Metódy inžinierskej práce

Prednáška 8:

Kvalita kódu a refaktoring

Jakub Šimko
jakub.simko@stuba.sk

STU FIIT
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Zo spätnej väzby (k Esejám)

Kde nájdeme slovenskú šablónu na LaTeX?

Doplnil som na web

Draft už má byť v LaTeX-u alebo stačí vo Worde?

zo zadania nepriamo vyplýva LaTeX

hlavne však: robiť to vo Worde nemá zmysel

Čo má obsahovať priebežné odovzdanie eseje?

Opakujem: Nadpis + výroková osnova + zdroje

V priebeznom odovzdavani mame uviest zdroje a strukturu eseje. Dalo by sa ich nasledovne menit (pridavat/odstranovat)?

Áno

bit.ly/mip-dotaznik

Zo spätnej väzby (k Esejám)

Čo je to výroková osnova?

myšlienková štruktúra diela,
obsahujúca štruktúru nadpisov (sekcie, podsekcie)
a ďalej výrokov (viet),
dôležité myšlienky diela
zhruba zodpovedajú budúcim odstavcom textu

výrok = dôležitá myšlienka = základ odstavca

Ak som robil prezentáciu formou tvrdenie - dôkaz, môžu to byť tie tvrdenia?

Výborná otázka, lebo chápete princíp

Odpoveď je však: **skôr nie**

Dôvod: esej už bude posunutá

bit.ly/mip-dotaznik

Zo spätnej väzby (k Esejám)

Na stredných školách sme sa s takymito esejami nestretli, väčšinou sa vyžadovala len úvaha.

esej = úvaha na odbornú tému

subjektívne, hodnotiace zamyslenie sa,

opierajúce sa o objektívne skutočnosti

V akej osobe písať?

Autorský plurál

Diakritika v nadpisoch

`\v{z}` **= ž** `\'{a}` **= á**

bit.ly/mip-dotaznik

Zo spätnej väzby (k Esejám)

Na čo sa (v eseji) zamerať?

Kvalifikovane vyjadriť názor a podporiť ho argumentami

Akú by mala mať štruktúru?

Akú potrebujete...

má byť podriadená štruktúre argumentov

Neposkytli by ste príklad eseje?

Vždy keď počujem túto otázku, tak mám obavy

Články zadané ako témy sú tiež esejami (väčšina)

Kniha: *O softvéri od A po Z* (M. Bieliková, L. Litvová, 2009)
(zbierka esejí študentov FIIT)

bit.ly/mip-dotaznik

Nadpis: Techniky riešenia kríz v SW projektoch

Abstrakt (...)

Prečo sú softvérové projekty neúspešné

- *Väčšina softvérových projektov nekončí úspešne.* (...)
- *Aj s dobrým plánom je správne riadenie projektu kritické.* (...)
- *Zlyhania prichádzajú v dôsledku neskorého rozpoznania problémov a nekompetentného riešenia kríz.* (...)

Riešiť krízy analyticky alebo empiricky?

- *Kto si lepšie postaví záhradný plot: murár alebo architekt?* (analógia ...)
- *V ideálnom svete by sme mali expertný systém, ktorý nás v projekte bude kontrolovať.* (ako by mohol vyzerať...)
- *Vybudovať expertný systém ale nie je možné.* (dôvody prečo...)
- *Ostrieľaný manažér úlohu detektora a riešiteľa rizík zvládne.* (...)
- *Ostrieľaných manažérov je však málo.* (prečo ...)
- *Analytický a empirický prístup sú komplementárne.* (...)

Ako krízu rozpoznať?

- [ďalších N výrokov]

Ako krízu riešiť?

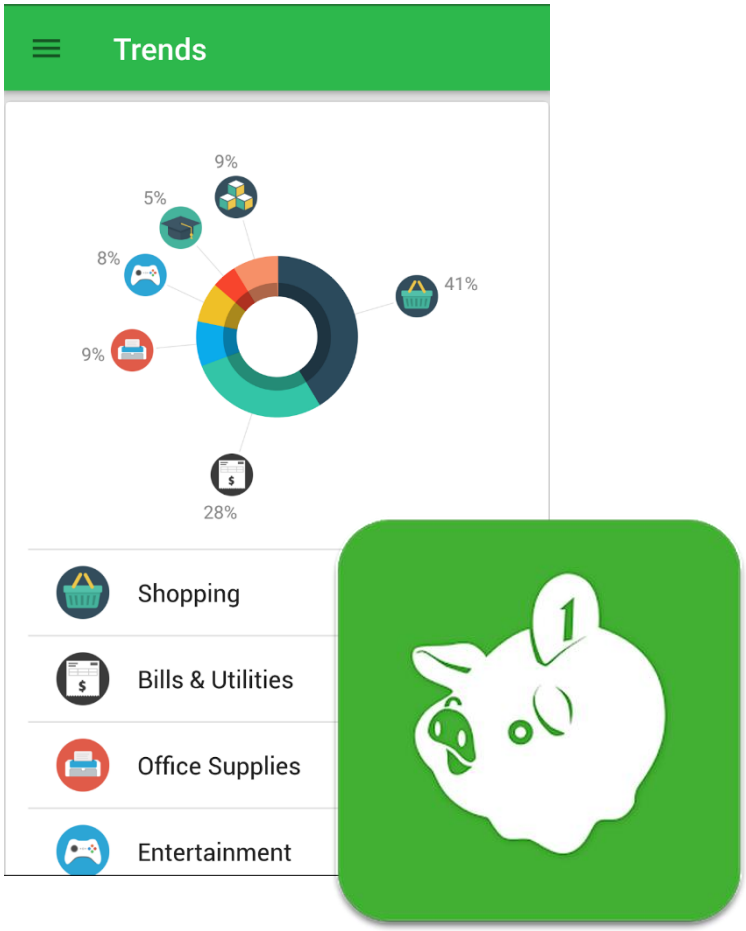
- [ďalších M výrokov]

