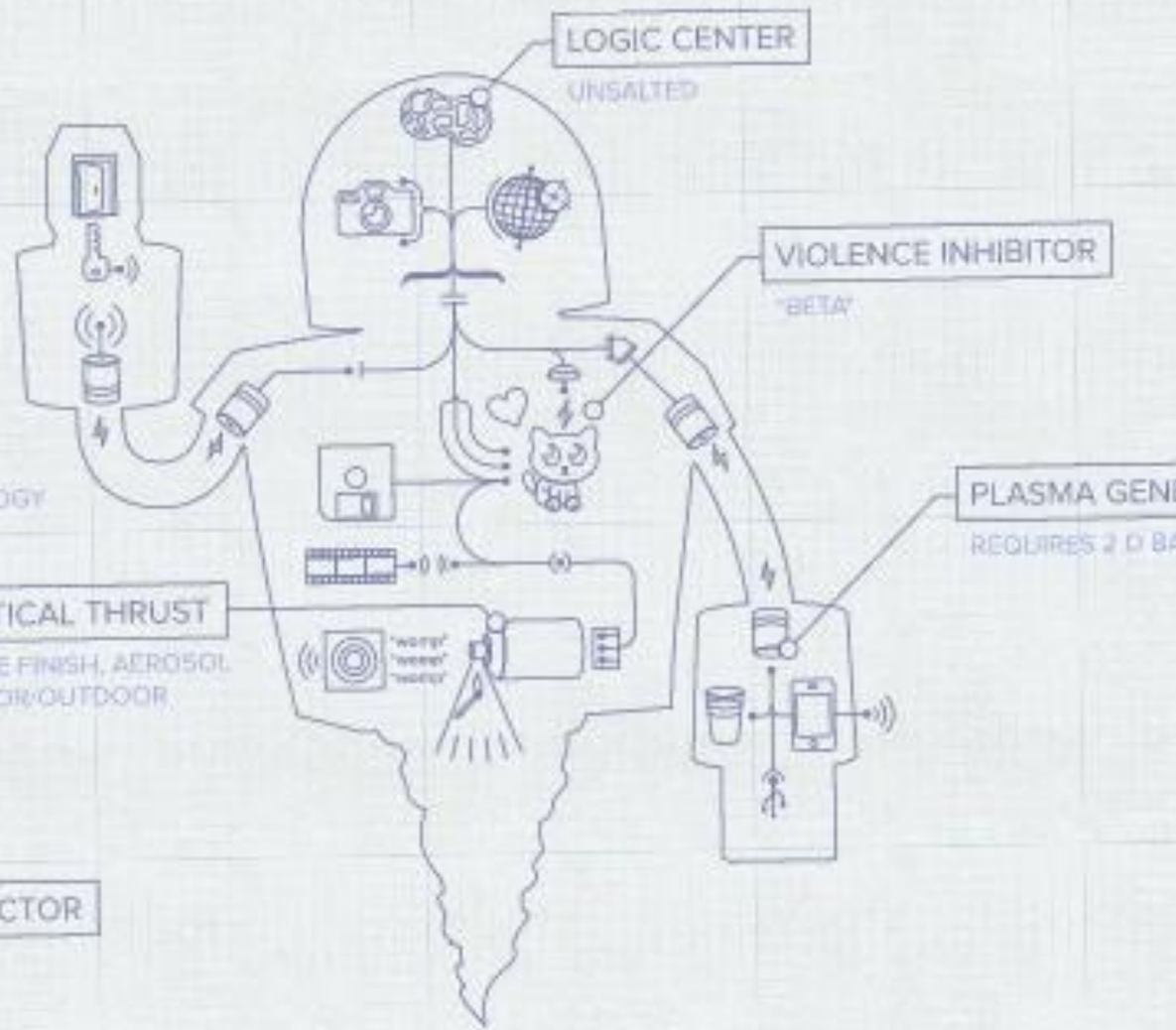
4. AFFILIATED COMPANY
(YCLUDE AFFILIATED COMPANY LOGO (IF APPLICABLE)

