Organisational Aspects of Software Development

Pedro Contreras

Department of Computer Science
Royal Holloway, University of London

January 29, 2008

Introduction

Creating software is a complex task.

- Organising tasks
- Managing people
- Maintaining code
- Controlling the development process
- Ensuring quality
- Estimating economical cost
- etc.

Introduction

We present some useful tools together with a few tips that can help when creating software.

This presentation is separated into two main sections.

- **Development process and collaborative work** deals with tools and techniques to improve communication among Virtual Organisations.
- **Software Engineering Tool** deals with tools and standards to make coding faster and understandable for other programmers.

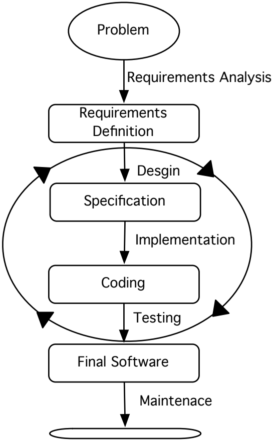
Virtual Organisation

Virtual Organisation (or community) comprises a set of independent individuals that share resources and skills to achieve its mission / goal, but that is not limited to alliance for profit enterprises.
Interaction among members of the virtual organization is mainly done through computer networks.

For example a group of people sharing common interest can organise themselves by creating a virtual community through a web portal.

Software Development Process

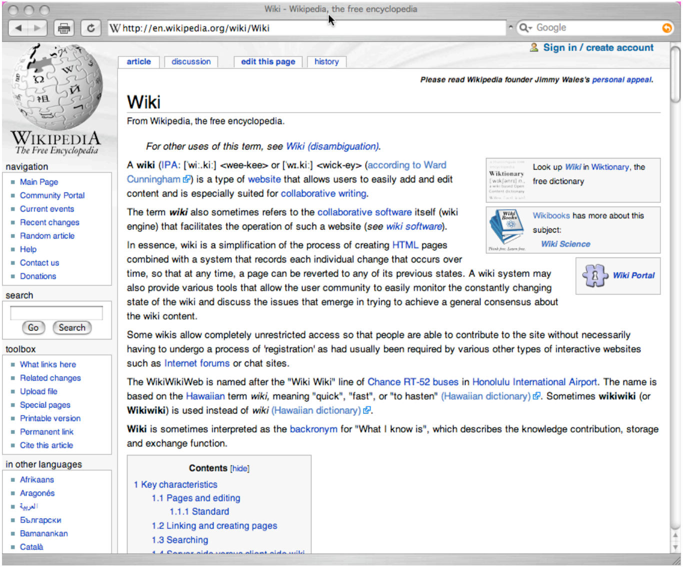
From the software engineering point of view, a virtual organisation will tackle problems related with the software development process.



Collaborative Work and Tools

- Wiki
- Internet Forum
- Blogs
- News
- e-mail (e-mail lists)
- Chat & instant messaging programs (e.g. MSN)
- Others: telephone, meeting, video conference.

Wiki example



Wiki - Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Wiki

Google

Sign in / create account

WIKIPEDIA
The Free Encyclopedia

article | discussion | edit this page | history

navigation

- Main Page
- Community Portal
- Current events
- Recent changes
- Random article
- Help
- Contact us
- Donations

search

Go | Search

toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this article

in other languages

- Afrikaans
- Aragonés
- العربية
- Български
- Bamanankan
- Català

Please read Wikipedia founder Jimmy Wales's personal appeal.

Wiki

From Wikipedia, the free encyclopedia.

For other uses of this term, see Wiki (disambiguation).

Look up *Wiki* in Wiktionary, the free dictionary

Wikibooks has more about this subject:
Wiki Science

Wiki Portal

A **wiki** (IPA: [ˈwiː.kiː] <wee-kee> or [ˈwɪ.ki] <wick-ey> (according to Ward Cunningham)) is a type of website that allows users to easily add and edit content and is especially suited for collaborative writing.

The term ***wiki*** also sometimes refers to the collaborative software itself (wiki engine) that facilitates the operation of such a website (*see wiki software*).

In essence, wiki is a simplification of the process of creating HTML pages combined with a system that records each individual change that occurs over time, so that at any time, a page can be reverted to any of its previous states. A wiki system may also provide various tools that allow the user community to easily monitor the constantly changing state of the wiki and discuss the issues that emerge in trying to achieve a general consensus about the wiki content.