Záver: len s podpornými nástrojmi to nedáme

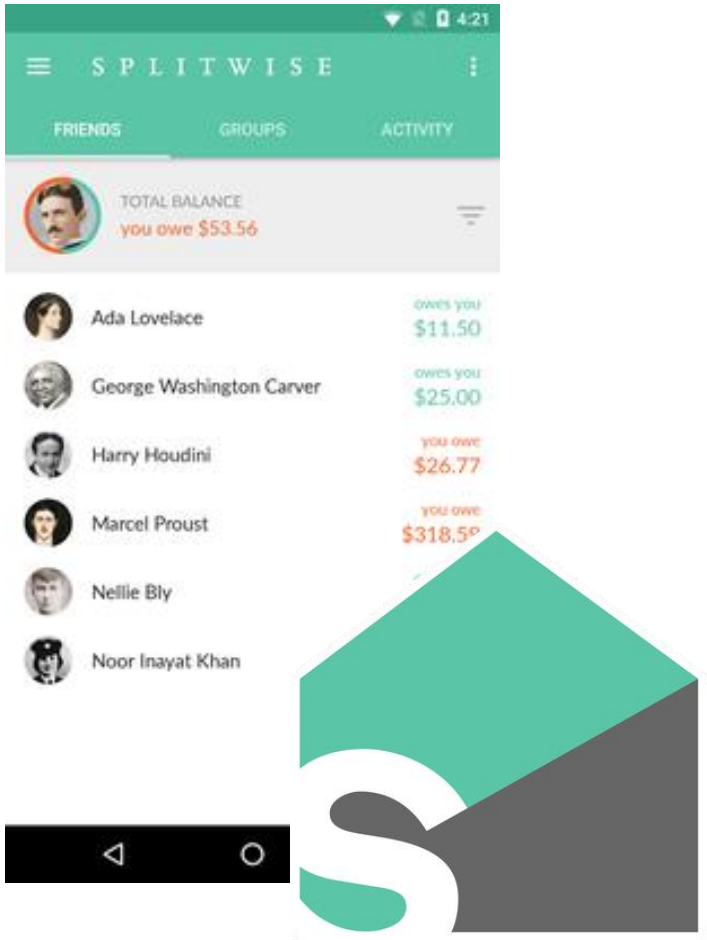
- [Zhrnutie kľúčových myšlienok eseje]

Appka týždňa: rozpočtové a podlžnostné aplikácie

Money Lover



Splitwise

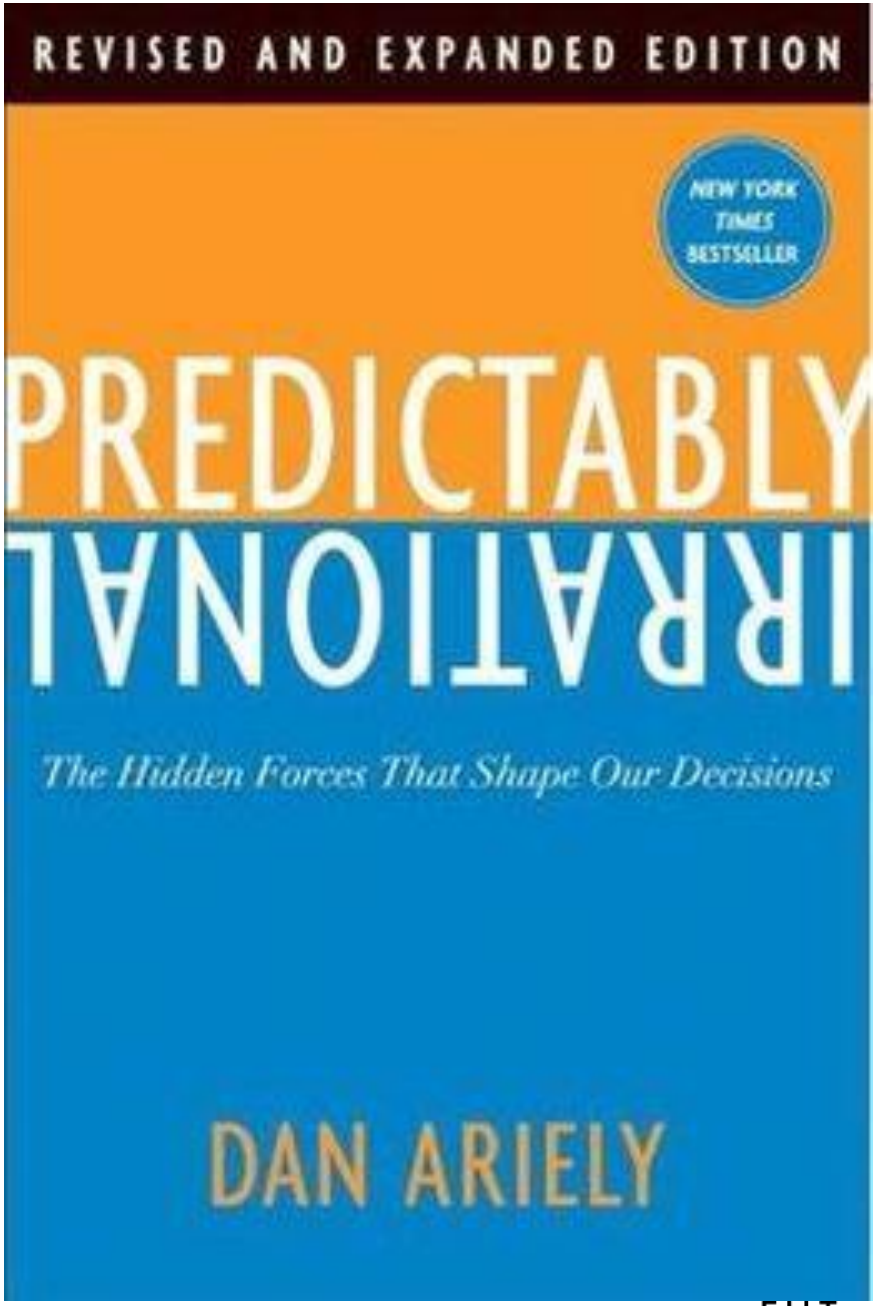


bit.ly/mip-dotaznik

STU FIIT

Kniha hodná prečítania

Dan Ariely:
Predictably Irrational



Refaktoring

==

zmeny zdrojového kódu programu,

bez zmeny jeho funkcionality,

za účelom zlepšenia jeho kvality.

