

Joelův test

12 kroků k lepšímu programování

Jaroslav Šnajdr





i

Co je Joelův test?

Co je to?

- 12 otázek o vašem vývojovém týmu
- Každá odpověď ano = 1 bod

Jaký je výsledek?

- Plných 12 bodů: dobře organizovaný, produktivní tým
- Méně než 10 bodů: vážné problémy

K čemu je to dobré?

- Zorganizujte si svůj projekt nebo firmu
- Otestujte si potenciálního zaměstnavatele

Co potom?

- Každý bod je vstupní branou do složitějších metodik

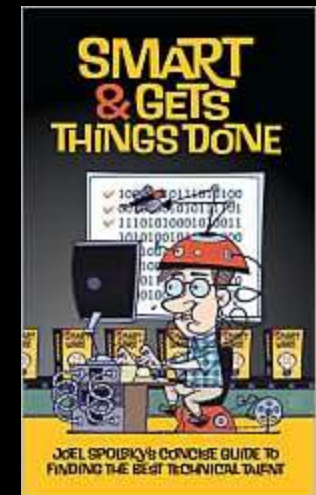
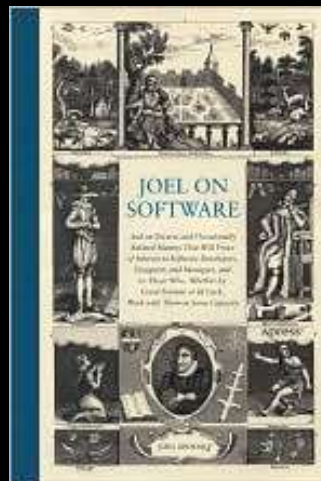


ii

Kdo je
Joel Spolsky?

Joel on Software: www.joelonsoftware.com

- Publicista, blogger, programátor, manažer, podnikateľ
- Zakladateľ Fog Creek Software v New Yorku
- Autor 4 kníh



A vertical image on the left side of the slide showing the dark, intricate silhouette of bare tree branches against a warm, orange-hued sky with soft clouds, suggesting a sunset or sunrise.

1

Používáte
systém pro
správu verzí?

Kam ukládáte zdrojové kódy?

Co umí Version Control System?

- Check-out
- Check-in
- Tag
- Branch
- Merge

Proč ho potřebujete?

- Pomáhá při spolupráci mezi vývojáři
- Udržuje historii změn a jejich autorech
- Branch & Merge: paralelní vývoj, experimenty
- Bezbolestné zálohování

Nejznámější produkty

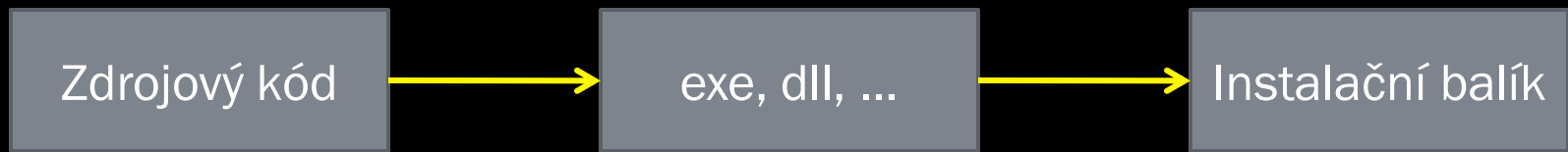
- CVS
- Subversion

Software Configuration Management



2

Dokážete
udělat build v
jednom kroku?



Vše automaticky, v jednom kroku

- Všechny verze, jazykové mutace, platformy

Proč?

- Náročný proces náchylný na chyby
- Nové verze často vydáváme ve stresu

Možné problémy

- Některé komponenty jsou obtížně skriptovatelné - installery
- Inkrementální build po komponentách



3

Děláte denní
buildy?

Denní build je:

- Automatický
- Pravidelný
- Kompletní

Proč?

- Vývojáři commitují chyby, které znemožňují ostatním pracovat
- Chyby v denním buildu mají nejvyšší prioritu
- Pravidelné zásobování QA, zákazníků a ostatních zájemců
- Archiv starších verzí pro dohledávání chyb

Continuous integration

- Agilní a extrémní metodiky



4

Máte databázi
chyb?

V paměti udržíme nejvýše 3 poslední chyby

- Nezbytné i pro one-man team

Každý bug musí mít:

- Kroky k reprodukci
- Očekávané chování
- Pozorované chování
- Komu je chyba přiřazena
- Opraveno: ano/ne

Další užitečná pole:

- Závažnost
- Priorita
- Výsledek řešení

Bugzilla

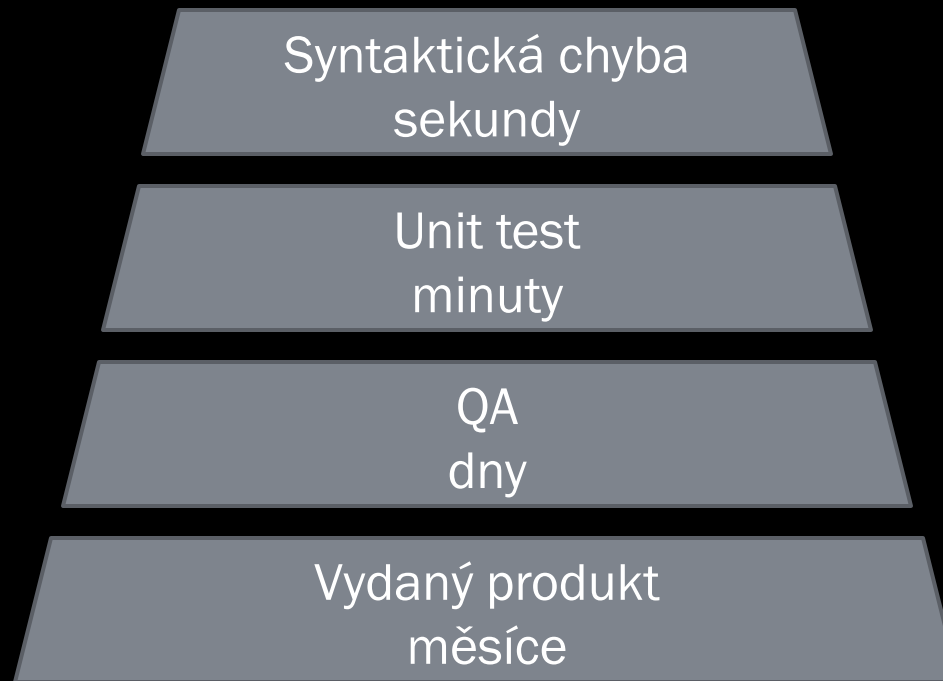




5

Opravujete
chyby předtím,
než začnete
psát nový kód?

Náklady na opravu chyby rostou s časem:



Časové odhady na opravu chyb jsou těžké

- Nové vlastnosti + oprava chyb = nespolehlivý plán

Řízení kvality ve výrobě

- Jidoka (autonomation): zastavení linky při výskytu problému
- Andon: signální světlo

6

Máte
aktualizovaný
plán prací?

32	
30	Kuala Lumpur
23	Singapore
Cancelled	
48	Taipei
44	Taipei
24	Xiamen
7	Shanghai/Pudong
2	Laoag
Cancelled	
27	Bangkok
17	Chengdu
21	Shanghai/Pudong
71	Shanghai/Pudong
20	Sydney
18	Melbourne
64	Guilin
35	Cairns
	Changsha
	Auckland

Nikomu se nechce dělat projektový plán

- Náročná mravenčí práce, vyžaduje přemýšlení
- Plány málokdy vycházejí

Tři funkce projektového plánu

1. Závazek vůči ostatním
 - Obchodní oddělení
 - Zákazník
2. Konkrétní úkoly, konkrétní lidé
 - Úkoly v rozsahu hodin: přemýšlení, závislosti
 - Omezení rozsahu
3. Sledování postupu
 - Priority, rizika
 - Plán je živý dokument

Původní odhad	Současný odhad	Delta
8 hodin	12 hodin	+4

A close-up photograph of an architectural drawing on a light blue background. A silver ruler is placed diagonally across the top of the drawing, showing measurements in centimeters. A pair of compasses is positioned in the lower center, drawing a circle on the drawing. The drawing itself features various lines, including solid, dashed, and hatched areas, representing a technical plan or section. Numerical values like 2.85, 1.64, 2.67, 1.08, 80.1, 80, 1.83, and 9 are visible, indicating dimensions. The overall scene suggests a professional architectural or engineering workspace.

7

Píšete
specifikace?

Specifikace pomáhá implementovat správný produkt

- Nutná pro diskusi, revizi, změny a zpětnou vazbu
- Změna ve specifikaci je snadná, ve funkčním kódu skoro nemožná
- Nezbytná pro kvalitní návrh
- Umožňuje komunikaci 1:N

Funkční specifikace

- Uživatelský pohled
- Okna, stránky, dialogy, volby, stavové diagramy

Technická specifikace

- Návod pro implementaci
- Algoritmy, datové struktury, databázové modely, jazyky a nástroje



8

Pracují vývojáři
v klidném
prostředí?

