

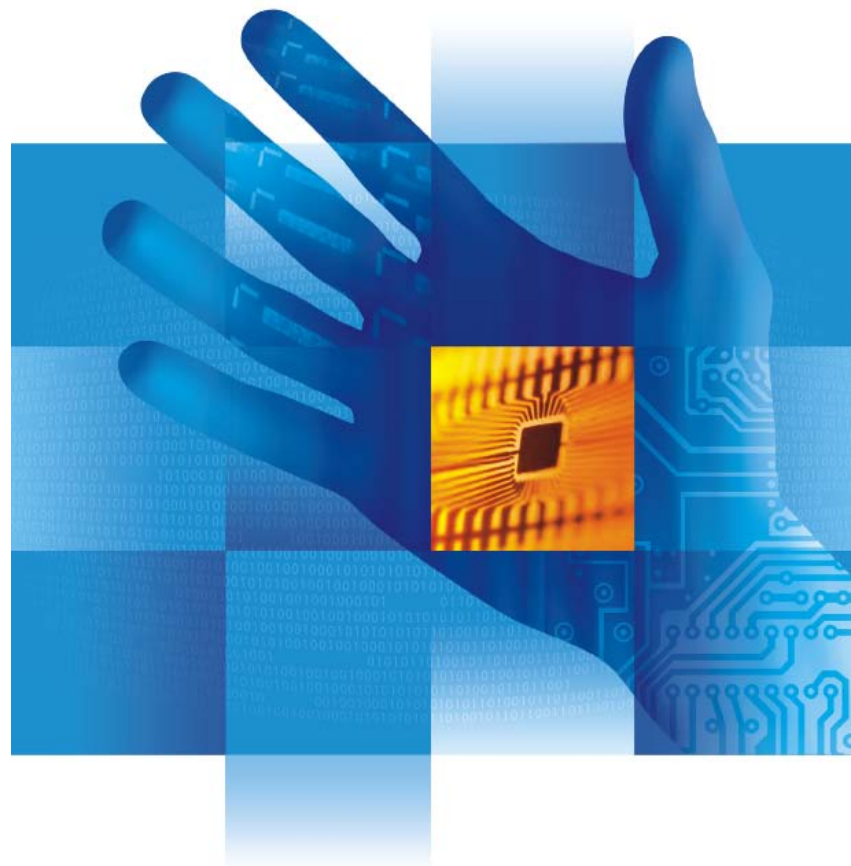
Základy programování

Zdrojový kód, dokumentace, týmová práce

doc. RNDr. Petr Šaloun, Ph.D.

VŠB-TUO, FEI

(přednáška připravena z podkladů
Ing. Michala Radeckého)



Algoritmus

- Algoritmus
 - Postup popisující konkrétní jednotlivé kroky vedoucí k vyřešení dané úlohy.
 - Vlastnosti algoritmů
 - Hromadnost a univerzálnost
 - Jednoznačnost (determinovanost)
 - Konečnost
 - Rezultativnost (rozhodnost)
 - Správnost
 - Algoritmická složitost (prostorová, časová)
 - Druhy algoritmů (rekurzivní, hladové, genetické, heuristické, apod.)