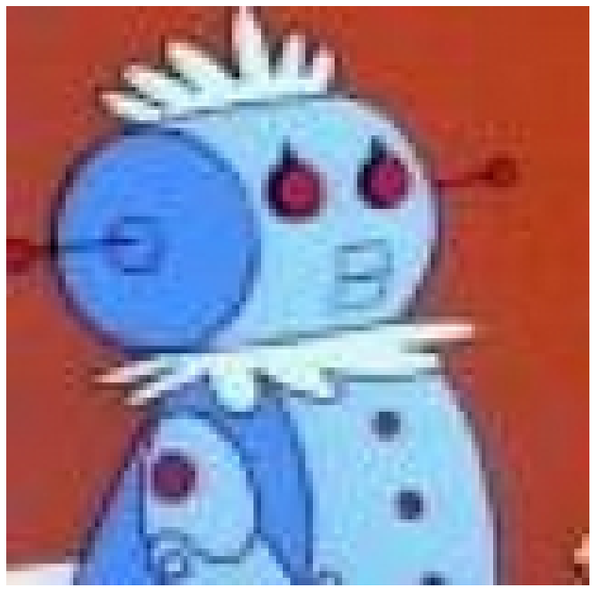
5. SIGNATURE OF INVENTOR
PLEASE DO NOT ATTEMPT TO SIGN WHILE INEBRIATED

TO **VIEW HUBOT'S SOURCE CODE**
T HTTP://GITHUB.COM/GITHUB/HUBOT/.)

☐ ~~**CREATE MY HUBOT**~~ USE **`HUBOT --CREATE`** TO CREATE HUBO
(UPGRADE TO NEW VERSIONS VIA PACKAGE.JSON)







OpenShift Bot

openshift-bot

http://github.com/openshift

Joined on May 25, 2012

0 followers **0** starred **0** following

Contributions | Repositories | Public Activity | Follow

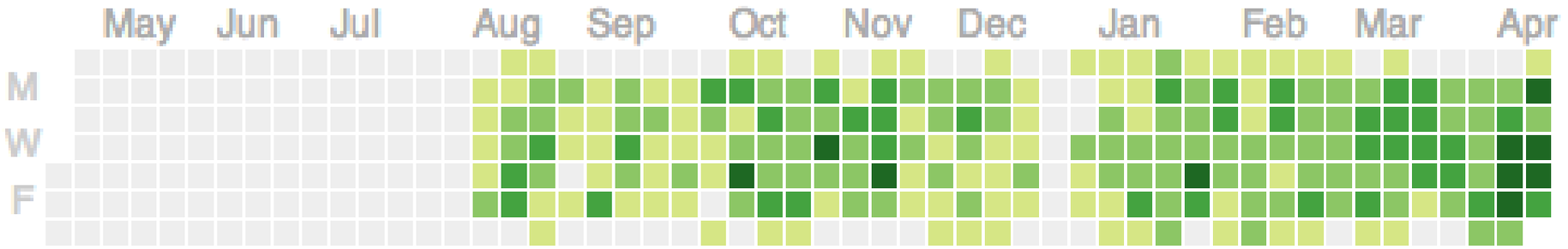
Popular repositories

openshift-bot doesn't have any repositories you can view.

Repositories contributed to

- **openshift/origin-server** 338 ★
  Public open source repositor…
- **openshift/rhc** 138 ★
  Public open source repositor…
- **openshift/origin-dev-tools** 9 ★
  OpenShift Development Tools
- **openshift/openshift-java-client** 40 ★
  java client for the OpenShift …

openshift-bot's Open Source Contributions



Summary of Pull Requests, issues opened and commits. Learn more.

Less More

1,384 Total	34 days	12 days
Apr 26 2012 - Apr 26 2013	February 03 - March 08	April 15 - April 26
Year of Contributions	**Longest Streak**	**Current Streak**

Contribution Activity

Period: 1 Week

案例

- docker
- virtualbox vagrant puppet


ORACLE
VirtualBox


VAGRANT


puppet
labs

案例

- Ruby 2.0
- Dtrace or SystemTap
- Joyent
- 在线调试怎么玩？

Q & A

@程显峰-Mars