bit.ly/mip-dotaznik

Refaktorujeme, aby sme znížili technický dlh

Technický dlh je ako neporiadok v kuchyni

https://pythonsweetness.tumblr.com/post/64740079543/how-to-lose-172222-a-second-for-45-minutes/amp

Analógia dlhu a neporiadku v kóde je prekvapivo presná

Pôžička = keď neporiadok vytvoríme

Splatenie = keď neporiadok odstránime

Úroky = keď sa v neporiadku snažíme pracovať (a platíme tak *extra cenu*)

Jeden rozdiel by tu bol:

technický dlh občas nemusíme zaplatiť

ale veľmi často si to len namýšľame

bit.ly/mip-dotaznik

Koho už technický dlh brzdil?
Myslím v kóde ☺

bit.ly/mip-dotaznik

Technický dlh môže viesť k veľkým problémom

Bola raz jedna veľká investičná firma, čo automaticky obchodovala na burze...

Knight

http://pythonsweetness.tumblr.com/post/64740079543/how-to-lose-172222-a-second-for-45-minutes

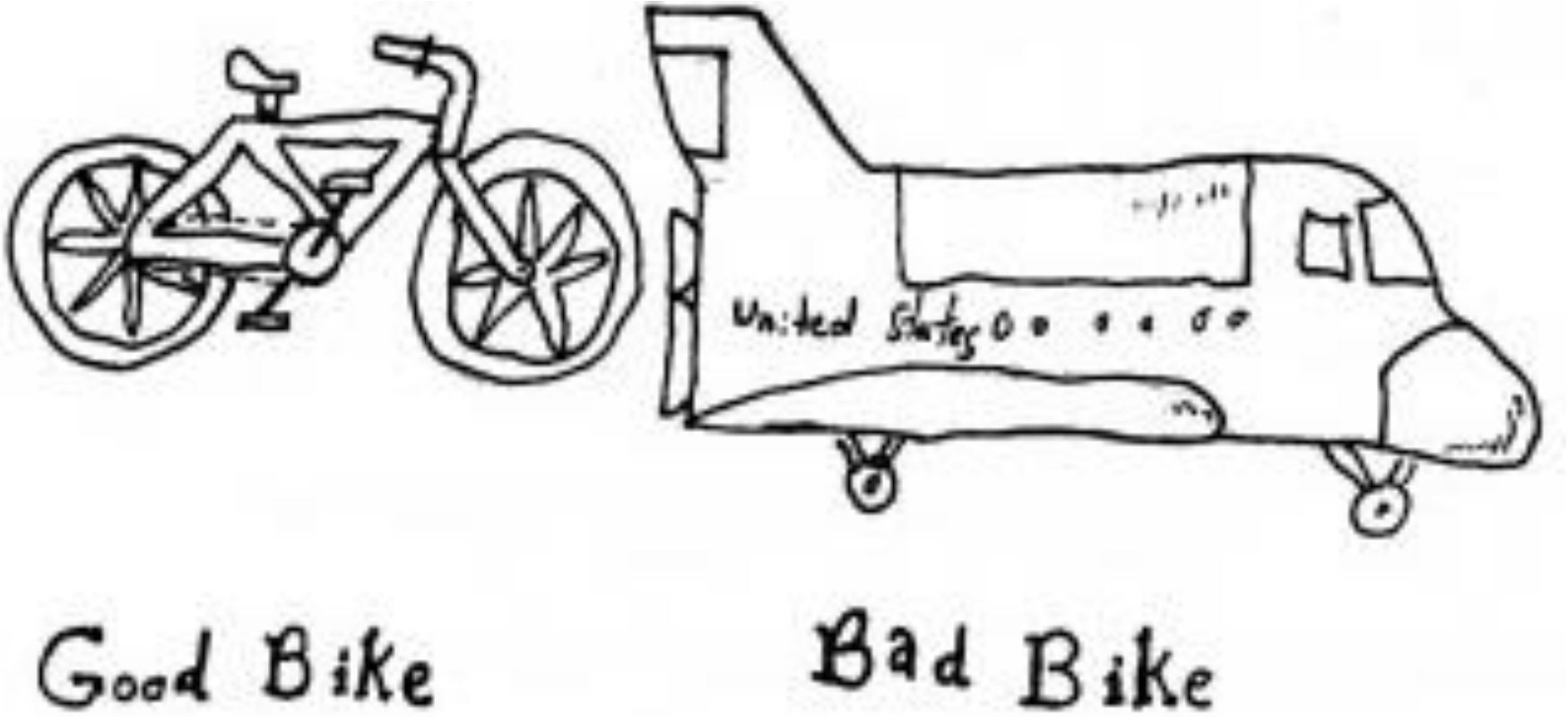

Prišli o $440 miliónov
V priebehu 45 minút

Opačný extrém technického dlhu:
over-engineering (prešpekulovanie)

Do šírky (ktorú nepotrebujeme)