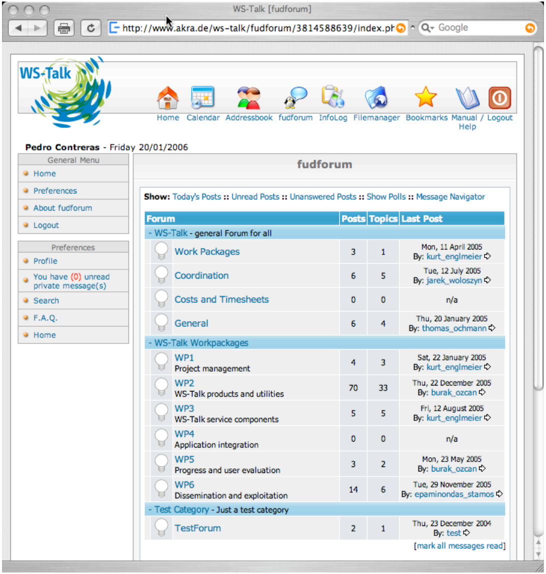
Some wikis allow completely unrestricted access so that people are able to contribute to the site without necessarily having to undergo a process of 'registration' as had usually been required by various other types of interactive websites such as Internet forums or chat sites.

The WikiWikiWeb is named after the "Wiki Wiki" line of Chance RT-52 buses in Honolulu International Airport. The name is based on the Hawaiian term *wiki*, meaning "quick", "fast", or "to hasten" (Hawaiian dictionary). Sometimes **wikiwiki** (or **Wikiwiki**) is used instead of *wiki* (Hawaiian dictionary).

Wiki is sometimes interpreted as the backronym for "What I know is", which describes the knowledge contribution, storage and exchange function.

Contents [hide]

Forum example



Pedro Contreras - Friday 20/01/2006

fudforum

Show: Today's Posts :: Unread Posts :: Unanswered Posts :: Show Polls :: Message Navigator

Forum	Posts	Topics	Last Post
- WS-Talk - general Forum for all			
Work Packages	3	1	Mon, 11 April 2005 By: kurt_englmeier
Coordination	6	5	Tue, 12 July 2005 By: jarek_woloszyn
Costs and Timesheets	0	0	n/a
General	6	4	Thu, 20 January 2005 By: thomas_ochmann
- WS-Talk Workpackages			
WP1 Project management	4	3	Sat, 22 January 2005 By: kurt_englmeier
WP2 WS-Talk products and utilities	70	33	Thu, 22 December 2005 By: burak_ozcan
WP3 WS-Talk service components	5	5	Fri, 12 August 2005 By: kurt_englmeier
WP4 Application integration	0	0	n/a
WP5 Progress and user evaluation	3	2	Mon, 23 May 2005 By: burak_ozcan
WP6 Dissemination and exploitation	14	6	Tue, 29 November 2005 By: epaminondas_stamos
- Test Category - Just a test category			
TestForum	2	1	Thu, 23 December 2004 By: test

[mark all messages read]

Software Engineering Tools

Coding standards & Java

In the same way that any language has it own writing standards and protocols, coding in Java has some basic guide lines. This has the propose of making the code more readable and understandable for other developers. The following section describes some simple principles when writing code in Java.

Software Engineering Tools

- Adhere to the style of the original
- Adhere to the principle of least astonishment
- Do it right at first
- Document any deviation

Software Engineering Tools

Java coding conventions

- Formatting
- Naming
- Documentation
- Packaging code

Java coding conventions

- **Formatting:** This includes things such as indentation for block statements, breaking up long lines, and use of white spaces instead of “hard tabs” -what looks perfectly formatted in one environment, can look as complete chaos in another-
- **Naming:** Usually Java Software Development Kit convention from Sun Microsystems are used, which includes some of the following: how to name classes, variables, methods, and constants, and when Capitalise.

