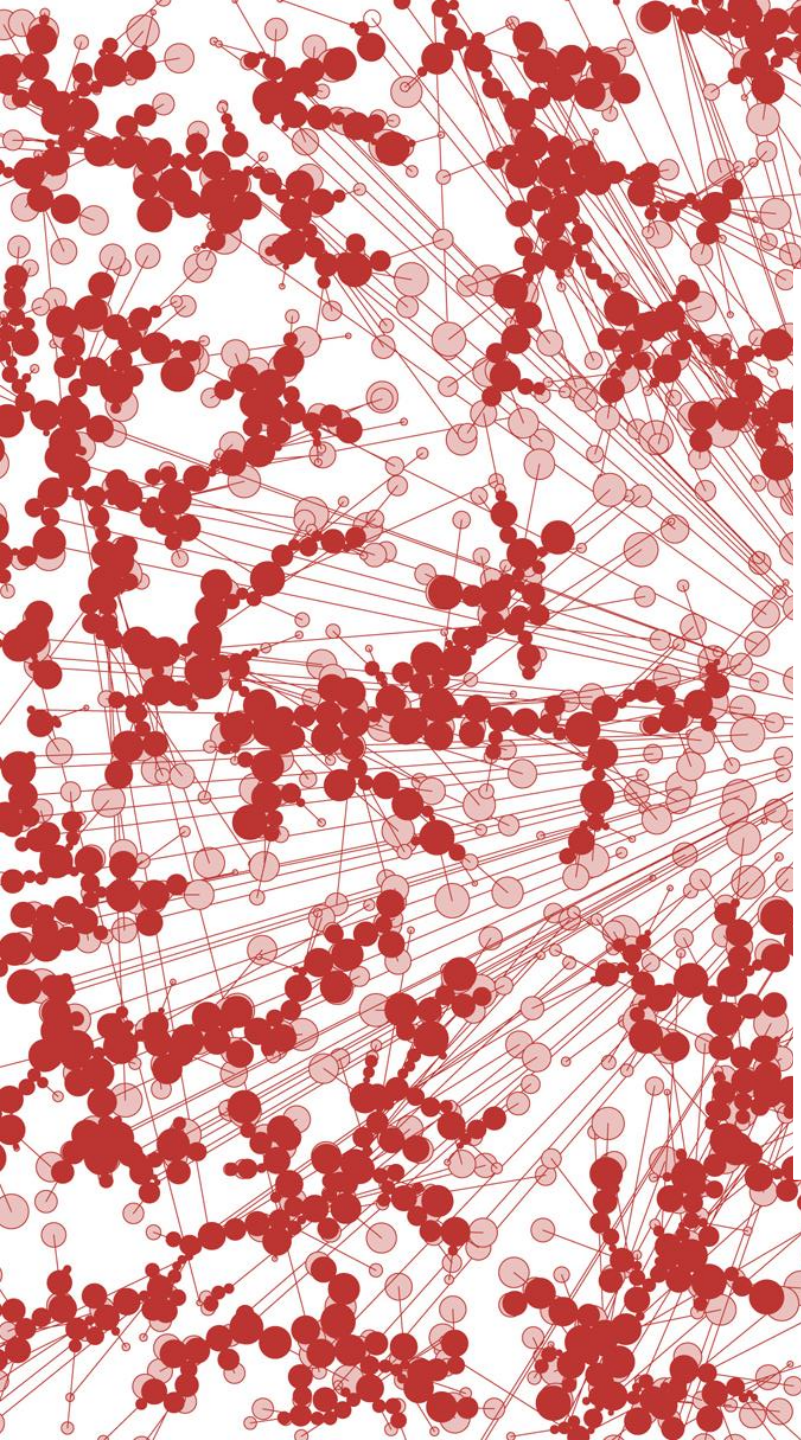


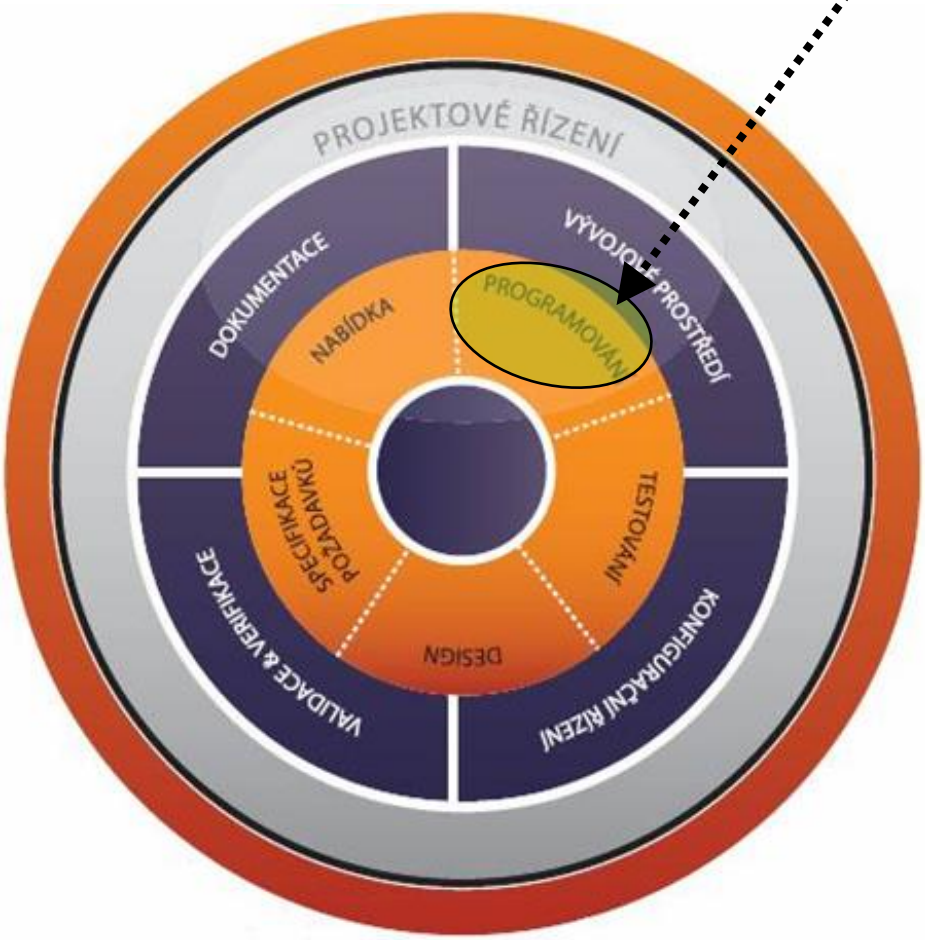
PROFINIT
new frontier group

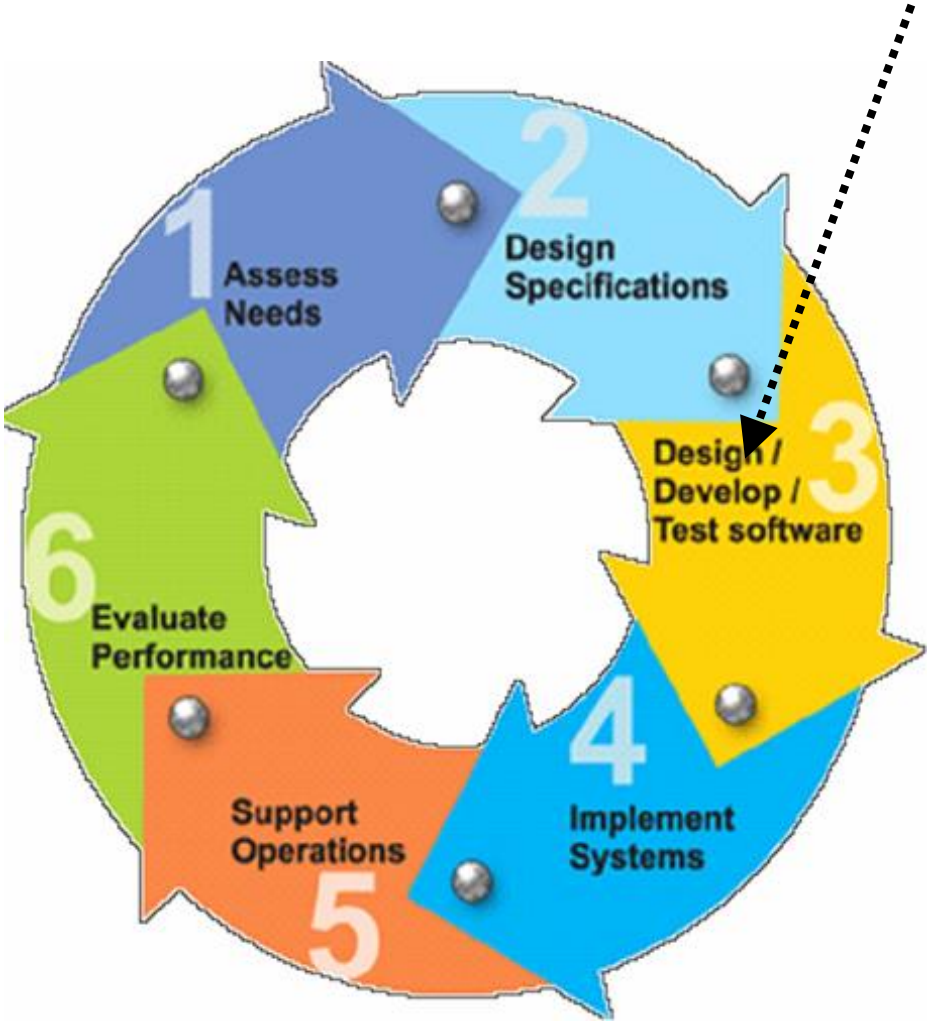
Construction

Tomáš Krátký, Bohumír Zoubek, Jiří Toušek
tomas.kratky , bohumir.zoubek, jiri.tousek@profinit.eu
http://www.profinit.eu/cz/podpora-univerzit/univerzitni-vyuka

