**Hosting projektów informatycznych**

Analiza dostępnych systemów online umożliwiających hosting projektów informatycznych.

**1.ASSEMBLA.COM**

**1. Assembla** jest zbiorem zadań na chmurze (cloud-based) i urządzeń do zarządzania kodem dla software developerów. Assembla została stworzona w 2005r. Znajduje się tam ponad 100,000 hostów (komercyjnych i open-source), projektów . Assembla jest używana przez ponad 800,000 użytkowników w 100 krajach. W przypadku korzystania za opłatą ceny (dla odpowiedniej liczby użytkowników) przedstawiają się następująco:

- 10 użytkowników- 75$ miesięcznie lub 750$ rocznie

- 15 użytkowników- 101$ miesięcznie lub 1012$ rocznie

- 20 użytkowników- 135$ miesięcznie lub 1350$ rocznie

- 30 użytkowników- 171$ miesięcznie lub 1710$ rocznie

- 50 użytkowników- 285$ miesięcznie lub 2850$ rocznie

- 70 użytkowników- 381$ miesięcznie lub 3815$ rocznie

- 100 użytkowników- 545$ miesięcznie lub 5450$ rocznie

- 150 użytkowników- 747$ miesięcznie lub 7470$ rocznie

- 200 użytkowników- 996$ miesięcznie lub 9960$ rocznie

- przy liczbie ponad 200 użytkowników cena jest ustalana

Każda licencja zawiera nielimitowany pakiet danych, projektów, repozytoriów.

**2. Funkcjonalność Assemblii:**

**- Wiki**- tworzenie nowych stron, usuwanie stron, edycja stron, podgląd historii zmian, układanie stron w podstrony, spersonalizowane uprawnienia,

**- Aktywność-** zdarzenia, powiadomienia mailowe,

**- Administracja-** możliwość zarządzania: serwisem (darmowa/płatna subskrypcja), narzędziami, wyglądem i bezpieczeństwem,

**- Wiadomości-** możliwość dodawania / usuwania wiadomości – sposób komunikacji w zespole, narzędzie dostosowane do innych narzędzi w serwisie- możliwość dodawania linków do zgłoszeń, stworzenie nowej wiadomości- tytuł, treść, załączniki, możliwość edycji utworzonej wiadomości, możliwość dodawania komentarzy, możliwość określania do kogo zostanie wysłane powiadomienie e-mail (podczas tworzenia wiadomości),

- **Punkty docelowe-** Możliwość utworzenia punktu (milestone’a) w którym coś ma zostać wykonane, przeprowadzone, itp. planowanie pracy w projekcie (daty ukończenia poszczególnych funkcjonalności), możliwość określenia- tytułu, daty ważności, odpowiedzialnego użytkownika, terminarza, opisu, informacje o ilości zgłoszeń i ich stanach dla poszczególnego punktu,

- **Tworzenie raportu-** możliwość dostępu i sprawdzenia raportu**,**

**- Tickety-** status, priorytet (jak ważne jest zgłoszenie), zlecenie – osoba do której przypisane jest zgłoszenie, punkt docelowy – do jakiego punktu (milestona) ma zostać dodane, oszacowanie – jak długo ono potrwa, lista wszystkich ticketów, szczegóły zadania, aktywność, załączniki, powiązane zadania, lista osób obserwujących, raportowanie czasu pracy związanego z obsługą danego zadania,

-**Filtry-** możliwość wyszukiwania zadań spełniających podane kryteria,

-**Terminarz-** pozwala sprawdzić nad jakimi zgłoszeniami aktualnie pracuje zespół,

-**Cardwall-** miejsce służące do sprawdzenia statusu zgłoszeń, pozwala na sprawdzenie jakie zgłoszenia są w jakim stanie, możliwe stany zgłoszenia- New – nowe zgłoszenie, Accepted – przypisane do użytkownika, Test – poprawione, w fazie testów, Invalid – nieprawidłowe (błąd nie jest błędem, nie występuje), Fixed – naprawione, Closed – zamknięte,