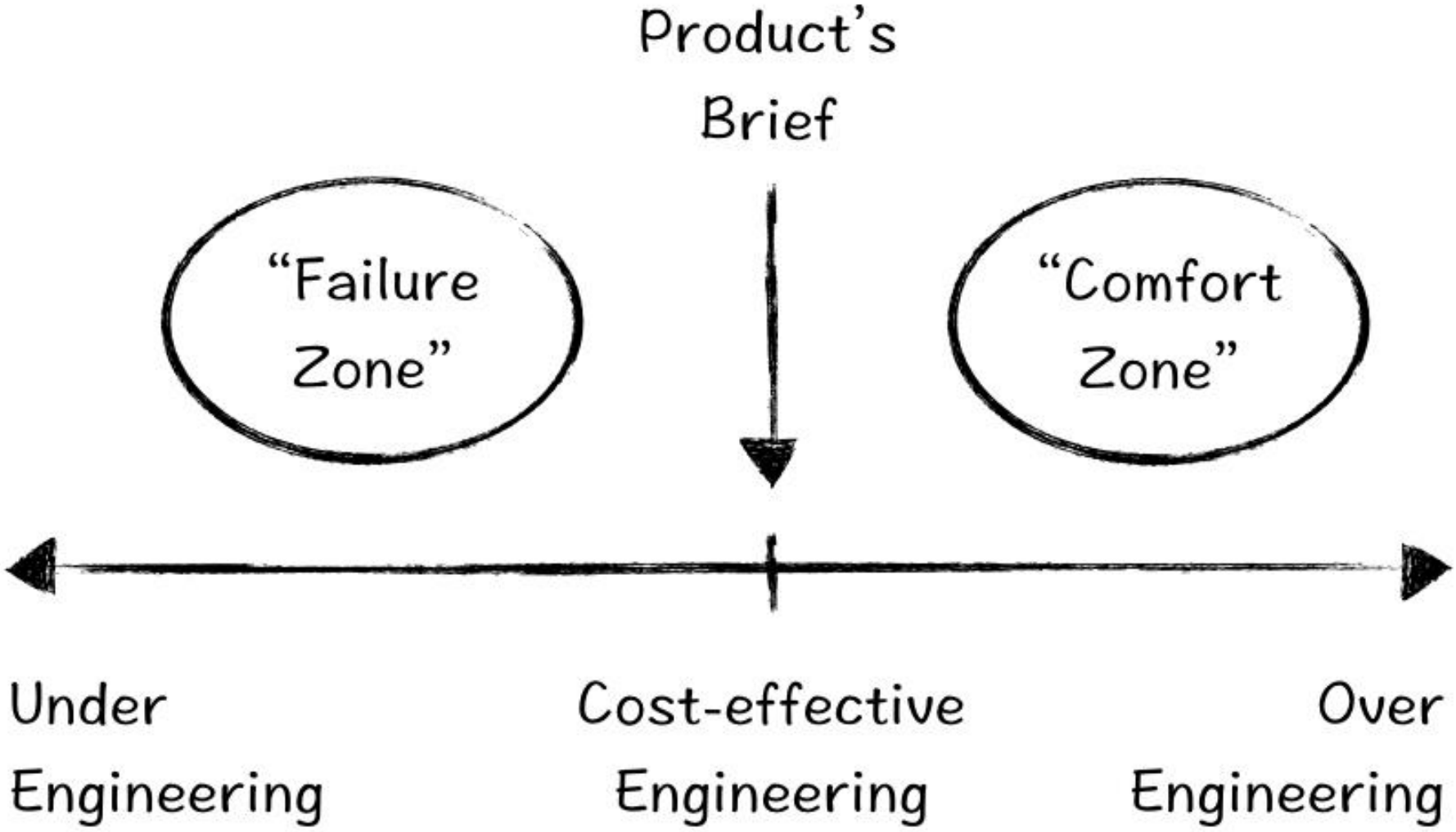
bit.ly/mip-dotaznik

Opačný extrém technického dlhu:
over-engineering (prešpekulovanie)



Do hĺbky (ktorú nepotrebujeme)

Treba pracovať vyvážene



bit.ly/mip-dotaznik

V informatike vyvážene vyzerá takto:

Opakujeme nasledovné:

1. **Dodávame funkcionalitu (dlh sa zvyšuje)**
2. **Refaktorujeme (dlh sa znižuje)**

bit.ly/mip-dotaznik

Ešte jeden pojem si dajme:
pachy v kóde (code smells)

bit.ly/mip-dotaznik

Pachy v kóde sú prejavy technického dlhu, refaktoringom sa ich snažíme odstraňovať

bit.ly/mip-dotaznik

Motivácia by bola,
poďme trochu konkrétnejšie...

bit.ly/mip-dotaznik

Disclaimer:

Budem sa snažiť byť obrazný, ale ak si nebudete vedieť nič predstaviť, povedzte to.

Aj jednoduché praktiky refaktoringu pomôžu veľa.

bit.ly/mip-dotaznik

Ešte predtým si povedzme, aké sú iné vlastnosti kódu okrem funkčnosti?

(ide mi o kvalitatívne hľadiská)

Čitateľnosť

Modifikovateľnosť

Prenositeľnosť

Poznámka: vlastností je viac

A niekedy aj stoja proti sebe (napr. výpočtová efektívnosť)

bit.ly/mip-dotaznik

Môžeme definovať aj anti-vlastnosti kódu
(podľa Roberta C. Martina, autora knihy Clean Code)

Rigídnosť – keď robím zmenu, všetko sa musí upraviť

Rozbitnosť – keď robím zmenu, ľahko sa niečo pokazí

bit.ly/mip-dotaznik

Čitateľnosť

bit.ly/mip-dotaznik

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Uvažujme názov premennej

```
int n
int number
int height
int heightPerson
int heightServicePerson
int heightServicePersonMilimeters
int heightServicePersonMilimetersOldValue
int height_service_person_milimeters_old_value
```

Buďme opisní, ale s mierou !