Softwarový proces

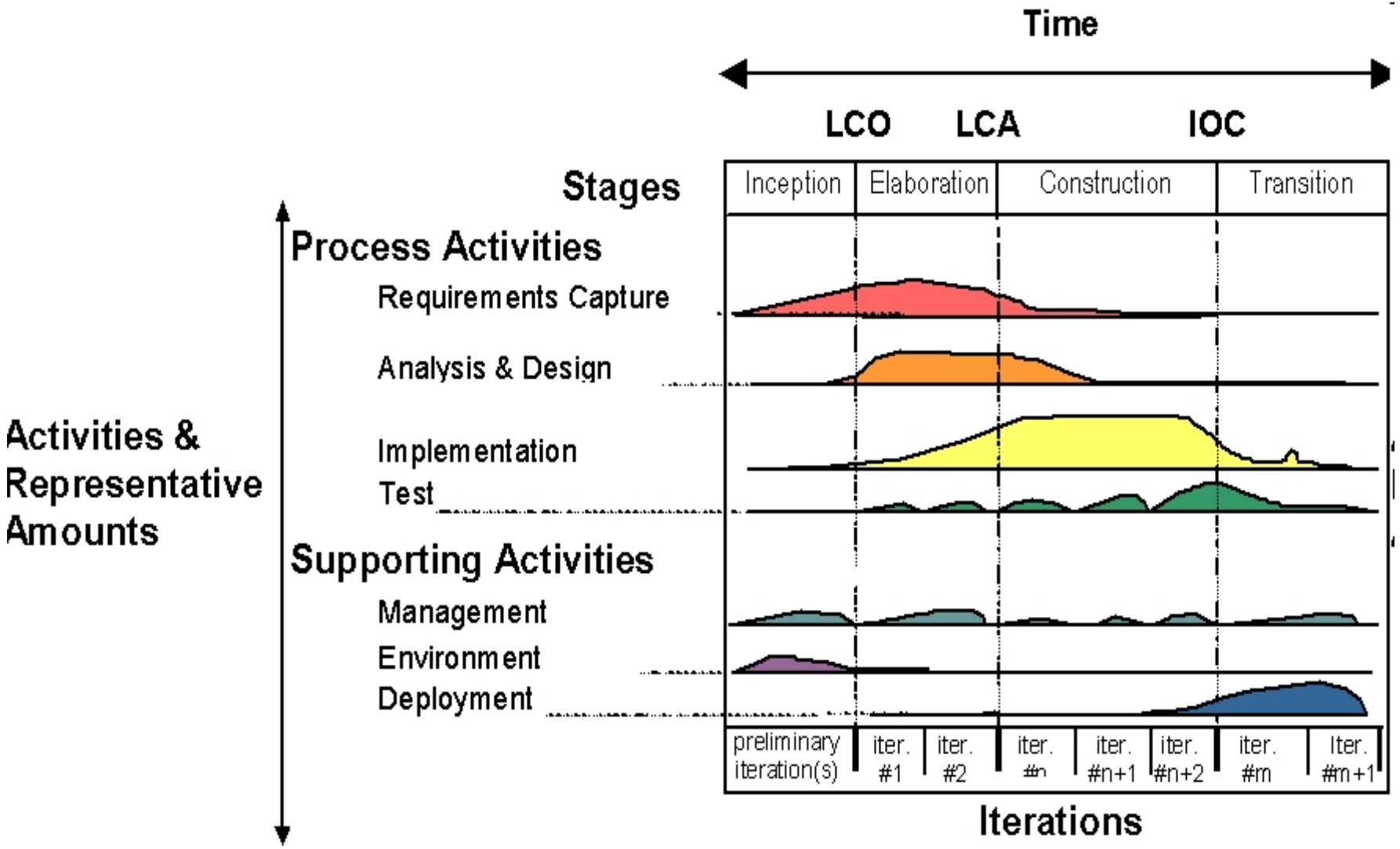
PROFINIT
new frontier group





Softwarový proces

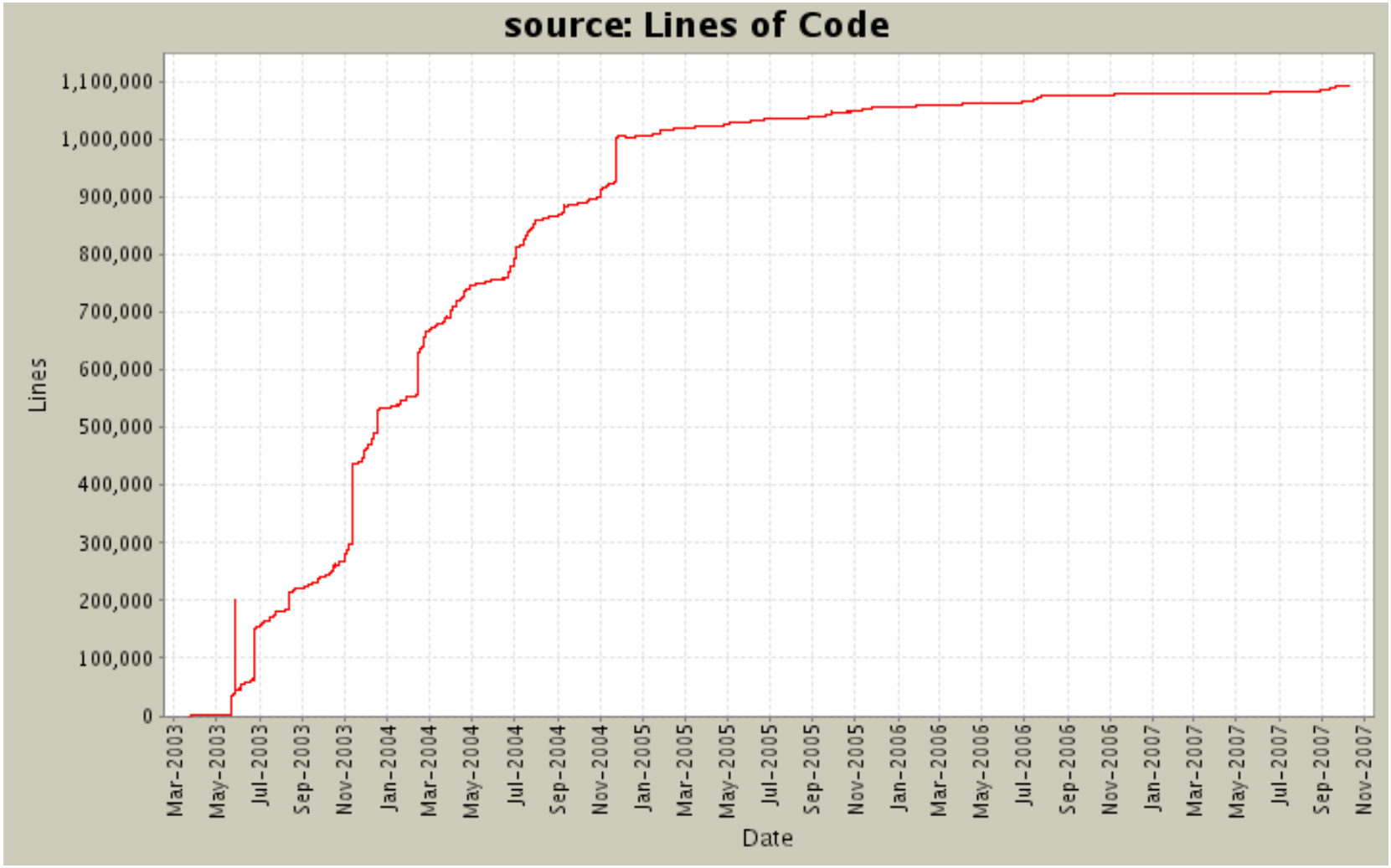
PROFINIT
new frontier group



Převzato z http://csse.usc.edu/csse/research/CORADMO/

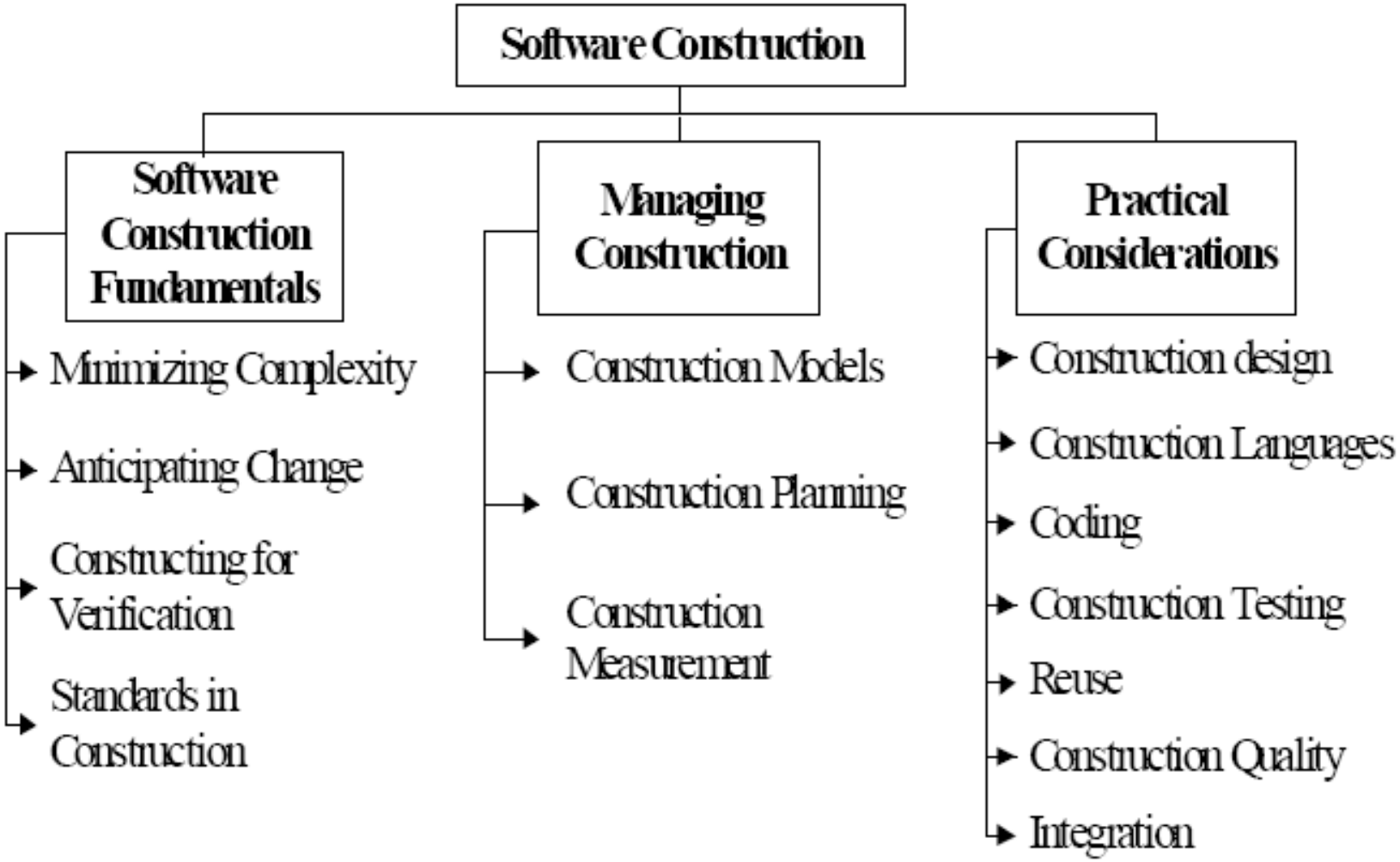
Softwarový proces

PROFINIT
new frontier group



SWEBOK

PROFINIT
new frontier group



Poznatky z praxe

PROFINIT
new frontier group

- Programovat, když je jasné co (a jak)
  - viz vazba na LCA
  - některé části lze i předem (např. aspekty typu security, logování, výjimky, …)
- Ctít architekturu a dizajn
  - jak zajistit dodržování základních pravidel?
    - automatizace kontrol, nástroje, …
- Používat vhodné nástroje a hotové věci
  - testování, IDE, CI, skripty, evidence, …
  - znalost frameworks a knihoven je zásadní → orientace, přehled, …

Poznatky z praxe

PROFINIT
new frontier group

- Znát a chápat základní koncepty SW konstrukce
