

Popiš, testuj, programuj

Honza Javorek

Běžný způsob vývoje
školního projektu



Dostanu zadání



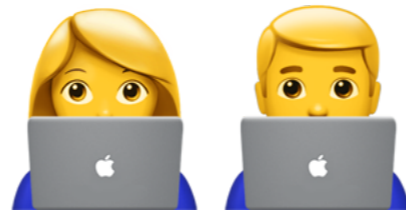
Jdu do parku



Jdu na pivo



Hraju deskovky



"Tak co, jak jsi na tom?"



"Taky nic..."



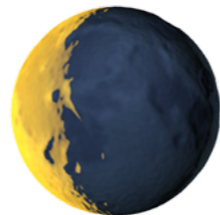
Jdu na pivo...



2 dny před termínem



Pořádně čtu zadání



2 dny = 1 dlouhý den



Otevřu soubor, bouchám kód, zkouším
spouštět a koukám, co program dělá

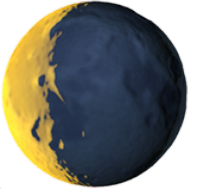


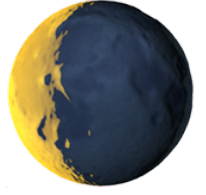
Když se zdá, že to jede,
odevzdávám



Když se zdá, že to nejede, ladím...
23:59 odevzdávám cokoliv

Několik problémů







Dostanu zadání... čtu zadání

Otevřu soubor, bouchám kód, zkouším
spouštět a koukám, co program dělá

Když se **zdá**, že to jede/nejede...

Nejasné zadání

= rozdílná očekávání

= nedomyšlené hraniční situace

= ztracené body, ztracené iluze

= hranolky s tatarkou nebo kečupem?

Dokumentace Testy

Nepotřebujeme?

Běžný způsob vývoje
projektu v praxi



Nikdy nedostanu jasné zadání

Otevřu soubor, bouchám kód, zkouším
spouštět a koukám, co program dělá

Když se **zdá**, že to jede/nejede...

Nejasné zadání

= rozdílná očekávání

= nedomyšlené hraniční situace

= smutní nebo rozzlobení lidé

= rentgeny, chirurgické zákroky

Žádná dokumentace

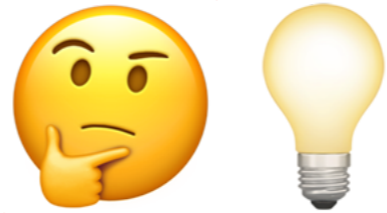
= nikdo to už nikdy nebude umět použít

Žádné testy

- = nikdo už nikdy nezjistí, jak se to má chovat
- = při úpravách kusu kódu se rozbije zbytek
- = opravená chyba se objeví znovu



Týmová práce...



Jde to jinak?

TDD

Test Driven Development

- 1 jako první napíšu test
- 2 spustím ho, je červený
- 3 snažím se do něj trefit implementací
- 4 spustím ho, je zelený



Automatizuje

"zkouším spouštět a koukám, co program dělá"



Dokumentace chování



Musím vidět červený test!

Vím, že jsem opravdu něco změnil, opravil



Musím psát test podle zadání!

Ne podle implementace (proto taky psaní předem)



Mám chybu? Napíšu červený test

Jasně identifikuju chybu, pak opravím, zabráním regresi



Má někdo jiný chybu? Napíšu červený test

Největší pomoc!



Nikdy nedostanu jasné zadání
Neexistuje dokumentace



Unit testy, integrační testy

BDD

Behavior Driven Development

Gherkin / Cucumber

Feature: Beer glass

Background:

Given you expect the glass to contain "0.5" l of beer

Scenario: Empty glass

When the glass contains "0" l of beer

Then the glass owner will shout "ještě jedno!"

And the glass will be scheduled to be filled

Scenario: Full glass

When the glass contains "0.5" l of beer

Then the glass owner will NOT shout "ještě jedno!"

And the glass will be left intact

- 1 zadavatel napíše textové scénáře
- 2 nad scénáři se pobavíme, doladíme
- 3 test píšou pro jednotlivé řádky
- 4 pustím přes Cucumber



Gherkin, Cucumber



Jasně zadání



Testovatelná dokumentace



Scénáře může číst / psát kdokoliv



Scénáře mohou být testovány v
jakémkoliv jazyce



Já sám jsem zadavatel → kanón na vrabce
Neexistuje přehledná dokumentace



Spíš integrační testy
Zadavatel je někdo jiný

RDD

RDD

Readme Driven Development

<http://tom.preston-werner.com/2010/08/23/readme-driven-development.html>

README.md

Requests

The `requests` library allows you to perform HTTP requests from your Python code.

Example

```
```python
>>> r = requests.get('https://google.com')
>>> r.status_code
200
```
```

License

MIT



Markdown



Šance nejdřív přemýšlet



Dokumentace je hotová



Tým může věc použít dřív,
než existuje



Lze se o věci s kýmkoliv
snadno pobavit



Naučím se formulovat myšlenky a
komunikovat s ostatními lidmi

README **se nesmí** úpravami
rozejít s tím, jak funguje kód

Je to jádro toho, co **slibujeme**
uživateli, co od programu **očekává**



Nešlo by to nějak zajistit?

doctest

```
python -m doctest README.md
```



Testovatelná dokumentace!



Mám základní testy i dokumentaci předem
a není to ani opruz!

Díky tomu, že návrh testuji, můžu navrhovat
před tím, než programuji

Co můžu použít hned?



Pište testy! Vždy! A jako první!



Chtějte školní projekty zadávány
v podobě Gherkin scénářů



Pro své hobby projekty
si nejdříve napište README

Co v Apiary?



Silná kultura automatického
testování a psaní dokumentace



Pull Request bez testů
a bez odpovídající dokumentace...



Plánujeme velký přechod
na Gherkin/Cucumber



Cucumber + Selenium = chimp.readme.io



Apiary je jedno velké RDD,
akorát pro HTTP API

README = API Blueprint
doctest = Dredd

Demo