Zápis algoritmů

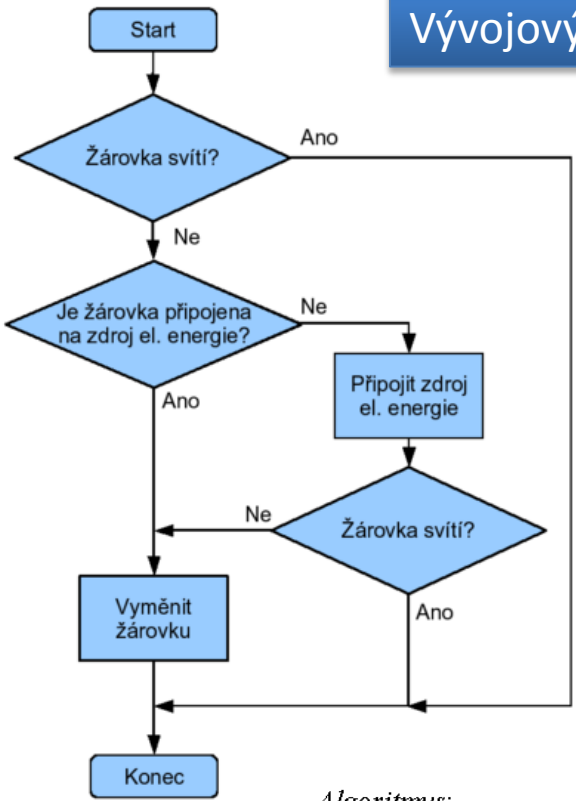
- Grafická forma
 - Vývojové diagramy
 - Strukturogramy
 - UML diagramy (aktivitní diagramy)
- Textová forma
 - Přirozený jazyk
 - Pseudokód
 - Formální jazyk
 - Programovací jazyk (taky formální jazyk)





Grafické vyjádření algoritmů

Vývojový diagram



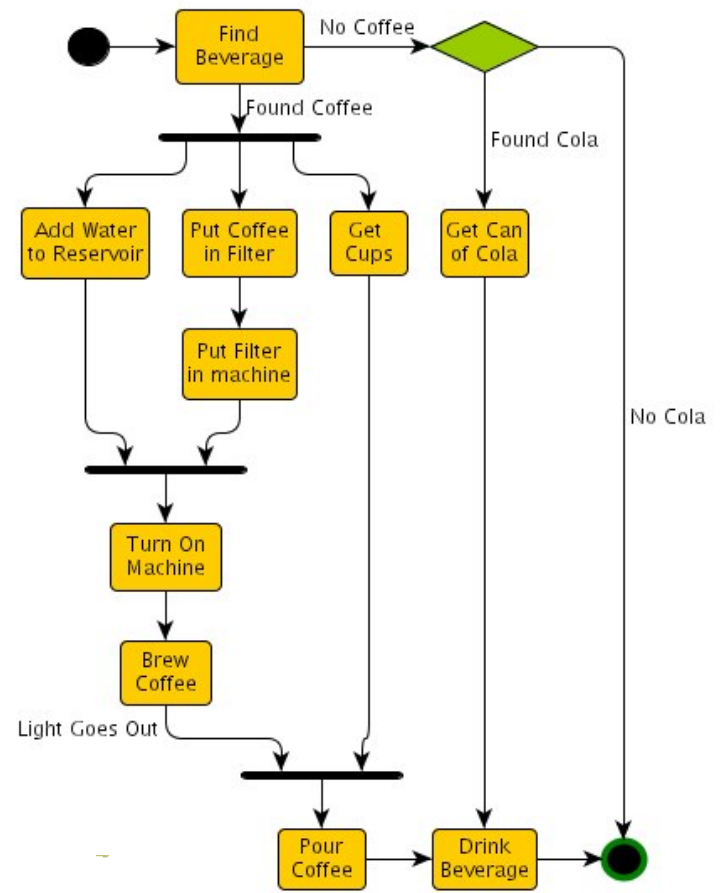
Algoritmus:

Jdi na začátek řádku	
N = 50	
N > 0 ?	
-	+
	<i>zasad' brambor</i>
	<i>popojdi o 25 cm</i>
	<i>N = N - 1</i>
uhrab pole	
jdi domů	

cyklus

Počet brambor zde hraje úlohu počítadla, které udává počet cyklů, než řešení postoupí dále v přímé větvi