  - synchronizace, vlákna, deadlocks, datové struktury, algoritmy, …
- Psát testy (testovatelnost!), aplikovat statickou analýzu kódu
- Pečovat o kód (revize, přezkoumání, …)
- Vyhýbat se „rychlým, dočasným“ řešením, mít jasná pravidla, refaktoring
- Chápat ekonomiku projektu (firmy)
  - cena za bastlení, nízká kvalita
  - cena za „gold plating“

Zajímavá témata

Test driven development

PROFINIT
new frontier group

- Co to znamená?
  - Testy
  - Implementace
  - Testy
  - ...
- Jaká je podstata?
  - Testovatelný SW
  - Existence testů
- Pozor
  - Na zahlcení testy
  - Na údržbu testů

Refaktoring

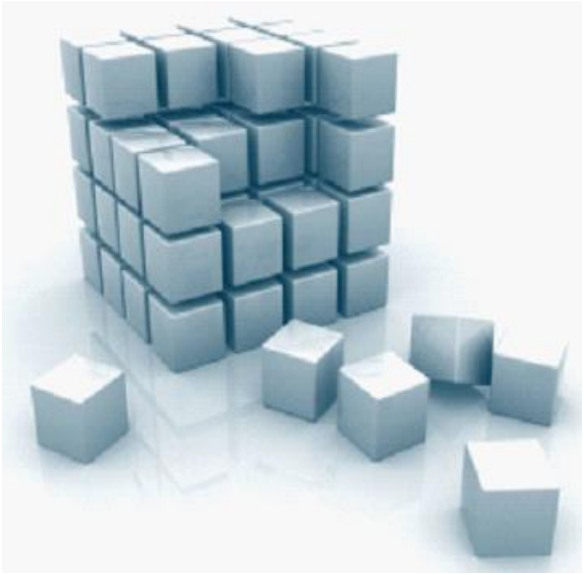
PROFINIT
new frontier group

What is Refactoring?

Refactoring is a disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior. Its heart is a series of small behavior preserving transformations. Each transformation (called a 'refactoring') does little, but a sequence of transformations can produce a significant restructuring. Since