-**Metryka-** udostępnia podsumowanie w formie wykresu dla projektu, wykres słupkowy z biletami aktywnymi oraz zakończonymi, tabele z ilością biletów w zależności od- priorytetu, statusu, aktywnych oraz zamkniętych biletów dla danego użytkownika, aktywnych biletów z najwyższym priorytetem, dostępne są statystyki dzienne oraz tygodniowe,

-**Kod źródłowy- SVN-** Browse – przegląd wszystkich plików w repozytorium, Changset - przegląd wszystkich commitów do repozytorium

**2.GITHUB.COM**

**1. GitHub** – hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang. Serwis działa od kwietnia 2008 roku. W kwietniu 2011 ogłoszono, iż GitHub obsługuje 2 miliony repozytoriów. Github udostępnia darmowy hosting programów open source oraz płatne prywatne repozytoria.

**2. Funkcje Github’a:**

* bugtracker,
* forki repozytoriów, czyli kopia w osobnym repozytorium należąca do innego użytkownika,
* *pull requesty* – osoba mająca forka może zgłosić swój kod do złączenia z głównym repozytorium,
* statystyki,
* organizacje zrzeszające programistów pracujących nad repozytoriami,
* web hooks – wywołanie operacji na repozytorium wysyła informację do innego serwisu lub skryptu,
* wiki dla celów dokumentacji
* Branches
* Tags
* Blame
* Ignore list
* Code review

**3. Płatności:**

**-Developer**- 7$ miesięcznie. Zawiera: osobiste konto, nielimitowane publiczne repozytoria, nielimitowane prywatne repozytoria, nielimitowanych kolaboratorów/współpracowników. Darmowe dla studentów w ramach Student Developer Pack

**-Team**- 9$ miesięcznie (od użytkownika). Zawiera: Organization account, , nielimitowane publiczne repozytoria, nielimitowane prywatne repozytoria, Team and user permissions.

**-Business (Hosted on Github.com)**- 21$ miesięcznie (od użytkownika). Zawiera: Organization account, , SAML single sign-on, Access provisioning, 24/5 support with 8-hour response time, 99.95% Uptime SLA, Team sync (Coming 2017)

**-Business (Github Enterprise)**- 21$ miesięcznie (od użytkownika- sprzedawany pakietach dla 10 użytkowników na rok). Zawiera: Multiple organizations, SAML, LDAP, and CAS, Access provisioning, pomoc 24/7 do nagłych/ pilnych problemów, Advanced auditing, Host on your servers, AWS, Azure, or GCP. Możliwość darmowej próby (trial).

**3.CODEPLEX.COM**

**1. CodePlex** - prowadzony przez Microsoft portal opensource'owy, hostujący projekty o otwartym źródle. Codeplex hostuje wszystkie otwartoźródłowe projekty niezależnie od ich związków z technologiami Microsoftu, aczkolwiek wśród projektów zaznacza się swego rodzaju tendencja: jedna trzecia z nich jest objęta licencją [Microsoft Public License](https://pl.wikipedia.org/w/index.php?title=MS-PL&action=edit&redlink=1), a drugą najczęściej używaną licencją jest GPL (około 2000 projektów). Około 70 procent wszystkich programów hostowanych na Codeplex jest napisane w C#, po nim pojawiają się w rankingu, ale znacznie dalej: JavaScript, VisualBasic i C/C++.

**2. Funkcjonalność Codeplex:**

**-** tworzenie projektów

- szukanie projektów

- śledzenie projektów

- przyłączenie się do projektu

- tworzenie i edycja stron wiki

**4.SOURCEFORGE.NET**

**1. SourceForge** – darmowy system zarządzania i kontroli projektów Open Source, skomercjalizowany również do *closed-source license*. Projekty udostępniane są w wersjach dla różnych systemów operacyjnych, najczęściej jednak dla Linuksa i Windowsa.