Strukturogram



UML aktivní diagram

Zásady zápisu ZK

- Dodržování pravidel a zásad umožňuje tvorbu *snadno udržitelného a udržitelného kódu*
- Tvorba programu *není jednorázová záležitost*
 - nové přístupy pro řešení konkrétních úloh či problémů
 - nové znalosti a schopnosti programátora
 - nové požadavky na funkcionalitu a rozšiřování
 - noví programátoři
- Čitelnost
 - vlastního zdrojového kódu
 - algoritmů a jazykových konstrukcí
- Konvence a syntaxe
- Konvence v závislosti na programovacím jazyce
- Dodržování jednotného programátorského stylu

Zásady zápisu ZK

- Jeden příkaz na jeden řádek

```
if (i==10) a++;b--; System.out.println(b); //Spatne
if (i==10) a++; //Dobre
b--;
System.out.println(b);
```

- Odsazování řádků

```
if (i!=j){ //Spatne
a=i*j
b--;
System.out.println(b)}
if (i!=j) //Dobre
{
    a=i*j
    b--;
    System.out.println(b)
}
```



Zásady zápisu ZK

- Vynechávání řádku mezi logickými celky

```
public Simplify()           //Zacatek jedne metody
{
    X=new ArrayList<Double>();
    Y=new ArrayList<Double>();
    ...
}                           //Konec jedne metody
                             //Prazdny radek
public void ClearAll()     //Zacatek druhe metody
{
    X.clear();
    Y.clear();
}                           //Konec druhe metody
```

- Rozlišování velkých a malých písmen

```
double a;
double A; //Jedna se o jine promenne
sin(a);   //Spravne
Sin(a);   //Spatne
```

Zásady zápisu ZK

- Komentování kódu (co, jak, proč, kdy,

Kýms)

```
if (v==1) //Vrcholy V1,V2 trojuhelniku
{
    //Pocatecni bod ve stredu strany
    x1=(x_point.get(v2)+x_point.get(v3))/2;
    y1=(y_point.get(v2)+y_point.get(v3))/2;

    //Koncovy bod ve stredu strany
    x2=(x_point.get(v1)+x_point.get(v3))/2;
    y2=(y_point.get(v1)+y_point.get(v3))/2;
```

- Délka řádku s ohledem na šířku obrazovky a zalamování dlouhých řádků na logickém místě
- Rozumná délka jednotlivých logických bloků (metod, podprogramů, apod.)

Zásady zápisu ZK

- Mezery ve vztahu k okolí

```
public void AddPoint( double x,double y,double z ) //Spatne
{
    x_point.add(x);
    y_point.add(y);
    z_point.add(z);
}
public void AddPoint(double x, double y, double z) //Dobre
{
    x_point.add(x);
    y_point.add(y);
    z_point.add(z);
```

- Snažit se pro vše používat jen jeden jazyk (čeština – nepoužívat diakritiku, angličtina)
- Nepoužívat příliš dlouhé identifikátory a dodržovat jednotná pravidla pro tvorbu názvů a identifikátorů

Zásady zápisu ZK

- Používat výstižné a odlišitelné názvy identifikátorů a metod (bez kolize)

```
//Spatne
```

```
class A  
double k, l, m;  
int x(double y);
```

```
//Spravne
```

```
class Graphics  
double scale, rotation, distance  
int degrees(double angle)
```

- Podmínky se snažit zapisovat v pozitivní formě

```
if (!(a<b)) //Spatne  
if (a>=b) //Dobre
```

- Nepoužívat příkazy skoku



Zásady zápisu ZK

- Komentáře psát tak a tam, aby byla zajištěna pochopitelnost kódu i jinému člověku příp. za delší dobu
- V kódu přímo komentovat významná místa (složité algoritmy, apod.)
- Komentovat proměnné a jejich význam
- Komentovat hlavičky metod a metody
- Komentovat změny v kódu a místa s upozorněním