Vždy uvažujme kontext !

bit.ly/mip-dotaznik

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Čo robí tento kód?

```
for(int i=1; i<=5; i++) {
    for(int j=1; j<=10; j++) {
        if(i%2 == 0) {printf("X"); continue;}
        if(j%2 == 0)
            printf("*");
        else
            printf("X");
    }
    printf("\n");
}
```

```
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
```


bit.ly/mip-dotaznik

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Čomu sa tu dá zlepšiť názov je premenná cyklu (index)

```
for(int i=1; i<=5; i++) {
    for(int j=1; j<=10; j++) {
        if(i%2 == 0) {printf("X"); continue;}
        if(j%2 == 0)
            printf("*");
        else
            printf("X");
    }
    printf("\n");
}
```

```
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
```


bit.ly/mip-dotaznik

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Čomu sa tu dá zlepšiť názov je premenná cyklu (index)

```
for(int row=1; row<=5; row++) {
    for(int column=1; column<=10; column++) {
        if(row%2 == 0) {printf("X"); continue;}
        if(column%2 == 0)
            printf("*");
        else
            printf("X");
    }
    printf("\n");
}
```

```
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
```


bit.ly/mip-dotaznik

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Tie podmienky tiež nie sú bohvie čo...

```
for(int row=1; row<=5; row++) {
    for(int column=1; column<=10; column++) {
        if(row%2 == 0) {printf("X"); continue;}
        if(column%2 == 0)
            printf("*");
        else
            printf("X");
    }
    printf("\n");
}
```

```
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
XXXXXXXXXX
*X*X*X*X*X
```

```
int even(int number) {
    return (number+1)%2;
}
```


bit.ly/mip-ulohy

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Tie podmienky tiež nie sú bohvie čo...

```
for(int row=1; row<=5; row++) {
    for(int column=1; column<=10; column++) {
        if(even(row)) {printf("X"); continue;}
        if(even(column))
            printf("*");
        else
            printf("X");
    }
    printf("\n");
}
```

```
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
XXXXXXXXXX
*X*X*X*X*X
```

```
int even(int number) {
    return (number+1)%2;
}
```


bit.ly/mip-dotaznik

Veci bývajú zle pomenované
Názvy (funkcií, premenných,...) nie sú opisné

Tento kód je o poznanie zrozumiteľnejší
(pričom by sa dalo ísť ešte ďalej)

```
for(int row=1; row<=5; row++) {
    for(int column=1; column<=10; column++) {
        if(even(row)) {printf("X"); continue;}
        if(even(column))
            printf("*");
        else
            printf("X");
    }
    printf("\n");
}
```

```
X*X*X*X*X*
XXXXXXXXXX
X*X*X*X*X*
XXXXXXXXXX
*X*X*X*X*X
```

```
int even(int number) {
    return (number+1)%2;
}
```


bit.ly/mip-dotaznik

Čo s komentármi?
Písať ich veľa? Málo? Žiadne?

bit.ly/mip-dotaznik

V prvom rade nepíšte zbytočnosti

```
// check if the unit price is larger or equal to 20
if (product.UnitPrice >= 20)
```


bit.ly/mip-dotaznik

Komentáre ako API dokumentácia sú viac-menej užitočné

```
class Foo {

    /**
     * Print a <b>message</b> to the <i>console</i>, but really
     * we are just showing off the visibility of javadoc through
     * hover tool tips.  I can do all sorts in here:
     * <a href="google.ca">Google</a>.
     *
     * @param message the message to be printed
     *
     * @author Andy Clement
     */
    public void printSomething(String message) {
        print message
    }

}


new Foo().printSomething('hello world')
```

```
void Foo.printSomething(String message)

Print a message to the console, but really we are just showing off the visibility of javadoc
through hover tool tips. I can do all sorts in here: Google.

Parameters:
      message the message to be printed
Author:
      Andy Clement

                                                                          Press 'F2' for focus
```

```
class Program
{

    static void Main(string[] args)
    {
        Console.WriteLine(add());
    }

    /// <summary>
    /// Returns the sum of i and j
    /// </summary>
    /// <param name="i">the first int value</param>
    /// <param name="j">the second int value</param>
    /// <returns>the sum of i and j</returns>
    static int add(int i, int j)
    {
        return i + j;
    }
}
```

(int i, int j):int
Returns the sum of i and j
i: the first int value

Tu je komentár prehnaný: nepridáva nič

```
/// <summary>
/// Calculate the rendering face rectangle
/// </summary>
/// <param name="faces">Detected face from service</param>
/// <param name="maxSize">Image rendering size</param>
/// <param name="imageInfo">Image width and height</param>
/// <returns>Face structure for rendering</returns>
1 reference | Ana Gonzalez | 1 author, 1 change
public static IEnumerable<Face> CalculateFaceRectangleForRendering
  (IEnumerable<Azure.FaceApi.Contract.Face> faces, int maxSize, Tuple<int, int>
   imageInfo)
{
```

Tu už komentár niečo pridáva

bit.ly/mip-dotaznik

Úplne v poriadku je tiež vysoko-úrovňová dokumentácia (celých súborov či programov)

```
20⊕ import static org.apache.hadoop.hdfs.protocol.proto.DataTransferProtos.Status.SUCCESS;
96
97
98⊖ /****************************************************************
99   * DFSOutputStream creates files from a stream of bytes.
100  *
101  * The client application writes data that is cached internally by
102  * this stream. Data is broken up into packets, each packet is
103  * typically 64K in size. A packet comprises of chunks. Each chunk
104  * is typically 512 bytes and has an associated checksum with it.
105  *
106  * When a client application fills up the currentPacket, it is
107  * enqueued into dataQueue.  The DataStreamer thread picks up
108  * packets from the dataQueue, sends it to the first datanode in
109  * the pipeline and moves it from the dataQueue to the ackQueue.
110  * The ResponseProcessor receives acks from the datanodes. When an
111  * successful ack for a packet is received from all datanodes, the
112  * ResponseProcessor removes the corresponding packet from the
113  * ackQueue.
114  *
115  * In case of error, all outstanding packets and moved from
116  * ackQueue. A new pipeline is setup by eliminating the bad
117  * datanode from the original pipeline. The DataStreamer now
118  * starts sending packets from the dataQueue.
119  ****************************************************************/
120 @InterfaceAudience.Private
121 public class DFSOutputStream extends FSOutputSummer
122     implements Syncable, CanSetDropBehind {
123   private static final int MAX_PACKETS = 80; // each packet 64K, total 5MB
```