Główna strona projektu **SourceForge.net** służy programistom do wymiany informacji. Dostępne są również na niej kody źródłowe oraz pliki wykonywalne projektów. W chwili obecnej SourceForge.net jest najpopularniejszym światowym portalem rozwoju oprogramowania Open Source. Posiada ponad 1,8 mln zarejestrowanych użytkowników i ponad 180 tys. projektów. Witryna udostępnia największe repozytorium kodów źródłowych w Internecie. Posiada także bardzo szybkie serwery, co zwiększa popularność serwisu.

Najpopularniejszymi (najwięcej pobrań) programami powstałymi w ramach SourceForge są w kategoriach:

* grafiki komputerowej: pakiet dla systemu Windows programu GIMP
* aplikacji użytkowych: 7-zip, Audacity, Pakiet w:PortableApps.com
* wymiany plików: eMule, Azureus, Ares, BitTorrent, DC++, Shareaza.

SourceForge hostuje także takie popularne projekty, jak MediaWiki, Miranda, Pidgin, phpBB czy phpMyAdmin.

Właścicielem portalu SourceForge.net jest firma GeekNet (dawniej VA Software).

**2. Możliwości Sourceforge.net:**

Niezarejestrowany użytkownik ma możliwość wyszukiwania projektów, oraz pobierania plików lub kodu źródłowego. Po rejestracji otrzymuje możliwość dołączenia do istniejącego projektu lub stworzenia nowego, co niesie ze sobą takie korzyści jak:

* Serwer SVN, Git i Mercurial
* Forum dyskusyjne
* System newsów
* Możliwość dodawania zrzutów ekranowych
* Hosting plików
* Łatwa instalacja skryptów takich jak MediaWiki, phpBB
* Miejsce na serwerze WWW
* Baza MySQL

**5.CODE.GOOGLE.COM**

**1. Google Code** – witryna Google z narzędziami programistycznymi, API i zasobami technicznymi. Strona zawiera dokumentacje o narzędziach Google oraz API Google, jak również grupy dyskusyjne i blogi dla programistów korzystających z produktów Google.

W zasobach serwisu istnieją API dla niemal wszystkich popularnych produktów Google, takich jak np. Google Maps, YouTube, Google Apps, a nawet Google Wave.

Na stronie znajdują się także różnorodne produkty i narzędzia programistyczne przeznaczone dla programistów. Google App Engine jest serwisem hostingowym dla aplikacji sieciowych. Project Hosting umożliwia użytkownikom dostępu do kodu open source. Google Web Toolkit pozwala programistom tworzyć aplikacje AJAX w języku Java.

Witryna zawiera również informacje o projektach oprogramowania, w których Google bierze udział, takich jak system operacyjny Android dla telefonów komórkowych czy OpenSocial.

**2. Funkcjonalność code.google.com:**

**-** najpopularniejsze usługi i platformy- Android, Cloud Platform, Firebase, Maps, Web, Analitics

**6.TFS.APP.VISUALSTUDIO.COM**

**1. Program Team Fundation Sever-** udostępnianie kodu. Śledzenie pracy. Publikowanie oprogramowania. Zintegrowany zestaw serwerów zawierający narzędzia programistyczne przeznaczone dla profesjonalistów.