Jmenné konvence v C/C++

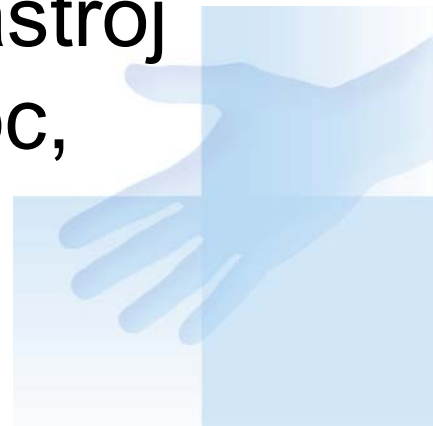
- Podstatná, přídavná jména a slovesa
- Velká a malá písmena
 - Pascal case – BeginInvoke, GetElementValue, GetHtml
 - Camel case – beginInvoke, getElementValue, getHtml
 - Underscore case – begin_invoke, get_element_value
- Vlastní datové typy, třídy – PascalCase (bez prefixu)
- Vnější proměnné, atributy - PascalCase
- Vnitřní proměnné, parametry – camelCase
- Konstanty – UPPER_UNDERSCORE_CASE
- Metody – camelCase()
- Namespace – model::lowercase
- Kolekce (pole, atd.) – množné číslo – points, element

<http://geosoft.no/development/cppstyle.html>



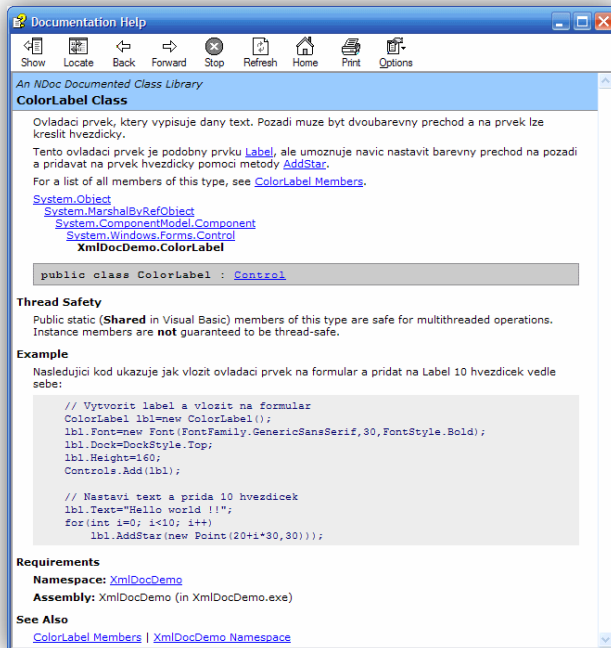
Dokumentace

- **Nejlepší dokumentace je čitelně a kvalitně napsaný kód a jeho komentáře**
- V některých jazycích je možné používat specifické způsoby tvorby komentářů, které následně umožní jejich automatický převod (pomocí odpovídajících nástrojů) do podoby dokumentace (Java, .NET)
- Obvykle vyžadují transformační nástroj (JavaDoc, XSL transformace, NDoc, Doxygen, apod.)





Komentování v C#



```

/// <summary>
/// Ovladaci prvek, který vypisuje dany text. Pozadi muze
/// byt dvoubarevny prechod a na prvek lze kreslit hvездicky.
/// </summary>
public class ColorLabel : Control
{
    /// <summary>
    /// Pridava hvездicku na pozici urcenou parametrem
    /// <paramref name="position" />
    /// </summary>
    /// <param name="position">Pozice hvездicky v pixelech</param>
    public void AddStar(Point position)
    {
    }
}

