Plastic SCM and Subversion

www.codicesoftware.com

códicesoftware

Summary

Plastic SCM both supports and integrates any development pattern (agile, RUP, cascase) with a company's development infrastructure. Subversion is an open-source tool that provides basic version control features for small teams with limited process need, but it does not provide with the required functionalities to manage Software Configuration Mangement requirements for medium and big teams.

Feature	Subversion	Plastic SCM
Visualization tools	Subversion provides with a limited number of visualization tools such as a graphic version tree on which display the relations between different versions of the history of item.	Plastic SCM provides with new-generation interactive visualization tools to improve both developers and managers productivity and performance, as well as increasing visibility of the development's evolution. Plastic Branch Explorer provides with an updated view of the state of the development for managers to be aware of the situation of the project at any time, and it also eases the work of integrators and developers on managing development tasks. The Statistics tool offers a transparent and effective view of the development areas which are being more often modified.
Distributed Support	Subversion does not support distributed mode.	Plastic SCM supports distributing "parts" of its repositories to other servers, so they can separately evolve and merged together afterwards. Every operation is available on the distributed repositories, providing with an ideal scenario to maintain teams working on parallel from different locations or for developers working disconnected from their laptops.
Support for real parallel development	Subversion does not support complex parallel development, which is nowadays, a must for any size company. It does not even provide with merge traceability, resulting on developers having to deal with very complex integrations and wasting their time trying to solve them.	Plastic SCM implements Branch Inheritance technology, which includes complete traceability of merges, and it provides with powerful visualization tools which ease the merge process to a great extent, enabling real parallel development through specific branches for every task, providing the best scenario for task per branch development.
Refactor support	Inaccurate management of moved, renamed and added files when integrating between branches turns very often into a release which cannot be compiled.	Plastic SCM full project's structure versioning ensures that refactor operations (move, rename, add...) are versioned and integrated on branches at the same level as they are on files. The integrations with IDEs (Visual Studio, Eclipse and

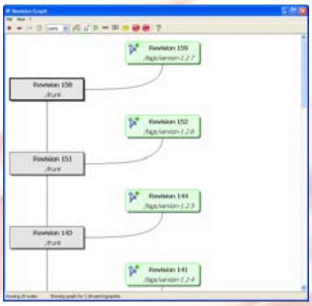
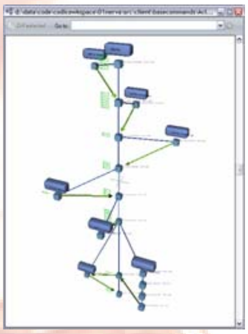
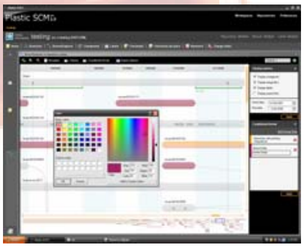
		JDeveloper) are in charge of properly propagating refactors into the version control.
Usability	Subversion is quite easy to use but it does not provide with the required features for assisting teams on a parallel development environment. There are third-party tools which provide with some of these features but each of them is a combination of a different set of characteristics.	Plastic SCM provides with complete graphic user and command line interfaces, offering its full set of functionalities on a consistent manner. Furthermore, Plastic´s Branch Inheritance technology, Branch Explorer and 3D version tree simplify parallel development to an even greater extent.
Analysis and reports	Subversion does not have a query interface, even though there are third party tools which provide with certain query ability.	Plastic SCM offers an extensive query tool which is an integrated part of the system, directly automated through the command line interface and supporting the most widely used standards such as XML. Additionally, Plastic SCM supports widely used database backends such as MySQL, SQL Server or Firebird, which provides serialized reporting and data-mining options.
Installation, maintenance and security	Subversion requires manual configuration in order to use authentication against Active Directory or LDAP. Permit support is limited to read/write.	Plastic SCM seamlessly integrates with Active Directory, LDAP or NIS and includes the required configuration by default to start working in 3 minutes. It also provides a security mechanism based on ACLs for every system object.

Support for parallel development processes

Visualization tools

Plastic SCM provides with new-generation interactive visualization tools to improve both developers and managers productivity and performance, as well as increasing visibility of the development's evolution.