Komentáre majú problém s aktuálnosťou

bit.ly/mip-dotaznik

KEEP
CALM
AND
WRITE
CODE

bit.ly/mip-dotaznik

Čitateľnosť znižuje ak sú veci zle umiestnené

Dlhšie trvá ich hľadanie

Práca je menej ergonomická

Katastrofa: robíme veci znova, lebo o nich nevieme

bit.ly/mip-dotaznik

Veci čo súvisia dávať spolu,
veci čo nesúvisia nedávať spolu.

Na to máme tie súbory a priečinky...

Ale aj v rámci súboru: prázdne riadky dobre oddeľujú

bit.ly/mip-dotaznik

Kód občas nemá čitateľné formátovanie

bit.ly/mip-dotaznik

```
        <ul id="bigBarNavigation">
<li><a href="/">HOME</a>
    </li><li><a href="/contact">CONTACT US</a></li><li>
        <a href="/about">ABOUT US</a></li></ul>
```

Confusing mess...

```
<ul id="bigBarNavigation">
    <li><a href="/">HOME</a></li>
    <li><a href="/contact">CONTACT US</a></li>
    <li>
        <a href="/about">ABOUT US</a>
        <div class="subMenu">
                <!-- Just an example to
                     show indentation -->
        </div>
    </li>
</ul>
```

Nice and clean. mmmmmmmm...

```php
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```

bit.ly/mip-dotaznik

Nielen odsadzovanie, ale aj zarovnávanie

```
#include <stdio.h>

int someDemoCode(   int fred,
                    int wilma)
{
        x();                                /* try making   */
        print("hello again!\n");            /* this comment */
        makeThisFunctionNameShorter();      /* a bit longer */
        for (i = start; i < end; ++i)
        {
                if (isPrime(i))
                {
                        ++numPrimes;
                }
        }
        return numPrimes;
}
```

bit.ly/mip-dotaznik

Trochu záleží na jazyku (ale nie veľmi)

Python	Other language
if a is not 5 :	if (a != 5) { ...
if a is 5 :	if (a == 5) { ....

while (*a* is True and *b* is False) :
python code
	while (a == true && b == false)
	{
	....other code
	}

while (a is True or B is False) :	while (a == true || b == false) { ...
print "hi there"	console.log("hi there")

bit.ly/mip-dotaznik

Modifikovateľnosť

bit.ly/mip-dotaznik

Symptómom zlej modifikovateľnosti je, že zmeny idú len pomaly a s veľkým úsilím

Veľa možných príčin (pachov)

bit.ly/mip-dotaznik

Častým pachom je duplicitný kód

copy-paste chyby

Prečo je to také zlé?

bit.ly/mip-dotaznik

Iným pachom je shotgun surgery
a.k.a jednu vec robím na veľa miestach

bit.ly/mip-dotaznik

Príliš otvorený kód. (Kód by mal skrývať veci, ktoré sú v danom kontexte nepodstatné.)

Tento kód robí viac vecí.

```
for(int row=1; row<=5; row++) {
    for(int column=1; column<=10; column++) {
        if(even(row)) {printf(“X”); continue;}
        if(even(column))
            printf(“*”);
        else
            printf(“X”);
    }
    printf(“\n”);
}
```


bit.ly/mip-dotaznik

Kúsok lepší kód. (Kód by mal skrývať veci, ktoré sú v danom kontexte nepodstatné.)

```
void writePattern() {
    for(int row=1; row<=5; row++) {
        writeLine(row);
        printf("\n");
    }
}

void writeLine(int row) {
    if(even(row)) writeSolidLine();
    else writeVaryingLine();
}

void writeSolidLine() {
    for(int column=1; column<=10; column++) printf("X");
}

void writeVaryingLine() {
    for(int column=1; column<=10; column++)
        if(even(column)) printf("*");
        else printf("X");
}
```

```
int even(int number) {
    return (number+1)%2;
}
```

Každá funkcia robí jednu vec

Málo parametrov

Enkapsulácia

X X X X X*

XXXXXXXXXX

X*X*X*X*X*

Funkcie čo majú veľa parametrov zapáchajú...

```
drawBlock(float length,
          float width,
          float height,
          int color,
          int dashing) { … }
```

A hrozí že budú pribúdať

Riešenie: použiť na opis kvádra štruktúru