```

```

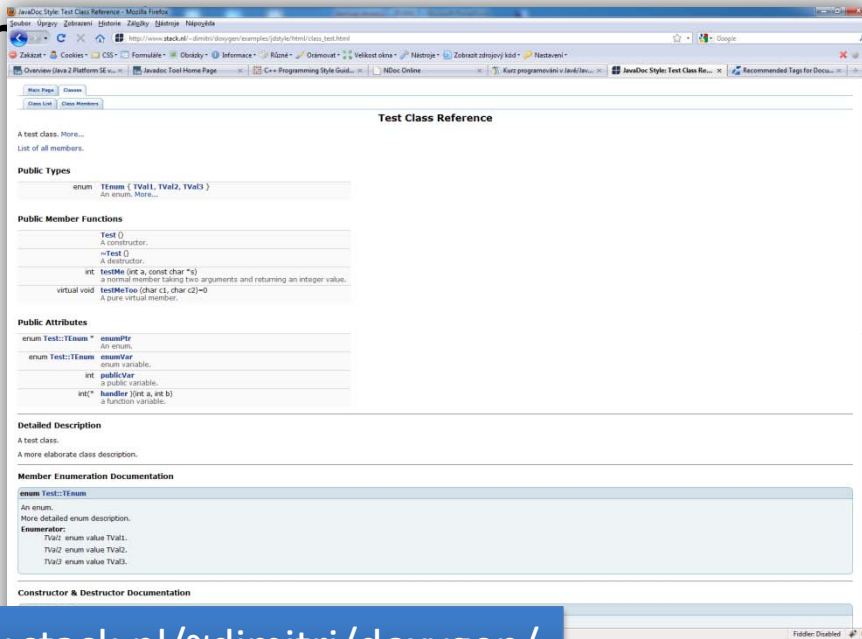
/// <summary>...</summary>
/// <example>
/// Pomoci metody <c>AddStar</c> lze vkladat hvездicky.
/// Nasledujici kod ukazuje jak vlozit hvездicku na nahodne místo.
/// <code>
///     Random rnd=new Random();
///     lbl.AddStar(new Point(rnd.Next(lbl.Width),rnd.Next(lbl.Height)));
/// </code>
/// </example>
public void AddStar(Point position)
{
    ...
}

```

VS 2008 nebo NDoc
<http://ndoc.sourceforge.net/>

Komentování v C/C++

- Pomocí univerzálního nástroje Doxygen
- Má vlastní konvence, ale je možné používat styl JavaDoc `/** **/`
- Podpora celé řady jazyků a výstupních formátů



<http://www.stack.nl/~dimitri/doxygen/>

```
/**
 * A test class. A more elaborate class description.
 */
class Test
{
public:
    /**
     * An enum.
     * More detailed enum description.
     */
    enum TEEnum {
        TVal1, /** enum value TVal1. */
        TVal2, /** enum value TVal2. */
        TVal3 /** enum value TVal3. */
    }
    enumPtr, /** enum pointer. Details. */
    enumVar; /** enum variable. Details. */

    /**
     * A constructor.
     * A more elaborate description of the constructor.
     */
    Test();

    /**
     * A destructor.
     * A more elaborate description of the destructor.
     */
    ~Test();

    /**
     * A normal member taking two arguments and returning an integer value.
     * @param a an integer argument.
     * @param c a constant character pointer.
     */
    int Test(int a, const char *c);

    /**
     * A pure virtual member.
     */
    virtual int Test();

    /**
     * A public variable.
     * Details.
     */
    int publicVar;

    /**
     * A function variable.
     * Details.
     */
    int (*handler)(int a, int b);
};
```


JavaDoc komentáře

```
/**
 * @author „Michal Radecky”
 */
```

```
/*
 * @version ZpProject1.2
 */
```

```
/** @deprecated Tato metoda je zastarala, nahrada {@link #getValue}*/
```

```
/** Další informace <a href={@docRoot}/info.html></a>. */
```

```
/** tato metoda n-tou mocninu
 * @param number cislo, které chceme umoznit
 * @param power vyjadruje cislo mocniny
 * @return vraci cislo na n-tou
 */
public static float getPower(float number, int power)
{
    ...
}
```

```
/** odkaz v textu:
 * Text ... {@see <a href="http://java.sun.com/">Java
 * Sun</a>} ...
 * odkaz za textem:
 * @see java.lang.Object
 */
```