**2. Funkcjonalność tfs.app.visualstudio.com:**

### **- system git-** wszystkie potrzebne funkcje systemu Git oraz bezpłatne, prywatne repozytoria, społecznościowe przeglądy kodu- Społecznościowy wymiar przeglądu kodu dzięki zaawansowanym funkcjom. Stany komentarzy i dyskusje pogrupowane w wątki ułatwiają śledzenie opinii. Ciągła integracja obejmująca wszystkie zmiany do żądania ściągnięcia Możliwość konfiguracji scalania obejmującej m.in. niestandardowe komunikaty czy eliminowanie błędów. Dzięki @wzmiankom dotyczącym osób i elementów roboczych wszyscy mają dostęp do aktualnych informacji. Śledzenie kodu od etapu projektowania do wydania- tworzenie gałęzi bezpośrednio z poziomu listy prac lub tablicy Kanban. Łatwe śledzenie stanu kompilacji i wersji kodu. Zatwierdzenia i żądania ściągnięcia są automatycznie łączone i wyświetlane w elementach roboczych.Jedno rozwiązanie dla zespołu korzystającego z wielu platform. Wbudowana ciągła integracja i pomoc techniczna dotycząca środowiska Jenkins i innych narzędzi. Zaawansowana integracja z programami Visual Studio i Visual Studio Code. Rozszerzenia dla platform Android Studio, Eclipse, IntelliJ i pozostałych środowisk IDE firmy JetBrains. Utrzymywanie jakości za pomocą zasad gałęzi- gwarancja kompilacji zmian przed wprowadzeniem ich do kopii głównej. Ograniczenie kręgu osób z uprawnieniami do współtworzenia określonych gałęzi. Automatyczne dołączanie właściwego recenzenta dla każdej zmiany kodu. Wymagania dotyczące recenzentów kodu ułatwiają egzekwowanie stosowania najlepszych rozwiązań.

### - **kontrola wersji w programie Team Fundation**- gotowa do użycia w przedsiębiorstwie kontrola wersji dla zespołów różnych typów i rozmiarów- Narzędzia Visual Studio Tools zwiększające wydajność pracy programistów. Najwyższej jakości integracja z programem Visual Studio. Wyróżniaj zmiany w kodzie za pomocą zaawansowanych narzędzi do porównywania folderów i plików. Śledź zmiany w kodzie w rozgałęzieniach dzięki potężnym wizualizacjom. Zintegrowane narzędzia przeglądu kodu umożliwiają uzyskanie informacji zwrotnej przed zaewidencjonowaniem kodu. Jedno rozwiązanie dla zespołu pracującego na wielu platformach. Szczegółowa integracja z popularnymi środowiskami IDE, w tym z programami IntelliJ oraz Eclipse. Wbudowana ciągła integracja i pomoc techniczna dotycząca środowiska Jenkins i innych narzędzi. Narzędzia wiersza polecenia dla deweloperów w systemie Linux i na innych platformach.

**- narzędzia Agile-** Tablice Kanban. Elastyczne tablice Kanban zmieniają Twoje dane w wizualne szyldy, które dostosowują się do Twoich procesów i stylu.  Aktualizacje na żywo zapewniają, że każdy członek zespołu może monitorować przepływ i być na bieżąco.Pulpity nawigacyjne. Dostosowywane pulpity nawigacyjne pomogą Ci w utworzeniu dopasowanych do Twoich potrzeb widoków danych. Konfiguruj układ, który będzie odpowiadał Twojemu zespołowi, i z łatwością monitoruj postęp cyklu powstawania projektu. Połączenie z kodem. Wszystkie zmiany kodu są bezpośrednio połączone z historią, usterką lub zadaniem powiązanym z daną pracą.  Niespotykana możliwość śledzenia i widoczność w rozwijanej bazie kodu. Tablice Scrum- używaj wbudowanych tablic Scrum i narzędzi planistycznych, aby ułatwić swojemu zespołowi przeprowadzanie efektywnych spotkań aktualizacyjnych, planistycznych i retrospekcyjnych.