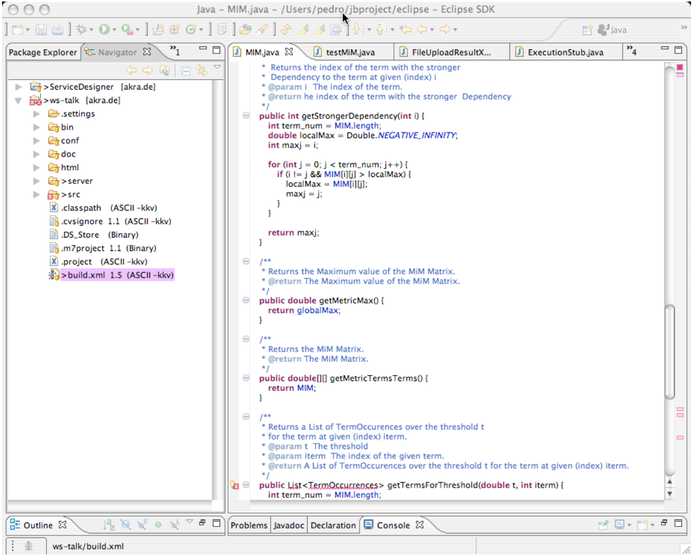
Software Engineering Tools

- **Documentation:** Write documentation for those who will be using the software as well as the people that will maintain it. Documenting Java code for other programmers can be done by means of using comments and using Java documentation package.
- **Packaging code:** Java code organises classes in packages, making it easy to to re-use code. For example, when creating a new package include only related classes, since when using the package it needs to be imported. Therefore if packages are not well organised the software becomes inefficient.

IDE Tools

Interactive Development Environment (IDE) is an integrated system to assist in the software writing; usually such systems includes tools to help with code editing, graphical design, compiling and running programs, and debugging

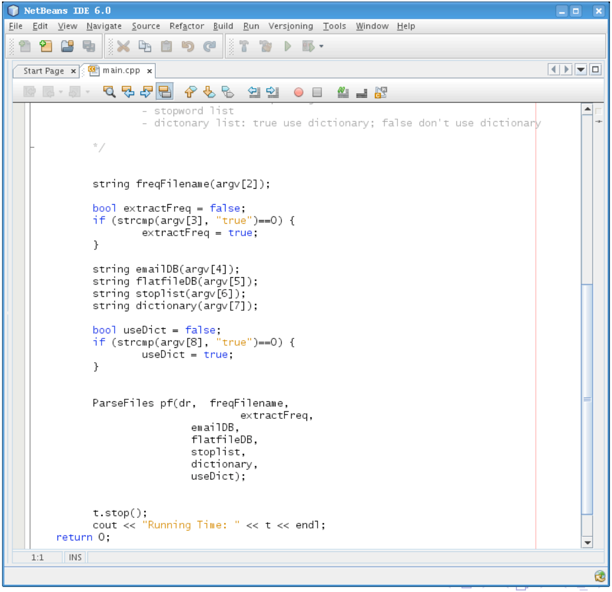
IDE Eclipse example



```
>ServiceDesigner  [akra.de]
>ws-talk  [akra.de]
    .settings
    bin
    conf
    doc
    html
    >server
    >src
    .classpath   (ASCII -kkv)
    .cvsignore  1.1  (ASCII -kkv)
    .DS_Store   (Binary)
    .m7project  1.1  (Binary)
    .project   (ASCII -kkv)
    >build.xml  1.5  (ASCII -kkv)
```

```
 * Returns the index of the term with the stronger
 * Dependency to the term at given (index) i
 * @param i  The index of the term.
 * @return he index of the term with the stronger  Dependency
 */
public int getStrongerDependency(int i) {
    int term_num = MIM.length;
    double localMax = Double.NEGATIVE_INFINITY;
    int maxj = i;

    for (int j = 0; j < term_num; j++) {
        if (i != j && MIM[i][j] > localMax) {
            localMax = MIM[i][j];
            maxj = j;
        }
    }

    return maxj;
}

/**
 * Returns the Maximum value of the MiM Matrix.
 * @return The Maximum value of the MiM Matrix.
 */
public double getMetricMax() {
    return globalMax;
}

/**
 * Returns the MiM Matrix.
 * @return The MiM Matrix.
 */
public double[][] getMetricTermsTerms() {
    return MIM;
}

/**
 * Returns a List of TermOccurences over the threshold t
 * for the term at given (index) iterm.
 * @param t  The threshold
 * @param iterm  The index of the given term.
 * @return A List of TermOccurences over the threshold t for the term at given (index) iterm.
 */
public List<TermOccurrences> getTermsForThreshold(double t, int iterm) {
    int term_num = MIM.length;
```