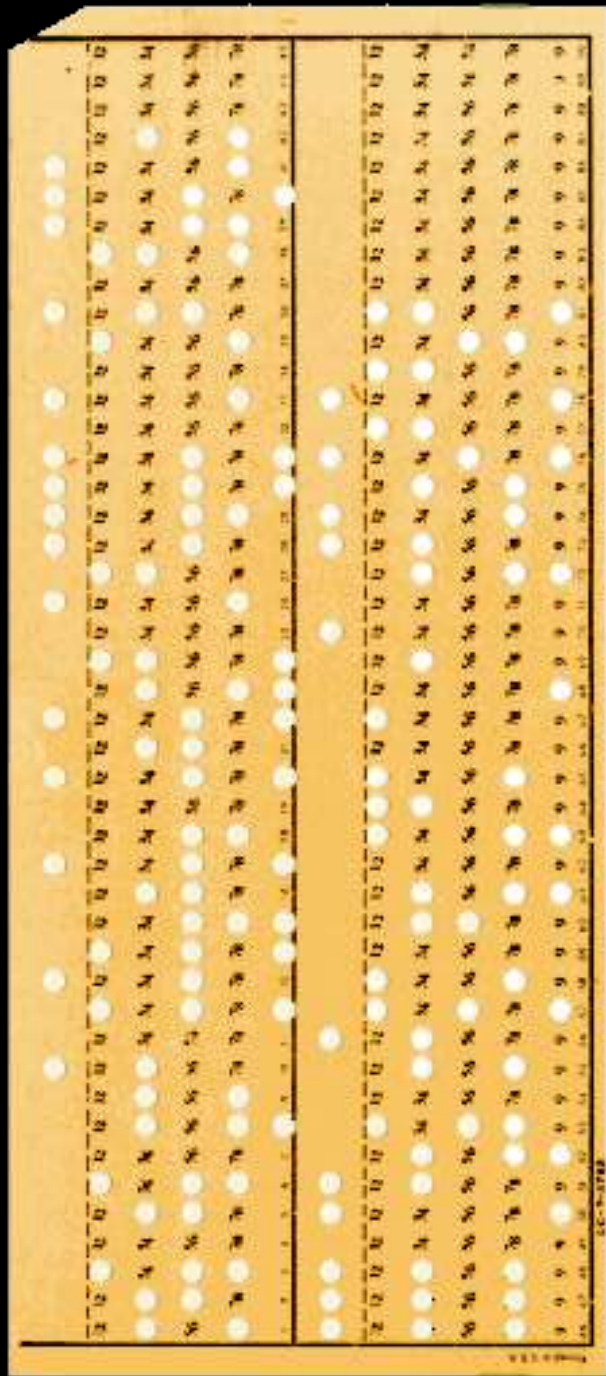
Programování je znalostní, tvůrčí práce

Tom DeMarco, Timothy Lister: Peopleware

- Režim největší produktivity: tzv. „zone“ neboli „flow“
- Dostat se do „zóny“ trvá přibližně 15 minut
- Častá vyrušení znemožňují dostat se do „zóny“
- Při programování je důležitá krátkodobá paměť
- Doporučeny oddělené kanceláře – 1 až 3 lidé

Proč je to často problém?

- Samostatná práce vs. komunikace v týmu
- Open space je levnější



9

Používáte
nejlepší
dostupné
nástroje?

Co všechno nám pomáhá při práci:

- Rychlý počítač = rychlý překlad, rychlé testy
- Více počítačů: Windows, Linux, Mac OS X
- Velké LCD = více zdrojového kódu, více oken v debuggeru
- Softwarové nástroje: profilery, statická analýza
- Testovací nástroje: počítače, sítě, Vmware
- Velké disky
- Odborná literatura: firemní knihovna



10

Máte testery?

Kvalitní software vyžaduje dedikované testery

Můžou testovat programátoři?

- Programátor nevidí vlastní chyby
- Testuje svůj program stále stejným způsobem

Testování na zákaznících nekončí dobře

Samostatné a nezávislé QA oddělení

- 1 až 2 testeři na jednoho programátora
- Problém: dobří testeři jsou vzácní a nechtějí být testery
- Automatizace testů
- Komplexní zátěžové testy

11

Píší uchazeči při přijímacím pohovoru kód?

```
#pragma once
#ifdef _MSC_VER > 1000
#ifndef __AFXWIN_H__
#error include 'stdafx.h' before including this file
#endif
#include "resource.h" // main symbols
// CDMotionApp:
// See DMotion.cpp for the implementation of the class
class CDMotionApp : public CWinApp
{
public:
    CDMotionApp();
// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CDMotionApp)
public:
    virtual BOOL InitInstance();
//}}AFX_VIRTUAL
// Implementation
//{{AFX_MSG(CDMotionApp)
afx_msg void OnAppAbout();
// NOTE - the ClassWizard will add and remove
// DO NOT EDIT what you see in these
MSG
//}}AFX_MSG
};
```

Přijímání nových pracovníků je nejdůležitější rozhodnutí

Častá, ale nesprávná kritéria:

- Zajímavý životopis
- Testy a, b, c na znalosti API
- Příjemný rozhovor

Důležité je, aby umět to co bude dělat: programovat

- Krátké příklady na kódování
- Jednoduchý OO design

Joelova kritéria: Smart & Gets Things Done

- Smart: umí myslet
- Gets Things Done: dosahuje praktických výsledků

12

Testujete
uživatelské
rozhraní na
lidech?



Test použitelnosti: testování rozhraní na lidech

- Požádejte kolegu nebo kamaráda, aby zkusil používat vaše GUI
- Prvních 5 uživatelů najde 95% problémů

I když nemáte zkušenosti s návrhem GUI, test použitelnosti vám pomůže

Pro UI design je důležitá empatie s uživatelem

- Nenuťte ho přemýšlet
- Zjednodušujte
- Nesnažte se mít všechno konfigurovatelné
- Nejlepší technologický design je pro uživatele neviditelný

Otázky?