**3. Ceny:**

**-small teams-** free- Dostęp do [funkcji podstawowych](https://www.visualstudio.com/?page_id=23900), takich jak nieograniczona liczba repozytoriów Git, narzędzia Agile, testowanie eksploracyjne, zarządzanie wydaniami i wiele innych, Nieograniczona liczba użytkowników z dostępem do elementów roboczych, 1 potok prywatny do uruchamiania kompilacji i wdrażania wydań z własnego serwera, 1 potok hostowany (4 godziny miesięcznie) do uruchamiania kompilacji i wdrażania wydań w chmurze

-**growing teams:**

**-** 30$ dla 10 użytkowników

- 110$ dla 20 użytkowników

- 350$ dla 50 użytkowników

- 750$ dla 100 użytkowników

- 1150$ dla 200 użytkowników

- 4350$ dla 1000 użytkowników

**7.** **RIOUXSVN.COM**

**1. The RiouxSVN** project is based on a team of talented volunteers whose backgrounds range from web developers to sysadmins. Riox jest darmowe- istnieje jednak możliwość upgrade’ów ( patrz niżej).

**2. Funkcjonalność riouxsvn.com:**

**-** prywatne repozytorium (50MB, 4 sloty, nielimitowana liczba użytkowników)

- codzienne kopie bezpieczeństwa danych użytkownika

- brak reklam

- możliwość działania w zespole

- możliwość upgrade’ów (donate’y za dodatkową pamięć w repozytorium i dodatkowe sluty)

**Oprogramowanie klienckie**

Analiza oprogramowani klienckich dla repozytoriów kodu

**1.** **TORTOISE SVN**

**1. TortoiseSVN** – (opensource) interfejsy oraz zestaw narzędzi umożliwiających dostęp do systemu kontroli wersji SVN z poziomu menedżera plików.

TortoiseSVN działa jako rozszerzenie powłoki systemu Windows udostępniając dodatkowe ikonki i menu kontekstowe dla dowolnego menedżera plików. Dostarczane są także dodatkowe narzędzia m.in. do porównywania plików i przeglądania repozytoriów.

TortoiseSVN is a really easy to use Revision control / version control / source control software for Windows. It is based on Apache™ Subversion (SVN); TortoiseSVN provides a nice and easy user interface for Subversion.

It is developed under the GPL. Which means it is completely free for anyone to use, including in a commercial environment, without any restriction. The source code is also freely available, so you can even develop your own version if you wish to.

Since it's not an integration for a specific IDE like Visual Studio, Eclipse or others, you can use it with whatever development tools you like, and with any type of file.

**Features of TortoiseSVN**

* Easy to use
  + all commands are available directly from the Windows Explorer.
  + only commands that make sense for the selected file/folder are shown. You won't see any commands that you can't use in your situation.
  + See the status of your files directly in the Windows explorer
  + descriptive dialogs, constantly improved due to user feedback
  + allows moving files by right-dragging them in the Windows explorer
* All Subversion protocols are supported
  + http://
  + https://
  + svn://
  + svn+ssh://
  + file:///
  + svn+XXX://
* Powerful commit dialog
  + integrated spell checker for log messages
  + auto completion of paths and keywords of the modified files
  + text formatting with special chars
* The big picture
  + Can create a graph of all revisions/commits. You can then easily see where you created a tag/branch or modified a file/folder
  + Graphs of commit statistics of the project
  + Easy comparing of two branches or tags
* Per project settings
  + minimum log message length to avoid accidentally committing with an empty log message
  + language to use for the spell checker
* Integration with issue tracking systems

TortoiseSVN provides a flexible mechanism to integrate any web based bug tracking system.

* + A separate input box to enter the issue number assigned to the commit, or coloring of the issue number directly in the log message itself
  + When showing all log messages, an extra column is added with the issue number. You can immediately see to which issue the commit belongs to.
  + Issue numbers are converted into links which open the webbrowser directly on the corresponding issue
  + Optional warning if a commit isn't assigned to an issue number
* Helpful Tools
  + TortoiseMerge
    - Shows changes you made to your files
    - Helps resolving conflicts
    - Can apply patchfiles you got from users without commit access to your repository
  + TortoiseBlame: to show blames of files. Shows also log messages for each line in a file.
  + TortoiseIDiff: to see the changes you made to your image files
  + SubWCRev: to include the revision numbers/dates/... into your source files
* Available in many languages
* TortoiseSVN is stable
  + Before every release, we create one or more "release candidates" for adventurous people to test first.
  + During development cycles, many people test intermediate builds. These are built every night automatically and made available to all our users. This helps finding bugs very early so they won't even get into an official release.
  + A big user community helps out with testing each build before we release it.
  + A custom crash report tool is included in every TortoiseSVN release which helps us fix the bugs much faster, even if you can't remember exactly what you did to trigger it.