IDE NetBeans example



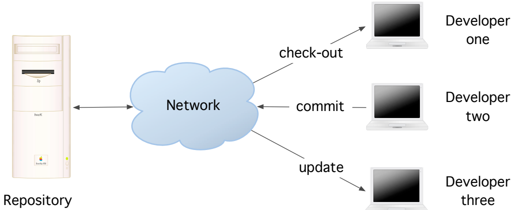
NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Versioning Tools Window Help
Start Page
main.cpp
1:1
INS

```
             - stopword list
             - dictonary list: true use dictionary; false don't use dictionary

        */


        string freqFilename(argv[2]);

        bool extractFreq = false;
        if (strcmp(argv[3], "true")==0) {
                extractFreq = true;
        }

        string emailDB(argv[4]);
        string flatfileDB(argv[5]);
        string stoplist(argv[6]);
        string dictionary(argv[7]);

        bool useDict = false;
        if (strcmp(argv[8], "true")==0) {
                useDict = true;
        }


        ParseFiles pf(dr,  freqFilename,
                                extractFreq,
                        emailDB,
                        flatfileDB,
                        stoplist,
                        dictionary,
                        useDict);


        t.stop();
        cout << "Running Time: " << t << endl;
  return 0;
```

Version Control System

A version Control System is a centralised place where files can be stored, and accessed from any machine with Internet connection. Also it provides a way to store different versions of a document. Then if any version needs to be recovered it can be done easily.

Example: Concurrent Version System



Bug Tracking System - Bugzilla

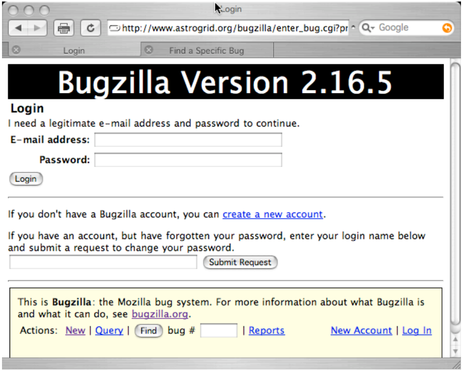
Bugzilla is a free Defect Tracking Systems that allows individual or groups of developers to keep track of outstanding bugs in their software effectively. Bugzilla creates a web-based central repository to maintain a running list of reported defects and their status.

Bug Tracking System - Bugzilla

Some of the following tasks can be carried out with Bugzilla:

- Track bugs and code changes
- Communicate with team-mates
- Submit and review patches
- Manage quality assurance (QA)

Bugzilla engine example



Bugzilla Version 2.16.5

Login

I need a legitimate e-mail address and password to continue.

E-mail address:

Password:

Login

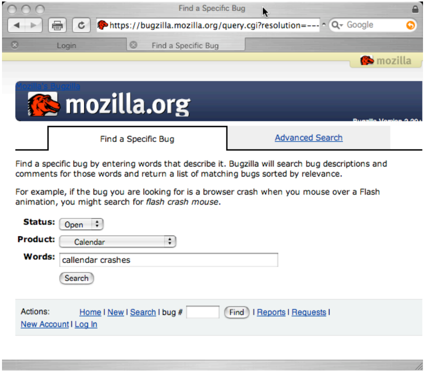
If you don't have a Bugzilla account, you can create a new account.

If you have an account, but have forgotten your password, enter your login name below and submit a request to change your password.

Submit Request

This is **Bugzilla**: the Mozilla bug system. For more information about what Bugzilla is and what it can do, see bugzilla.org.

Actions: New | Query | Find bug # | Reports New Account | Log In



Find a Specific Bug | Advanced Search

Find a specific bug by entering words that describe it. Bugzilla will search bug descriptions and comments for those words and return a list of matching bugs sorted by relevance.

For example, if the bug you are looking for is a browser crash when you mouse over a Flash animation, you might search for *flash crash mouse*.

Status: Open

Product: Calendar

Words: callendar crashes

Search

Actions: Home | New | Search | bug # Find | Reports | Requests | New Account | Log In

Apache Ant

Apache Ant is an open source Java-based build tool

- Ant build files are platform independent
- Ant tracks files dependencies
- Ant Java-based tasks

Apache Ant

Ant build files are platform independent

Ant resolve any platform dependencies such as Operating System (OS) commands (e.g. create directories) and how to format correctly the Java classpath.