Týmová práce

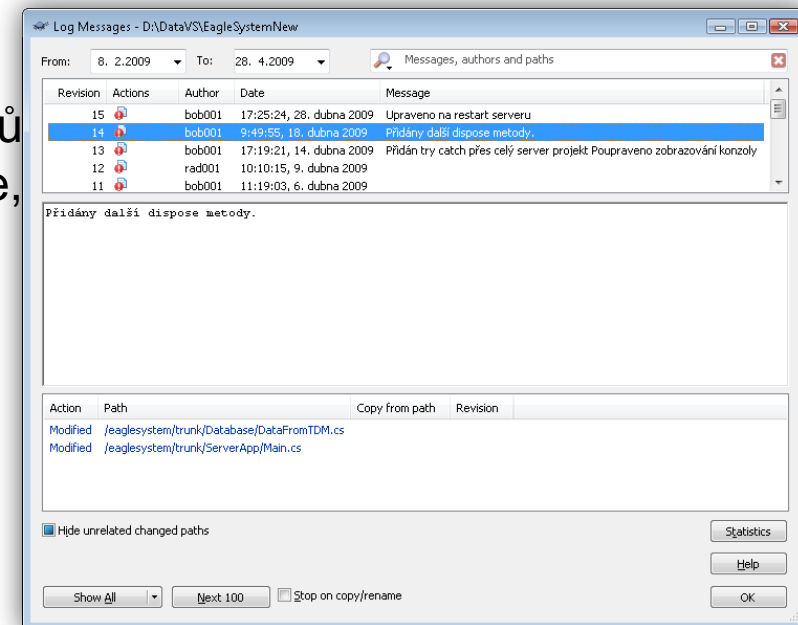
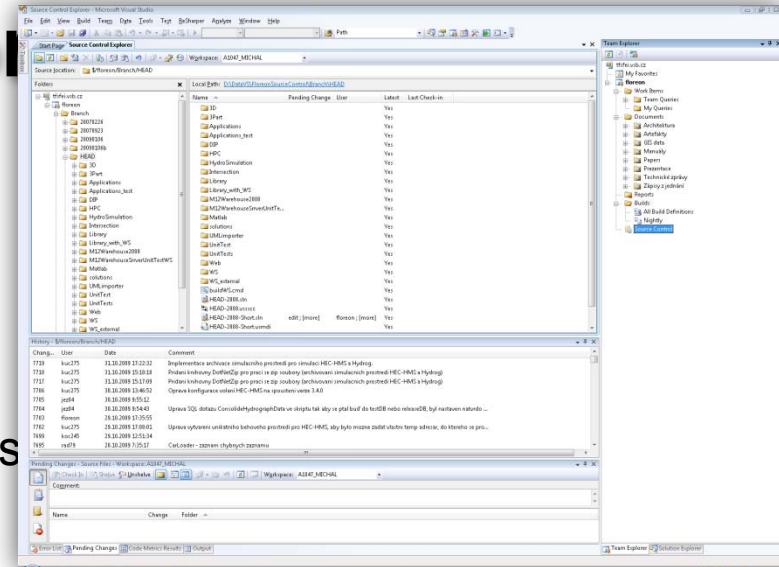
- Přímá komunikace v týmu
 - email, ICQ, Live Messenger, apod.
- Kolektivní komunikace
 - diskusní skupiny, fóra, konferenční hovory, apod.
- Řízení týmové spolupráce a reportování
 - workflow systémy, projektový management, správa chyb, apod.
- Spolupráce při vytváření obsahu
 - SVN, CSV, TFS, Google Docs, Live, apod.
- Dodržování stejných pravidel a konvencí napříč týmem

- Cílem je vytvoření prostředí pro komunikaci a spolupráci jednotlivých subjektů nad jedním projektem/úkolem/aplikací, atd.
- Často je využití těchto nástrojů efektivní také pro aplikací v „netýmovém“ prostředí