**2.** **TORTOISE GIT**

**Git** – rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2.

Pierwsza wersja narzędzia Git została wydana 7 kwietnia 2005 roku, by zastąpić poprzednio używany w rozwoju Linuksa, niebędący wolnym oprogramowaniem, system kontroli wersji BitKeeper.

**Dobre wsparcie dla rozgałęzionego procesu tworzenia oprogramowania**: jest dostępnych kilka algorytmów łączenia zmian z dwóch gałęzi, a także możliwość dodawania własnych algorytmów[[1]](https://pl.wikipedia.org/wiki/Git_(oprogramowanie)#cite_note-1).

**Praca off-line**: każdy programista posiada własną kopię repozytorium, do której może zapisywać zmiany bez połączenia z siecią; następnie zmiany mogą być wymieniane między lokalnymi repozytoriami.

**Wsparcie dla istniejących protokołów sieciowych**: dane można wymieniać przez HTTP(S), FTP, rsync, SSH.

**Efektywna praca z dużymi projektami**: system Git według zapewnień Torvaldsa, a także według testów fundacji Mozilla, jest o rzędy wielkości szybszy niż niektóre konkurencyjne rozwiązania.

**Każda rewizja to obraz całego projektu**: w przeciwieństwie do innych systemów kontroli wersji, Git nie zapamiętuje zmian między kolejnymi rewizjami, lecz kompletne obrazy. Z jednej strony wymaga to nieco więcej pracy aby porównać dwie rewizje, z drugiej jednak pozwala np. na automatyczną obsługę zmian nazw plików.

**3.** **TORTOISE CVS**

**CVS** (ang. *Concurrent Versions System*) – system kontroli wersji udostępniany na licencji GPL. Przeznaczony do pracy grupowej nad [kodem](https://pl.wikipedia.org/wiki/Kod_%C5%BAr%C3%B3d%C5%82owy) programów lub innych projektów realizowanych w zapisie elektronicznym. CVS zbudowany jest w architekturze klient-serwer. Stworzony na bazie [RCS](https://pl.wikipedia.org/wiki/Revision_Control_System) przez Dicka Grune w 1986 roku.

Od początku lat 90. XX wieku CVS jest wykorzystywany jako narzędzie pracy grupowej w wielu projektach programistycznych, których współpraca opiera się na wykorzystaniu Internetu – są to m.in. całe systemy operacyjne takie jak OpenBSD i NetBSD.

W stosunku do nowszych systemów kontroli wersji, CVS jest często krytykowany przez użytkowników za poniższe cechy:

* operacje na repozytorium nie są atomowe,
* stworzenie nowej gałęzi pochłania dużo miejsca na dysku,
* brak wersjonowania dla zmian nazw i usuwania plików,
* brak wsparcia dla rozproszonego systemu kontroli wersji.

W celu wyeliminowania powyższych braków, powstały bardziej niezawodne systemy mające docelowo zastąpić CVS jak np. Subversion

TortoiseCVS lets you work with files under CVS version control directly from Windows Explorer. It is freely available under the GPL.

With TortoiseCVS you can directly check out modules, update, commit and see differences by right clicking on files and folders within Explorer. You can see the state of a file with overlays on top of the normal icons within Explorer. It even works from within the file open dialog.