each refactoring is small, it's less likely to go wrong. The system is also kept fully working after each small refactoring, reducing the chances that a system can get seriously broken during the restructuring.

Martin Fowler, refactoring.com

- Není to zahození všeho a postavení z nuly
- Existence testů nutná!
- Dobrý design (např. encapsulation) podporuje refactoring



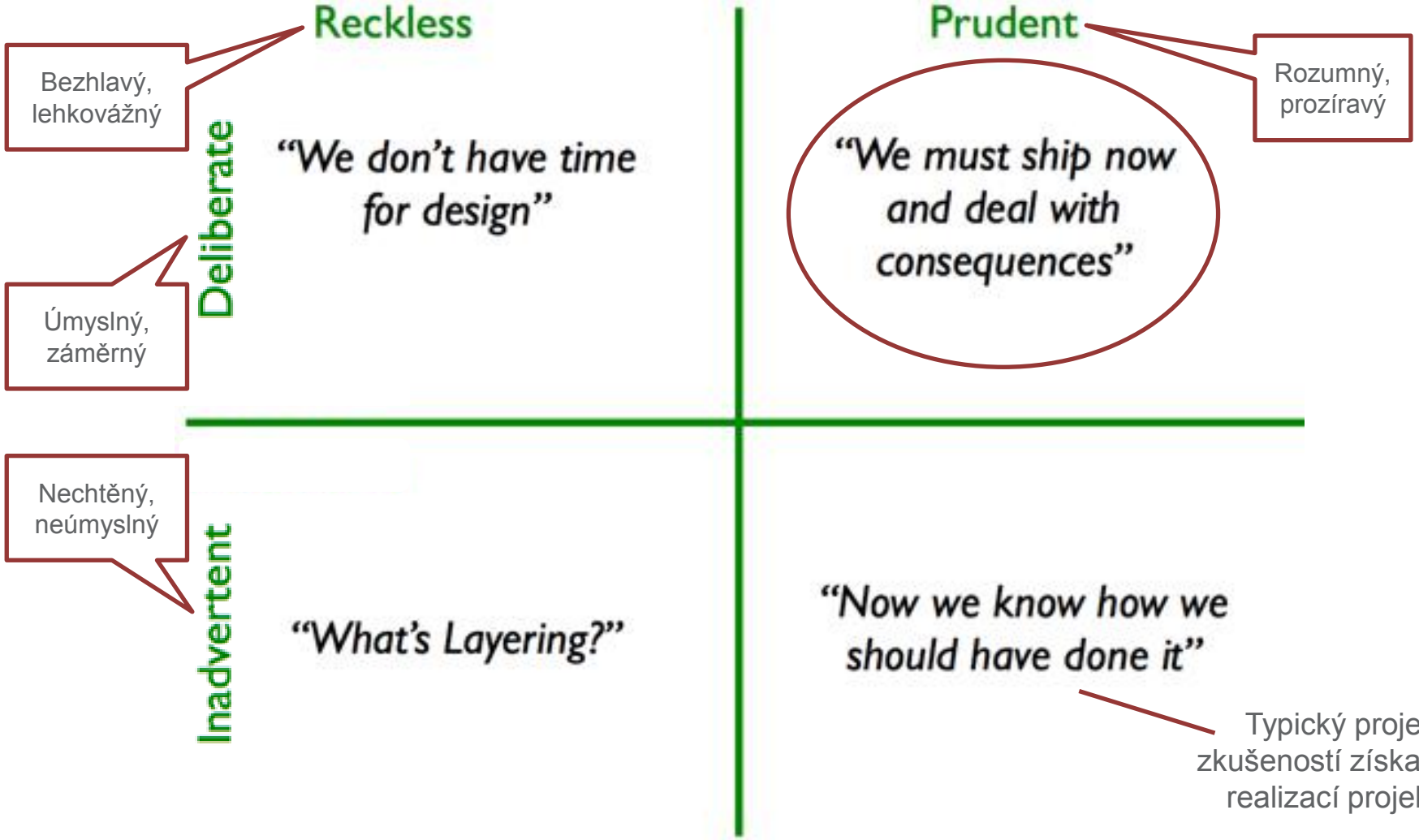
Self-documenting code

PROFINIT
new frontier group

- Skutečný význam ?
  - neznamená absenci dokumentace
  - čitelný kód nenahradí koncepční dokumentaci (architektura, design moments, …)
- Nezaměňovat s pojmem Literate programming
  - alternativa ke structured programming
  - autorem je slavný Donald Knuth
  - konstrukce velmi podobné lidskému jazyku
- Problémy při chybějící dokumentaci
  - Význam větších funkčních celků
  - Pre/post conditions, invariants
  - Inheritance
  - … a mnoho dalších …
  - Ukázka View-controller kuchařka