- The Branch Explorer provides with an updated view of the state of the development for managers to be aware of the situation of the project at any time, and it also eases the work of integrators and developers on managing development tasks. Its conditional format allows easy visualization of critical points requiring attention such as tasks which have not been integrated or query results. Subversion does not provide with any feature like Plastic´s branch explorer.
- Plastic SCM Statistics graphic tool offers a transparent and effective view of the development areas which are being more often modified. Statistics in Subversion are only applied to the amount of *commits* done by Developer and a history of the amount of commits.
- Plastic SCM 3D version tree is able to show the complex relationships resulting of the usage of intensive *branching*. Using Subversion, the version tree only provides with a limited view of complex history, where a diagram like this would be adding more value to visualization.

Feature	Subversion	Plastic SCM
Version Tree		
Branch Explorer	Subversion does not provide such a feature	
Statistics		

Distributed Development

Plastic SCM supports the pattern known as distributed SCM, which allows replicating copies of certain parts of a repository (usually a group of branches) into another server. Once replicated, branches can separately evolve and they can be merged together anytime.

This mechanism allows keeping replicated servers on different locations, teams doing changes in parallel and executing every version control operation without a decrease on performance, and afterwards, merging the changes back on both sides. This method provides an increase on productivity, opposed to solutions based on *proxies*.

Subversion does not have distributed support and the support it provides for disconnected usage does not allow users to do any version control operation until they regain connection.

Refactor support

Subversion does not provide with full project structure versioning. It does have both *move* and *copy* commands but the *move* command is no more than a mixture of *svn copy* and *svn delete.* This has special impact whenever a **refactor** operation needs to be propagated on a merge from one branch into another, which is a very common operation, especially for Java projects. Moved files are considered are totally independent entities, and their content are not merged. Let's see an example: a file, *class.java* is renamed as *myclass.java* on a branch created to work on a certain functionality or bug. Later, *class.java* is modified on the main branch. When a merge is done from the main branch to the bugfix, a new file *class.java* appears in the workspace, and *myclass.java* does not contain the changes. Detecting this behaviour is not as easy as inaccurate management of renamed items is only detected whenever compilation fails. From Subversion user guide: "Until Subversion improves, be very careful when merging copied and renamed items from one branch into another".

Branch inheritance technology

Plastic SCM uses branch hierarchy to represent the development structure as well as a company's processing model. As each branch has a clear position on the hierarchy, it is possible to represent them visually. As branches are only metadata, developers can use as many as needed for implementing the development process, without affecting too much the size of the repository.

Whenever a change is integrated on a parent branch, it is automatically detected on its child branches, unless these branches contain changes on the same file. With Subversion, creating branches is simple as branches are actually copies and there is no internal concept of branch. Subversion creates light copies, which do not increase the repositories load too much as it does not copy the contents of the files. Branches are usually created on a subdirectory called Branches, inside the project, where the whole project or part of it, can be copied. Even though a little of the branch structure can be seen through the "repo browser", it only shows the existing branches but not their relations or their life cycle. The version graphics can only show the history of an item but it does not provide with a global view of the project as Plastic´s Branch Explorer does.

Private branches

By using the branch per task or branch per Developer patterns (Stephen P. Berczuk, 2002), developers can commit (check-in) their changes without interfering with other developers´. Furthermore, it is the best mechanism for reviewing the code before integrating changes into the main line, as once changes are done on a branch, another developer can check them without having to phisically access the other developer´s workspace.

Statistics and Reports

Plastic SCM includes an extensive query tool which is integrated into the system and automated through the console, providing support for the most widely used standards such as XML. It is very easy to obtain reports importing data to Microsoft Office with just a click, or programming a task to generate a web report at a certain time.

Additionally, Plastic SCM supports widely used database backends such as MySQL, SQL Server or Firebird, which provides serialized reporting and data-mining options.

Finally, Plastic SCM provides with the changes statistics module, which allows not only viewing information but also interact with the diverse objects to get a clear image of the development parts which are being modified on each period of time.

Subversion provides with statistics and query capabilities which are very constrained due to its *backend* and the specific tools available.

Bibliography

Stephen P. Berczuk, B. A. (2002). *Software Configuration Management Patterns: Effective Teamwork, Practical Integration .* Addison-Wesley Professional.

códicesoftware

www.codicesoftware.com email: pr@codicesoftware.com