```
struct Block {
    float length,
    float width,
    float height,
    int color,
    int dashing
};
```

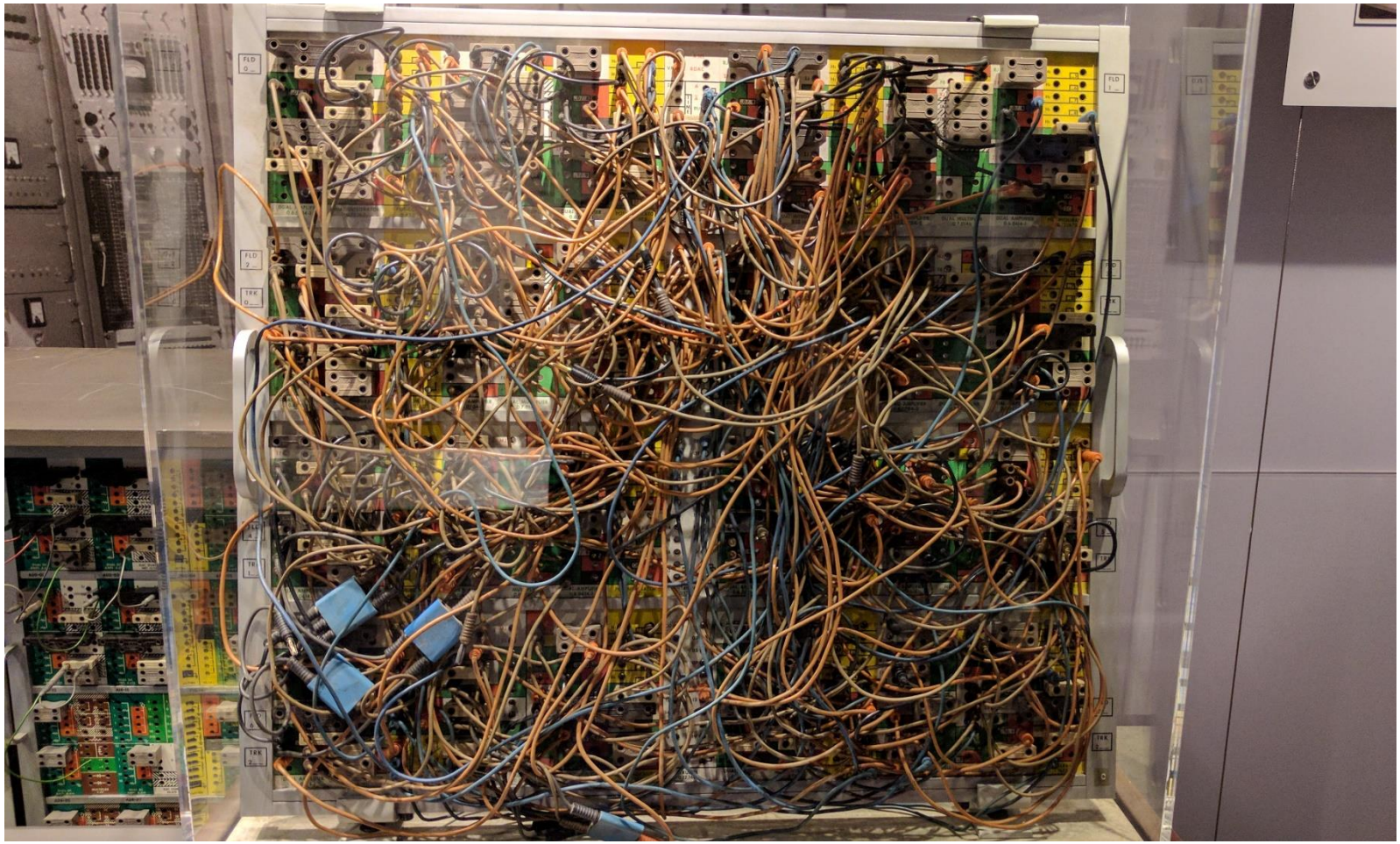
```
draw(Block block) {…}
computeMass(Block block) {…}
computeVolume(Block block) {…}
```

bit.ly/mip-dotaznik

Prenosnosť

bit.ly/mip-dotaznik

Hlavným nepriateľom prenositeľnosti sú závislosti



bit.ly/mip-dotaznik

Globálne premenné sú receptom na katastrofu, lebo priam generujú závislosti

Porušovanie enkapsulácie:

veci prestávajú byť pekne skryté a oddelené

Funkciu, ktorá používa globálnu premennú neviete znovupoužiť v inom programe

Hrozba „vedľajších efektov“:

nikdy si nie ste istí, ktorá funkcia čo mení

bit.ly/mip-dotaznik

Magické čísla

```
void writePattern() {
    for(int row=1; row<=5; row++) {
        writeLine(row);
        printf("\n");
    }
}

void writeVaryingLine() {
    for(int column=1; column<=10; column++)
        if(even(column)) printf("*");
        else printf("X");
}
```

Magické čísla – nahradiť konštantami

```
const int height = 5;
const int width = 10;

void writePattern() {
    for(int row=1; row<=height; row++) {
        writeLine(row);
        printf("\n");
    }
}