Technical debt

PROFINIT
new frontier group



Více informací na http://martinfowler.com/bliki/TechnicalDebtQuadrant.html

Zájem, vzdělávání, komunita

PROFINIT
new frontier group

- http://www.javacodegeeks.com/
- http://www.infoq.com/
- http://www.theserverside.com/
- http://martinfowler.com/
- http://www.joelonsoftware.com/

… a mnoho dalších …

… o knihách ani nemluvě …

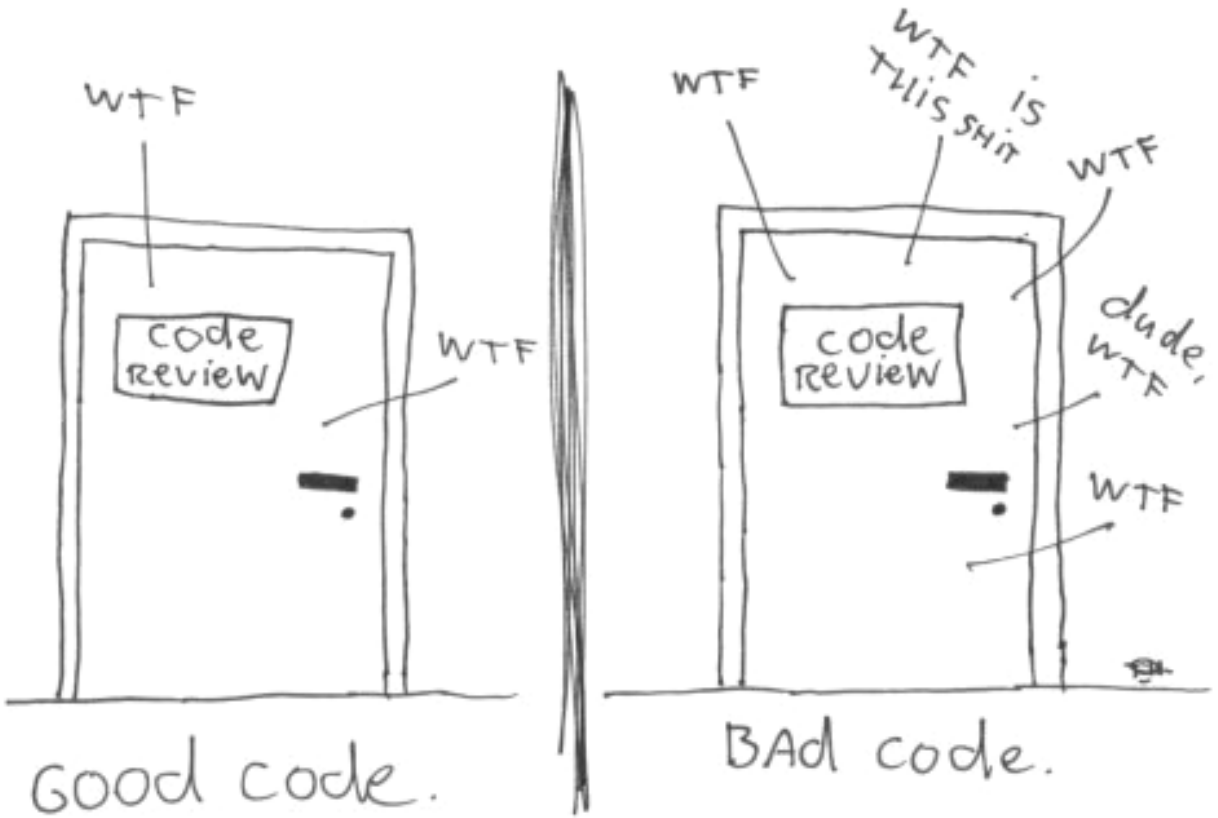
Co je to kvalitní kód?

Kvalitní kód je...

PROFINIT
new frontier group

- Bezchybný
- Rychlý / Efektivní
- Krátký?
- Rafinovaný?
- **Srozumitelný**
- **Udržovatelný**