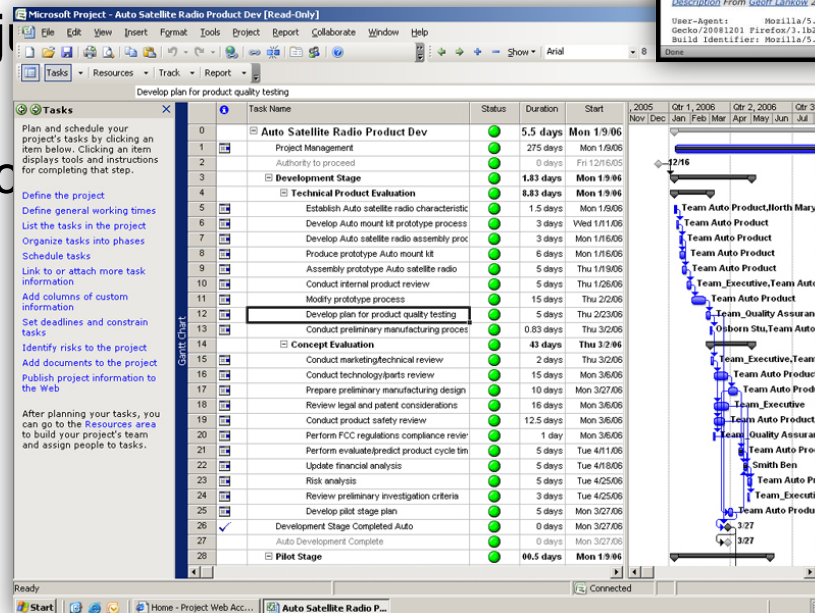
Nástroje pro team work

- Správa zdrojových kódů
 - verzování a historie
 - „mergování“ a správa úprav
 - víceuživatelský přístup
 - nezávislost na platformě a možnost multiplatformního přístupu
 - využívá se tzv. repository
 - nástroje: TFS, VSS (SourceSafe), CVS, SVN (<http://tortoisesvn.net/>, <http://svn1clicksetup.tigris.org/>)
 - existují volně dostupné systémy pro hostování zdrojových kódů projektů (Sourceforge, CodePlex, Google Code,



Nástroje pro team work

- Správa projektů a úkolů
 - správa členů vývojového týmu
 - správa úkolů a chyb pro jednotlivé členy skupiny v týmu (bug tracking, issue tracking)
 - podpora testování
 - reportování činností v týmu a vyhodnocování projektů
 - správa a toky činností a dokumentů
 - správa zdrojů
 - nástroje: Office Project, OpenProj, BugZilla



The screenshot shows a Bugzilla bug report for Bug 480443. The title is "URLbar should only show address tooltip when overflowing". The report includes the following details:

- Status:** NEW
- Whiteboard:** [polish-easy][polish-interactive]
- Keywords:** (empty)
- Product:** Firefox
- Component:** Location Bar and Autocomplete
- Version:** Trunk
- Platform:** All
- Importance:** minor
- Target Milestone:** (empty)
- Assigned To:** Nobody: OK to take it and work on it
- QA Contact:** location_bar@firefox.bugs
- URL:** (empty)
- Depends on:** (empty)
- Blocks:** (empty)
- Reported:** 2009-02-26 16:16 PST by Geoff Lankow
- Modified:** 2009-02-26 16:28 PST (history)
- CC List:** Add me to CC list, 2 users (edit): sdwlsh@foreurnderdesigns.com, tyler@christianlink.us
- Flags:** blocking-firefox-3.1, wanted-firefox-3.1, blocking-1.9.0.7, blocking-1.9.0.8, wanted-1.9.0.x, blocking-1.8.1.next, wanted-1.8.1.x, blocking-1.8.0.next, wanted-1.8.0.x, in-Itmus, in-testsuite
- Attachments:** patch (2.84 kb, patch) 2009-02-26 16:19 PST: Geoff Lankow, sdwlsh: review? (dietrich), Details | Diff