void writeVaryingLine() {
    for(int column=1; column<=width; column++)
        if(even(column)) printf("*");
        else printf("X");
}
```

Počkať, globálne???

Globálne konštanty (až tak) nevadia

Viete ich dať napr. do konfiguračného súboru teda na jedno miesto

Vadia globálne premenné

bit.ly/mip-dotaznik

Vo vašich zadaniach sa pozrite aspoň na ...

Je kód správne naformátovaný?

Správne odsadenia, ale aj prázdne riadky či zarovnania?

Nedajú sa zlepšiť pomenovania vecí?

Kuk aj na komentáre, nedajú sa nahradiť?

Sú veci na správnych miestach?

Funkcie ktoré spolupracujú sú blízko seba?

Sú funkcie krátke? Robia jednu vec?

Máte tam magické čísla?

Dáva súborová štruktúra zmysel?

Nedajú sa vylepšiť zložité ifovačky?

Ošetrujte okrajové prípady elegantne

Ďalšie čítanie

Blog na margo čitateľnosti (expert vs beginner):

https://simpleprogrammer.com/2013/04/14/what-makes-code-readable-not-what-you-think/

Knihy:

Clean Code

Code Complete

bit.ly/mip-dotaznik

bit.ly/mip-dotaznik

Vo vašich zadaniach sa pozrite aspoň na ...

Je kód správne naformátovaný?

Správne odsadenia, ale aj prázdne riadky či zarovnania?

Nedajú sa zlepšiť pomenovania vecí?

Kuk aj na komentáre, nedajú sa nahradiť?

Sú veci na správnych miestach?

Funkcie ktoré spolupracujú sú blízko seba?

Sú funkcie krátke? Robia jednu vec?

Máte tam magické čísla?

Dáva súborová štruktúra zmysel?

Nedajú sa vylepšiť zložité ifovačky?

Ošetrujte okrajové prípady elegantne

Zo spätnej väzby

Keď sme my vaši kolegovia, ste vy náš kolega?

Áno.

Akcia týždňa? Nejaký event ktorý bude prebiehať buď v Bratislave/sa oplatí sledovať online?

Rubyslava (ale nie online)
https://www.facebook.com/events/587144902028858/

Ďalšia nevýhoda prečo nepoužívať ntb na prednáškach:
Na prednáške som si kontroloval projekt na ZPRPR, chalan dva rady za mnou si ho fotil.

Prečo ste neboli ako najlepší prednášajúci na FIIT ? :/

Sú aj lepší ☺

Výsledky boli za letný semester

bit.ly/mip-dotaznik

Zo spätnej väzby

Linky do Github zadania ešte pošlem :(

Ako funguje svetlo v knižnici? (samovolné zhasínanie)

Personál knižnice → personál TPO (údržba)

Nejaké rady ohľadne financií? (ako vyžiť popri škole)

Príde Metod: Manažment osobných financií.

Koľko kreditov je minimum za prvý semester?

15 (studijny poriadok, cl. 17)

https://www.fiit.stuba.sk/buxus/docs/Studium/SP2019-20.pdf

výnimočné situácie konzultovať s prodekanom

bit.ly/mip-dotaznik

Zo spätnej väzby

Dá sa stíhať Fiitku a zároveň spať aj 8 hodín denne?

Neviem koľko som spal ja ...

... ale viem koľko som pregejmil.

Na extra prednáške ste spomenuli, že ste zažili "vyhorenie". Ako to prebiehalo a nejaké rady, ako sa tomu vyhnúť?

Bral som si na seba priveľa projektov.

Ako je to so známkami a výškou štipendia? Dá sa nejaké získať aj s "Céčkami"?

Treba byť v horných 10%. Mne na to céčka nestačili.

bit.ly/mip-dotaznik

Zo spätnej väzby

Ako by ste riešili situáciu, keď máte pocit, že ste v "začarovanom kruhu," pretože neustále doháňate povinnosti (učivo, zadania na poslednú chvíľu) a tým pádom nemáte čas začať pracovať na aktuálnych veciach včas? Ako si vhodne určiť priority (ktoré predmety, ktoré zadania...)?

Zhodnotiť realistickosť prejdenia všetkých

Dropnúť podľa potreby

Radšej failovať „riadenie“

Niekedy je lepšie robiť radšej aktuálne zadanie

Skúste sa poradiť aj s cvičacimi. Možno budú receptívni.

Hlavne však robiť veci lepšie ďalší semester

Identifikujte si, v čom ste robili chyby

bit.ly/mip-dotaznik

Zo spätnej väzby

Do budúcna trochu lepšia komunikácia s cvičiacimi pri zadani z Latexu.

- **Robíme čo môžeme**
- **Niektoré veci sa synchronizujú dodatočne**

bit.ly/mip-dotaznik

Zo spätnej väzby

Na izbe na ubytovni som sám. Pomaly nikoho nepoznám v Bratislave. Vždy, keď prídem na ubytovňu, tak stratím chuť do všetkého, nemám energiu sa učiť, jesť, ísť von. Káva ani energeťáky mi nepomáhajú. Bol som už vonku s kamarátmi a bolo mi fajn, ale vždy keď sa vrátim späť na izbu tak stratím všetku energiu and nevládzem nič robiť. Kvôli tomu zaostávam v učive a nestíham s projektami, lebo aj keď sa donútim učiť sa na izbe, tak mi to ide strašne pomaly. Ako sa môžem tohoto pocitu únavy a nevôle zbaviť?

bit.ly/mip-dotaznik

Zo spätnej väzby

Ako je chodenie na tejto výške cez vianoce? Budú nejaké prázdniny, je po skuškovom, alebo ako? A kedy sa potom znova ide do školy?

V prípade že ma nevyrazia a prejdem, uvidím vás ešte na nejakom inom predmete? (Ak nie, tak je to škoda)

Uvidíme sa na PSI (4. semester)

a na MTS/Tímáku (1. ročník Ing.)

bit.ly/mip-dotaznik

Zo spätnej väzby

Nebudem sa moct 21.11. zucastnit extra prednasky o 17. novembri - bude z nej video?

Bude

Prečo sú na bočných stenách 2 sady zásuvok skoro pri strope? Som síce vysoký ale tam si notebook asi nenabijem.

Záhada aj pre mňa

ako sme to nakoniec dopadli s tými počítačmi vs old-school poznámkami? :)

Vyhrali ručne písané poznámky

bit.ly/mip-dotaznik

Zo spätnej väzby

skor otázka k extra prednáške; ako prebieha výskum pri phd? velmi ma zaujima, ale narozdiel od bc a ing is neviem moc predstaviť taky typicky den/tyždeň doktoranda

Keď by som mal záujem. Je možné žeby som mohol byť 24h v škole? (Týmto by som vyzval ostatných na taký menší challenge)

Vysvetlite kto je to softverovy architekt a ci na fakulte je aj predmet ktory sa venuje tomuto odboru. Nie je to to iste ako softverovy inzinier? Aky je rozdiel medzi nimi?

bit.ly/mip-dotaznik

Zo spätnej väzby

Prečo by mali byť študenti špičkoví, ale mnohí cvičiaci alebo prednášajúci špičkoví nie sú? Príklad: ak študent nepríde raz s vyriešenou úlohou na UMZI alebo SMAT, automaticky dostáva neospravedlnenú absenciu, čím "letí preč" z predmetu. Za to ak cvičiaci príde na cvičenie s 15 minutovým meškaním a ďalších 20 minút telefonuje, pretože si potrebuje niečo vybaviť, tak je to v poriadku. Aký máte na to názor a "čo sa dá" proti tomu robiť?