The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE



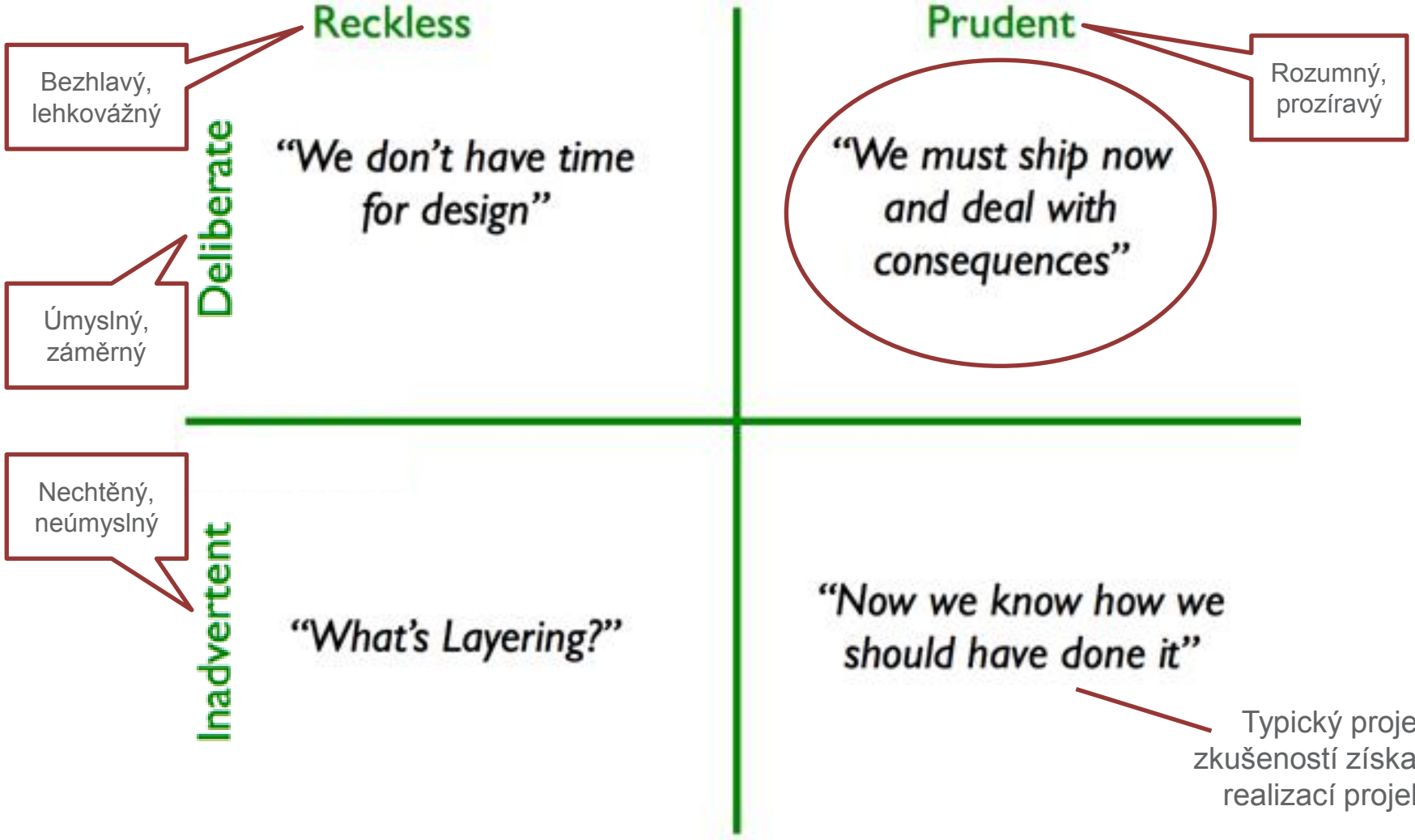
Proč psát srozumitelný kód

PROFINIT
new frontier group

- Kód píšete jednou, číst ho (vy nebo někdo jiný) budete pravděpodobně vícekrát
- Pokud váš kód někdo nepochopí, snadno v jeho použití nebo úpravě udělá chybu
- Nesrozumitelný kód nikdo nebude chtít používat

Kdy nepsat kvalitní kód?

PROFINIT
new frontier group



Jak psát srozumitelný kód

Co bude uživatel vaší metody číst

PROFINIT
new frontier group

1. Jméno metody
2. Deklarované typy parametrů a návratové hodnoty
3. Jména deklarovaných parametrů
4. Javadoc
5. Samotný kód metody

- Čím dříve se dozví vše, co potřeboval, tím dříve může přestat číst (a ušetřit čas)

Příklad: StringUtils

PROFINIT
new frontier group

```
join()
```

Příklad: StringUtils

PROFINIT
new frontier group

```
public static String join(Object[], String, int, int)
```

Příklad: StringUtils

PROFINIT
new frontier group

```
public static String join(Object[] array, String separator, int startIndex, int endIndex)
```

Příklad: StringUtils

PROFINIT
new frontier group

```
/**
Joins the elements of the provided array into a single String containing the provided list of elements.

No delimiter is added before or after the list. A null separator is the same as an empty String (""). Null objects or
empty strings within the array are represented by empty strings.

 StringUtils.join(null, *)               = null
 StringUtils.join([], *)                 = ""
 StringUtils.join([null], *)             = ""
 StringUtils.join(["a", "b", "c"], "--")  = "a--b--c"
 StringUtils.join(["a", "b", "c"], null)  = "abc"
 StringUtils.join(["a", "b", "c"], "")    = "abc"
 StringUtils.join([null, "", "a"], ',')   = ",,a"

Parameters:
    array the array of values to join together, may be null
    separator the separator character to use, null treated as ""
    startIndex the first index to start joining from. It is an error to pass in an end index past the end of the array
    endIndex the index to stop joining from (exclusive). It is an error to pass in an end index past the end of the array
Returns:
    the joined String, null if null array input
*/
public static String join(Object[] array, String separator, int startIndex, int endIndex)
```

Příklad: StringUtils

PROFINIT
new frontier group

```
/**
Joins the elements of the provided array into a single String containing the provided list of elements.

No delimiter is added before or after the list. A null separator is the same as an empty String (""). Null objects or
empty strings within the array are represented by empty strings.

 StringUtils.join(null, *)               = null
 StringUtils.join([], *)                 = ""
 StringUtils.join([null], *)             = ""
 StringUtils.join(["a", "b", "c"], "--")  = "a--b--c"
 StringUtils.join(["a", "b", "c"], null)  = "abc"
 StringUtils.join(["a", "b", "c"], "")    = "abc"
 StringUtils.join([null, "", "a"], ',')   = ",,a"

Parameters:
    array the array of values to join together, may be null
    separator the separator character to use, null treated as ""
    startIndex the first index to start joining from. It is an error to pass in an end index past the end of the array
    endIndex the index to stop joining from (exclusive). It is an error to pass in an end index past the end of the array
Returns:
    the joined String, null if null array input
*/
public static String join(Object[] array, String separator, int startIndex, int endIndex)
```

Příklad: StringUtils

PROFINIT
new frontier group

```
public static String join(Object[] array, char separator, int startIndex, int endIndex) {
    if (array == null) {
        return null;
    }
    int bufSize = (endIndex - startIndex);
    if (bufSize <= 0) {
        return EMPTY;
    }

    bufSize *= ((array[startIndex] == null ? 16 : array[startIndex].toString().length()) + 1);
    StrBuilder buf = new StrBuilder(bufSize);

    for (int i = startIndex; i < endIndex; i++) {
        if (i > startIndex) {
            buf.append(separator);
        }
        if (array[i] != null) {
            buf.append(array[i]);
        }
    }
    return buf.toString();
}
```

Název metody

PROFINIT
new frontier group

- To první a často jediné, co z vaší metody bude většina lidí číst – věnujte mu pozornost
- Měl by být stručný a přitom dostatečně popisovat, co metoda dělá
- Pokud se vám nedaří vhodný název vymyslet, může to být špatným návrhem metody
  - Metoda dělá více různých věcí (měla by dělat vždy jen jeden svůj úkol)
  - Nemáte vlastně jasno, co má metoda dělat
  - …
- Pokud jste při vymýšlení jména přišli na špatný návrh metody, ušetřili jste si dost času, který byste jinak strávili později přepisováním kódu

- Kvíz: co dělá tato metoda?
  `dontStoreDataToCache()`

Název metody

PROFINIT
new frontier group

```
isSaveBasicDataAndTemplateAttributeValuesDeactivateDocumentReturnPossible()
```

Jména a typy parametrů

PROFINIT
new frontier group

- Jméno by mělo popisovat, co daný objekt reprezentuje – nejen sám o sobě, ale i v kontextu použití parametru:

```
void merge(Graph graph1, Graph graph2)
```

vs.

```
void merge(Graph inputGraph, Graph outputGraph)
```

- U typovaných jazyků využívejte datové typy
  - Toto by měla být naprostá samozřejmost
  - Speciálně v Javě se ale často setkávám zejména s ignorováním enums (místo enumu se často používají booleany nebo soustavy konstant)
  - Zdá se, že velmi populární je místo „strongly typed“ používat „stringly typed“ přístup:
    - Text: `"abc"`
    - Číslo: `"42"`
    - Objekt obsahující jednu textovou a dvě číselné položky: `"abc_42_666"`
    - Přístup k první číselné položce objektu: `myObjectString.split("_")[1]`