You can perform tagging, branching, merging and importing, and you can go directly to a browser web log (using [ViewCVS](http://www.lyra.org/viewcvs/) or [CVSWeb](http://stud.fh-heilbronn.de/~zeller/cgi/cvsweb.cgi/)) on a particular file.

* If you want to work with an existing repository, [download](http://www.tortoisecvs.org/download.shtml) TortoiseCVS and read the User's Guide.
* To create your own local repository to work alone on, [read this FAQ entry](http://www.tortoisecvs.org/faq.shtml#cvsinit).
* If you want to set up a new shared repository, you will need a CVS server. We recommend [CVSNT](http://www.cvsnt.org/) for both Unix and Windows machines. TortoiseCVS can also be used with original [CVS](http://ximbiot.com/cvs/), but some features depend on a [CVSNT](http://cvsnt.org/wiki) server.

There are some CVS things that you can't do with TortoiseCVS - have a look at the [feature requests database](http://sourceforge.net/tracker/?atid=451975&group_id=48103&func=browse). The GPL license lets you improve TortoiseCVS, as long as you make any changes to the source code available. If you would like to help implement some features, please see the [development](http://www.tortoisecvs.org/development.shtml) section.

**3. APACHE SUBVERSION**

Apache Subversion is a full-featured version control system originally designed to be a better CVS. Subversion has since expanded beyond its original goal of replacing CVS, but its basic model, design, and interface remain heavily influenced by that goal. Even today, Subversion should still feel very familiar to CVS users.

The following list of features is presented with the assumption that you, the reader, have a basic understanding of what version control is and how version control systems work in general. If there's a feature that you're looking for that is not represented in this list, feel free to ask about it on our project mailing lists — perhaps we just didn't think to list it here. If Subversion truly lacks a feature you need, your feedback will help us to improve Subversion, and in the meantime, perhaps we can help you meet your need with the features that Subversion does have.

* **Most CVS features.**

CVS is a relatively basic version control system. For the most part, Subversion has matched or exceeded CVS's feature set where those features continue to apply in Subversion's particular design.

* **Directories are versioned.**

Subversion versions directories as first-class objects, just like files.

* **Copying, deleting, and renaming are versioned.**

Copying and deleting are versioned operations. Renaming is also a versioned operation, albeit with some quirks.

* **Free-form versioned metadata ("properties").**

Subversion allows arbitrary metadata ("properties") to be attached to any file or directory. These properties are key/value pairs, and are versioned just like the objects they are attached to. Subversion also provides a way to attach arbitrary key/value properties to a revision (that is, to a committed changeset). These properties are not versioned, since they attach metadata to the version-space itself, but they can be changed at any time.

* **Atomic commits.**

No part of a commit takes effect until the entire commit has succeeded. Revision numbers are per-commit, not per-file, and commit's log message is attached to its revision, not stored redundantly in all the files affected by that commit.

* **Branching and tagging are cheap (constant time) operations.**

There is no reason for these operations to be expensive, so they aren't.

Branches and tags are both implemented in terms of an underlying "copy" operation. A copy takes up a small, constant amount of space. Any copy is a tag; and if you start committing on a copy, then it's a branch as well. (This does away with CVS's "branch-point tagging", by removing the distinction that made branch-point tags necessary in the first place.)

* **Merge tracking.**

Subversion 1.5 introduces merge tracking: automated assistance with managing the flow of changes between lines of development, and with the merging of branches back into their sources. The 1.5 release of merge tracking has basic support for common scenarios; we will be extending the feature in upcoming releases.

* **File locking.**

Subversion supports (but does not require) locking files so that users can be warned when multiple people try to edit the same file. A file can be marked as requiring a lock before being edited, in which case Subversion will present the file in read-only mode until a lock is acquired.

* **Symbolic links can be versioned.**

Unix users can place symbolic links under version control. The links are recreated in Unix working copies, but not in win32 working copies.