Apache Ant

Ant tracks files dependencies

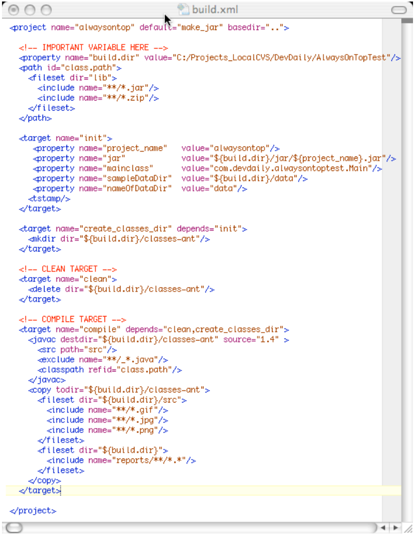
javac compiler is only invoked when source files have been changed. Thus when compiling files just the changes are recompiled and not everything.

Apache Ant

Ant Java-based tasks

Ant includes a wide range of tasks, that are very helpful for customising processes. For example Ant includes task for running JUnit tests. Also Ant can be extends by writing custom tasks.

Apache Ant Example



```xml
<project name="alwaysontop" default="make_jar" basedir="..">

  <!-- IMPORTANT VARIABLE HERE -->
  <property name="build.dir" value="C:/Projects_LocalCVS/DevDaily/AlwaysOnTopTest"/>
  <path id="class.path">
    <fileset dir="lib">
      <include name="**/*.jar"/>
      <include name="**/*.zip"/>
    </fileset>
  </path>

  <target name="init">
     <property name="project_name"   value="alwaysontop"/>
     <property name="jar"            value="${build.dir}/jar/${project_name}.jar"/>
     <property name="mainclass"      value="com.devdaily.alwaysontoptest.Main"/>
     <property name="sampleDataDir"  value="${build.dir}/data"/>
     <property name="nameOfDataDir"  value="data"/>
    <tstamp/>
  </target>

  <target name="create_classes_dir" depends="init">
    <mkdir dir="${build.dir}/classes-ant"/>
  </target>

  <!-- CLEAN TARGET -->
  <target name="clean">
    <delete dir="${build.dir}/classes-ant"/>
  </target>

  <!-- COMPILE TARGET -->
  <target name="compile" depends="clean,create_classes_dir">
    <javac destdir="${build.dir}/classes-ant" source="1.4" >
      <src path="src"/>
      <exclude name="**/_*.java"/>
      <classpath refid="class.path"/>
    </javac>
    <copy todir="${build.dir}/classes-ant">
      <fileset dir="${build.dir}/src">
        <include name="**/*.gif"/>
        <include name="**/*.jpg"/>
        <include name="**/*.png"/>
      </fileset>
      <fileset dir="${build.dir}">
        <include name="reports/**/*.*"/>
      </fileset>
    </copy>
  </target>

</project>
```

Software Testing, Unit Test

There are a lots of different kinds of testing that can be performed on a software project. In some cases testing requires extensive feedback from the end users; other testing form may require a dedicated Quality Assurance teams, or other extensive resources. Unit test, and more specifically JUnit, is a piece of code dedicated to exercise a very small, and specific functionality of the code to be tested

Secure Shell, SSH

SSH is a program for logging and executing commands into a remote machine. It provides secure encrypted communications between two non-trusted hosts over an insecure network. SSH can use different authentication methods such as RSA keys (algorithm for public-key encryption). Though remote log-in is the primary use of SSH, the protocol can be used as a general purpose cryptographic tunnel, capable of copying files, encrypting e-mail connections, and triggering remote execution of programs.

References

- Ant. Apache Ant, 2006. http://ant.apache.org/
- CVS. Concurrent Versions System, 2006. https://www.cvshome.org/
- JUnit. Unit Testing, 2006. http://www.junit.org/
- Sun Microsystems. Java Documentation Specification, 2007. http://java.sun.com/docs/
- SSH. Secure Shell, 2006. http://www.ssh.com/support/downloads/secureshellwks/non-commercial.html
- TWiki. Enterprise Collaboration Platform, 2006. http://www.twiki.org/
- A. Vermeulen, S. Ambler, G. Bumgardner, E. Metz, T. Misfell, J. Shur, and P. Thompson. The Elements of JavaTMStyle. Cambridge University Press, 2000.