Co nepovažuji za užití datových typů

PROFINIT
new frontier group

- `Map<Vertex, Set<Boolean>>`
- `List<List<List<String>>>`
- `Map<List<Object>, HashSet<Pair<Integer, Integer>>>`

Parametry

PROFINIT
new frontier group

- Metoda s příliš velkým počtem parametrů je nepřehledná

```
protected PdfPTable getPrijemcePlneniTable(float documentWidth, String
   nazevLeasingovky, String cisloLeasingoveSmlouvy, String smluvniServis,
   String prijemceCisloUctu, String prijemceJmeno, String
   prijemceUliceCP, String prijemceObecPSC, boolean
   pojisteniJindeExistuje, String pojisteniJindeNazev, Boolean
   dphAnoNeNull, String zpusobUhradyKey, String datum, boolean isDPH,
   String sTextPodPodpisem)
```

- Parametry typu boolean nejsou příliš vhodné
  - Nutno v názvu parametru jasně deklarovat, co znamená true a co false
    - Jméno parametru ale nemusí být vždy dostupné (např. v Javě, mám-li knihovnu bez javadoc a zkompilovanou bez jmen parametrů, vidím jen typ bez jména)
  - Při volání metody není na první pohled význam boolean parametru obvykle zřejmý
    - `newTable(tableName, parentDatabase, true)`
    - `newTable(tableName, parentDatabase, TableType.GLOBAL_TEMPORARY)`

Javadoc

PROFINIT
new frontier group

- Dohodněte se na jazyku a dodržujte ho
  - V angličtině může být problém se vyjádřit (zejména odborné pojmy)
  - Komentáře v češtině vyžadují neustále přepínat klávesnici a jazyk „v hlavě“
  - Komentáře v „cestine“ nevypadají dobře
- Dokumentujte chování vůči null (všude tam, kde to není očividné)
  - U atributů objektu / třídy – zda může někdy být null a co to znamená
  - U parametrů metod – zda může být null a co se v tom případě stane
  - U návratových hodnot – zda a v jaké situaci může metoda vrátit null
- U kolekcí a polí vnímejte rozdíl mezi prázdnou kolekcí a null
  - Většinou dává lepší smysl prázdná kolekce („seznam nalezených prvků je prázdný“) než null („seznam nalezených prvků neexistuje“)
  - Nenechte se vést leností nebo dojmem větší efektivity / úspory paměti při použití null (Java: `Collections.emptyList()`)

Javadoc

PROFINIT
new frontier group

```
/**
 * Metoda cislo zaznamu milestone v cache nebo -1
 * @param firstRow
 * @param lastRow
 * @param extraRecords
 * @return
 * @throws ExpiredDataException
 */
public CachedData getTimeMilestoneCachedData(..)
```

Kód

PROFINIT
new frontier group

- Dodržujte konvence daného jazyka (jména, formátování, …)
- Orientace na čtenáře – kód bude čten víckrát, píšete jej jednou
  - Lenost zde není dostatečným argumentem
- Jména proměnných (a všeho ostatního) volte dostatečně srozumitelná
  - Příklad z jedné revize:
    v cele třídě mnoho nevhodných názvů proměnných – „bi“, „is“, „bos“, „asr“, „s“, „m“, „baos“, „dos“ a můj favorit, pole s názvem „array“
  - Kvíz: co ošetřuje následující podmínka?

```
if(lv >= 0 && df >= 0 && this.hasRows() &&
    df < this.getRows().length) { … }
```

  - Všeho s mírou – obecně čím větší rozsah platnosti, tím důležitější je jasné jméno
  - Srozumitelnost != délka – např. význam proměnné „i“ je obecně známý
- Nepoužívejte jednu proměnnou ke dvěma účelům
- Ternární operátor je efektní, ale leckdy poněkud nečitelný
  - Kvíz: co vrátí následující výraz?

```
(0 < 1) ? 2 : 3 + 4
```

A Roguelike in less than 512 Bytes

PROFINIT
new frontier group

```
#include<stdlib.h>
#define F(n)for(j=0;j<n;j++)
#define r rand()
int main(){int x,s=46,n,i,j,z=77,l[z];char m[z*s],h[z];initscr();raw();F(z*s)j[
m]=35;F(s)for(j[l]=i=(r%4+3)*z+(n=r%17*z+r%s+z);n<=i;n+=z)for(x=n;x<=n+j/2;m[++
x]=s);F(9)l[j][m]=z,j[h]=2;m[*l]=64;*h=5;l[j][m]=62;F(z){x=n=l[i++,i%=9];if(i)!
i[h]||*l^(n+=r%3+r%3*z+~z)||--*h?0:abort();else{F(25)mvaddnstr(j,i,m+j*z,z);j=s
-getch();m[n+=j/3*z-j%3+153]^62||main();F(9)l[j+1]^n||--h[j+1]||n[m]--;}n[m]^s
||(m[l[i]=n]=x[m],x[m]=s);}}
```

This weighs in at a hefty 493 bytes of C source code. I call it the "Monster Caves".

Note that the game doesn't have enough bytes to remember (and print out) your kill total, so you'll have to do that yourself.

If you get stuck in a disconnected part of a level, try pressing some other keys... you may just be able to teleport out.

http://locklessinc.com/articles/512byte_roguelike/

Programming by Coincidence

Programming by Coincidence

PROFINIT
new frontier group

```
if (typZadosti == ZALOZENI_SMLOUVY) {
    zalozSmlouvu();
} else {
    zrusSmlouvu();
}
```

Programming by Coincidence

PROFINIT
new frontier group

```
if (typZadosti == ZALOZENI_SMLOUVY) {
    zalozSmlouvu();
} else if (typZadosti == ZRUSENI_SMLOUVY) {
    zrusSmlouvu();
}
```

Programming by Coincidence

PROFINIT
new frontier group

```
if (typZadosti == ZALOZENI_SMLOUVY) {
    zalozSmlouvu();
} else if (typZadosti == ZRUSENI_SMLOUVY) {
    zrusSmlouvu();
} else {
    throw new IllegalArgumentException(
                "Nepodporovany typ zadosti: " + typZadosti);
}
```

Programming by Coincidence

PROFINIT
new frontier group

```
if (typZadosti == ZALOZENI_SMLOUVY) {
    zalozSmlouvu();
} else if (typZadosti == ZRUSENI_SMLOUVY) {
    zrusSmlouvu();
}
// else do nothing
```

Programming by Coincidence 2

PROFINIT
new frontier group

```
int pocetCelychMesicuDoKonceKvartalu = 0;
switch(mesicVRoce%3) {
    case 0: pocetCelychMesicuDoKonceKvartalu = 0;break;
    case 1: pocetCelychMesicuDoKonceKvartalu = 2;break;
    case 2: pocetCelychMesicuDoKonceKvartalu = 1;break;
}
```

Programming by Coincidence 2

PROFINIT
new frontier group

```
int pocetCelychMesicuDoKonceKvartalu = 0;
switch(mesicVRoce%3) {
    case 0: pocetCelychMesicuDoKonceKvartalu = 0;break;
    case 1: pocetCelychMesicuDoKonceKvartalu = 2;break;
    case 2: pocetCelychMesicuDoKonceKvartalu = 1;break;
    default: throw new IllegalStateException(
                        "Nastala situace, ke ktere nemuze dojit");
}
```

Je tento konstruktor prázdný záměrně?