* **Executable flag is preserved.**

Subversion notices when a file is executable, and if that file is placed into version control, its executability will be preserved when it it checked out to other locations. (The mechanism Subversion uses to remember this is simply versioned properties, so executability can be manually edited when necessary, even from a client that does not acknowledge the file's executability, e.g., when having the wrong extension under Microsoft Windows).

* **Apache network server option, with WebDAV/DeltaV protocol.**

Subversion can use the HTTP-based WebDAV/DeltaV protocol for network communications, and the Apache web server to provide repository-side network service. This gives Subversion an advantage over CVS in interoperability, and allows certain features (such as authentication, wire compression) to be provided in a way that is already familiar to administrators

* **Standalone server option (svnserve).**

Subversion offers a standalone server option using a custom protocol, since not everyone wants to run an Apache HTTPD server. The standalone server can run as an inetd service or in daemon mode, and offers the same level of authentication and authorization functionality as the HTTPD-based server. The standalone server can also be tunnelled over ssh.

* **Parseable output.**

All output of the Subversion command-line client is carefully designed to be both human readable and automatically parseable; scriptability is a high priority.

* **Localized messages.**

Subversion uses gettext() to display translated error, informational, and help messages, based on current locale settings.

* **Interactive conflict resolution.**

The Subversion command-line client (svn) offers various ways to resolve conflicting changes, include interactive resolution prompting. This mechanism is also made available via APIs, so that other clients (such as graphical clients) can offer interactive conflict resolution appropriate to their interfaces.

* **Repository read-only mirroring.**

Subversion supplies a utility, svnsync for synchronizing (via either push or pull) a read-only slave repository with a master repository.

* **Write-through proxy over WebDAV.**

Subversion 1.5 introduces a write-through proxy feature that allows slave repositories (see read-only mirroring) to handle all read operations themselves while passing write operations through to the master. This feature is only available with the Apache HTTPD (WebDAV) server option.

* **Natively client/server, layered library design with clean APIs.**

Subversion is designed to be client/server from the beginning; thus avoiding some of the maintenance problems which have plagued CVS. The code is structured as a set of modules with well-defined interfaces, designed to be called by other applications.

* **Binary files handled efficiently.**

Subversion is equally efficient on binary as on text files, because it uses a binary diffing algorithm to transmit and store successive revisions.

* **Costs are proportional to change size, not data size.**

In general, the time required for a Subversion operation is proportional to the size of the changes resulting from that operation, not to the absolute size of the project in which the changes are taking place.

* **Bindings to programming languages.**

The Subversion APIs come with bindings for many programming languages, such as Python, Perl, Java, and Ruby. (Subversion itself is written in C.)

* **Changelists.**

Subversion 1.5 introduces changelists, which allows a user to put modified files into named groups on the client side, and then commit by specifying a particular group. For those who work on logically separate changesets simultaneously in the same directory tree, changelists can help keep things organized.

* **And more...**

...even when we manage to keep this list up-to-date, it isn't possible to list every little feature. See the documentation for more information.

**WYBÓR I UZASADNIENIE**

**4.** Moim zdaniem najlepszym systemem hostingu projektów informatycznych zarówno dla projektów komercyjnych jak i open-source jest Github, z następujących względów: Github jest stosunkowo tani ( w porównaniu do innych systemów, np. Assemblii), jest bardziej niezawodny (częste problemy z commit’owaniem w Assembli), a także posiada wiele funkcji, niektóre z nich nie są dostępne w innych systemach, np. Ignore list). Liczba osób korzystających z Github’a potwierdza tylko jego popularność, co również jest plusem ze względu na możliwość współpracy z innymi użytkownikami.

**5.** Myśle, że najlepszym oprogramowaniem klienckim dla repozytoriów kodu jest Tortoise SVN. Tortoise SVN jest stosunkowo prosty w obsłudze i daje wiele możliwości, m.in. narzędzia do  porównywania plików i przeglądania repozytoriów.