PROFINIT
new frontier group

```
protected MyObject() {

};
```

Je tento konstruktor prázdný záměrně?

PROFINIT
new frontier group

```
protected MyObject() {
    /* empty */
};
```

Dřív to fungovalo...

PROFINIT
new frontier group

```
data.id = new EntityId(this.data.size());
```

Perličky

Prostě to zařídíme

PROFINIT
new frontier group

```
// Vratit proste null.
return null;
```

Kdy se zobrazí nabídka úvěrů?

PROFINIT
new frontier group

```
/**
 * Method returns true if credit lines should be shown.
 *
 * @return true if credit lines should be shown otherwise false
 */
public boolean isShowCreditLines() {
    return
        // Showing depends of dynamic property and if user is in pilot mode (show only in pilot to
        // users in pilot mode if constant is true, if false, show to all)
        ((ebankingUser.getPilot() == null) || (!EbankingConstants.CREDIT_LINES_FUNCTIONS_PILOT) ||
(Boolean.TRUE.equals(ebankingUser.getPilot()) && EbankingConstants.CREDIT_LINES_FUNCTIONS_PILOT))
                // and (home tab is shown and there are no campaigns)
                && ((menuBean.isHomeTab() && !bcs.isShowCampaigns())
                // or ((we are on loans tab (TAB_LOANS) on specific menu item (MNU_LST_CRE_MORA_GET)
                // or on eshop (TAB_ESHOP)) and client has no approved application)
                || (((menuBean.isWantedTab(MenuNameS24.TAB_LOANS.name())
                // THU: tento kod je zakomentovan, protoze nefuguje nastavovani actualMenuItem spravne
                                /*&& (menuBean.getActualMenuItem() != null) &&
                                      menuBean.getActualMenuItem().name().toUpperCase().equals(
                                          MenuNameS24.MNU_LST_CRE_MORA_GET.name())*/
        )
                || menuBean.isWantedTab(MenuNameS24.TAB_ESHOP.name())) &&
                                                    bcs.getApprovedApplicationID() == null))
                // and there exist some credit lines
                && !isItemListEmpty();
}
```

Map není nikdy dost

PROFINIT
new frontier group

```
public interface IValidator {

   public void validate(
        Object[] objects,
        Map<Object, Object> globalParams,
        Map<Object, Object> localParams,
        Map<Object, Object> rowParams,
        Locale locale);

}
```

DB procedura

PROFINIT
new frontier group

```
...
**  Nazev ulozene procedury:  esipo_s_prehl_rotv
**
**  Funcionalita:  Procedura slouzi
**
**  Vstupni parametry:
...
```

DB access layer

PROFINIT
new frontier group

```
if ((@zadost = null and @smlouva = null) or
    (@zadost = null and @smlouva <> null))
  ...
```

Trocha logiky

PROFINIT
new frontier group

```
public void setPlatnostDoOdvolani(boolean platnostDoOdvolani) {
    if (platnostKontaktniAdresy == null) {
        platnostKontaktniAdresy = new IntervalOdDo();
    }
    else {
        platnostKontaktniAdresy = new
   IntervalOdDo(platnostKontaktniAdresy.getZacatek(), null);
    }
}
```

Velmi defenzivní

PROFINIT
new frontier group

```
ConfigFileReader oConfigFileReader =
    new ConfigFileReader(getConfFilePathSecurity());
if (oConfigFileReader == null) {
  ...
}
```

A na závěr to nejhorší...

PROFINIT
new frontier group

```
private Object[][] s24ReportsGroup = {
        {"9001", "GROUPSTACK", new int[]{11, 12, 13, 14, 15, 828}, "111112"},
        {"9002", "GROUP", new int[]{30, 822}, "dd"},
        {"9003", "GROUP", new int[]{40, 10, 824, 826}, "d0d0"},
        {"9004", "GROUP", new int[]{200, 205}, "dl"}                                    //Ac
};

private Object[][] b24ReportsGroup = {
        {"9005", "GROUPSTACK", new int[]{320, 321, 820}, "112"},
        {"9006", "GROUP", new int[]{300, 830}, "dd"},
        {"9007", "GROUP", new int[]{310, 311, 832, 834}, "d0d0"},
        {"9019", "GROUP", new int[]{310, 311}, "dl"},                                   //P
        {"9030", "GROUP", new int[]{870}, "d"},                                          //
        {"812", "GROUP", new int[]{812, 813, 814}, "ddd"}
};

private Object[][] uacReportsGroup = {
        {"9008", "GROUP", new int[]{210, 211, 212, 213}, "dddd"},                       //
        {"9009", "GROUP", new int[]{50}, "d"},                                         //Po
        {"9010", "GROUP", new int[]{60}, "d"},                                         //Po
        {"9011", "GROUP", new int[]{530, 531, 532}, "ddd"},                             //P
        {"9012", "ROWS", new int[]{540, 541, 542, 543}, "llll"},                        //
        {"9013", "GROUP", new int[]{221}, "ddd"},                          //Celkov po
        {"9014", "GROUP", new int[]{230}, "d"},                                        //Po
        {"9015", "GROUP", new int[]{560, 561, 562, 563, 564, 565, 566}, "ddddddd"},
        {"9025", "GROUP", new int[]{810}, "d"},
        {"9026", "GROUP", new int[]{811}, "d"}
};

private Object[][] gsmReportsGroup = {
        {"9016", "ROWS", new int[]{640, 641, 642, 643, 644, 645, 646, 647}, "llllllll"},
        {"9017", "GROUP", new int[]{750, 751, 752}, "ddd"}                              //S
};
```

Goodies

Templates, checklists, literatura

PROFINIT
new frontier group

Materiály SWENG - Construction

ČLÁNKY

- "Best Practices" Columns by Steve McConnell

KNIHY

- Steve McConnell, Code Complete, 2nd Edition. Redmond, Wa.: Microsoft Press, 2004

CHECKLISTS

- CxCheck_Conditionals.txt
- CxCheck_ConstructingRoutine.txt
- CxCheck_ControlStructureIssues.txt
- CxCheck_DataCreation.txt
- CxCheck_DocumentingCode.txt
- CxCheck_HighQualityModules.txt
- CxCheck_HighQualityRoutines.txt
- CxCheck_Layout.txt
- CxCheck_Loops.txt
- CxCheck_NamingData.txt
- CxCheck_OrganizingCode.txt
- CxCheck_UnusualControlStructures.txt
- CxCheck_UsingData.txt

Všechny odkazované materiály jsou poskytnuty výhradně za účelem výuky softwarového inženýrství.