SIEMENS
Ingenuity for life

Five things most Subversion users don't know

When the Subversion® (SVN) versioning and revision control software was introduced, it quickly became the market-leading version control system, relegating all others to virtual obsolescence – something unprecedented for an open-source product. But even as widespread as Subversion is today, most companies still have a limited understanding of its potential to manage far more than just source code. If you are using Subversion only for source code management, you're definitely not leveraging all the power and potential SVN can deliver.

A white paper issued by: Siemens PLM Software

Contents

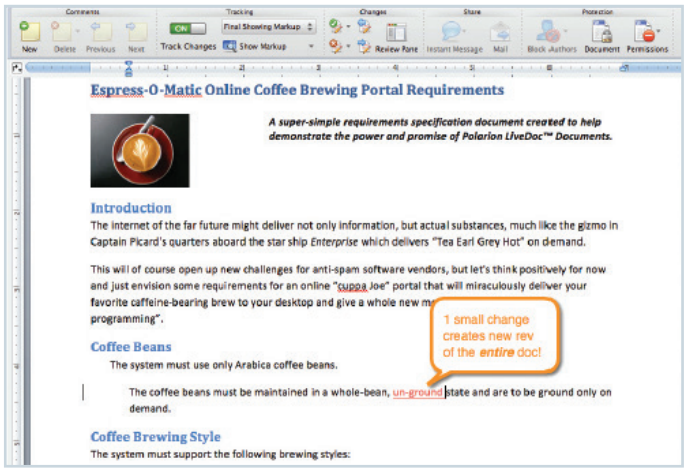
Five things most users never do with SVN

1. Use SVN to track everything
2. Use SVN to create and manage traceability
3. Use SVN to control your development process
4. Pull information from your SVN repository for reports
5. Use SVN to unify your entire development organization

Let's look closer at each of these missed opportunities.

1. Track everything
You know that SVN is great for versioning source code. So why stop there, as most people do? You can use Subversion to track every single artifact in your development process: requirements, defects/issues, change requests, test cases, test execution results. Whatever else you have, you can track it with Subversion.

Take requirements, for example (or "user stories," if you use Scrum or another Agile development methodology). The most common practice is to commit entire specification documents to an SVN repository, mainly because that's better than nothing. But people quickly discover that managing the inevitable change from inception through implementation is a pain.



Requirement changed in Microsoft Word

Change one requirement in a Microsoft® Word document, and you have a new revision of the whole document. It's a tedious chore to find out what content was changed, when, why, and by whom. Change seldom involves the entire specification. Usually it's just one requirement, or a few. So wouldn't it be much better if you could version each single requirement in the specification document? Subversion can in fact handle such changes, but most SVN users simply don't realize it.

There are further difficulties with the "version the whole document" approach. What if you have a formal review and approval process? Assuming you get the right revision of the spec in front of all reviewers at the same time, you need to track who has reviewed which individual requirements and/or the whole specification, and who has approved or rejected which specific requirements (or the whole document). It is also advantageous to capture comments when someone sends something back to the drawing board, especially when you must prove due diligence to a regulator to get your new product approved for sale. The fact is, it's possible to do all this with Subversion. We'll talk about how later on.

2. Create traceability

Suppose you have requirements for something you are developing, and that you also have test cases that verify those requirements. Perhaps you are so far advanced as to be versioning each requirement and each test case in your Subversion repository. Versioned test cases and requirements are important, but thus far there is no linking that enables you to demonstrate which test cases verify which requirements, or conversely, that a requirement is verified by a specific test case.

The same holds true for any other relationships you may need to track, including parent/child, dependency, etc. A database can be used to solve the problem, but would introduce yet another tool in the chain to be installed, administered, configured and maintained. Wouldn't it be preferable to track relationships with Subversion? The good news is that you can. Traceability links, and the relationships they define, can be properties of any objects you manage in a Subversion repository. We'll talk more about specifics later.

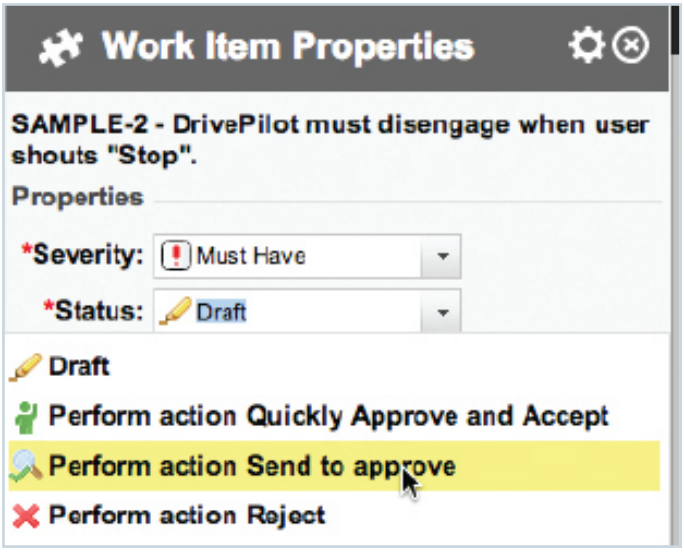
Artifacts in SVN normally have no traceability. This can be remedied.

3. Control process

Workflow is a hugely important issue, with impact ranging from team productivity to the company's ability to compete. Most companies invest heavily in creating the most efficient process they can. They must not only define it, but also ensure that people know what it is and how to follow it. Process automation, where possible, delivers additional benefits. In regulated industries, it is vital to have the process well documented and controlled to ensure that nothing is ever missed or omitted. Lots of information is involved.

Imagine the benefits if that information could be leveraged in tools that would communicate about and guide people's work so they don't need to know the entire complex process, but just the steps they have to do. What if the same tools could track objects through the process so that people always know the current status of both the objects (requirements, tests, defects, etc.), and the project as a whole?

And what if all that information could be handled by Subversion, which you already have and are comfortable hosting, administering and using day to day? You may never have stopped to think that Subversion can be used as the central repository for this kind of information. But it can be. Again, we'll see some solution specifics later.



Use SVN for workflow control. No need to know the entire process, only possible next step(s).

4. Report from your SVN repo

Once you have information tracked in a Subversion repository, it is possible to extract different pieces, combine them in different ways, and output highly useful reports for teams, managers, customers and other stakeholders. The possibilities are limited only by your imagination and programming skills. The bottom line is information. Once you have it, you can buy or build tools to manipulate it in whatever ways you need.

Traceability Report

Expand Tools

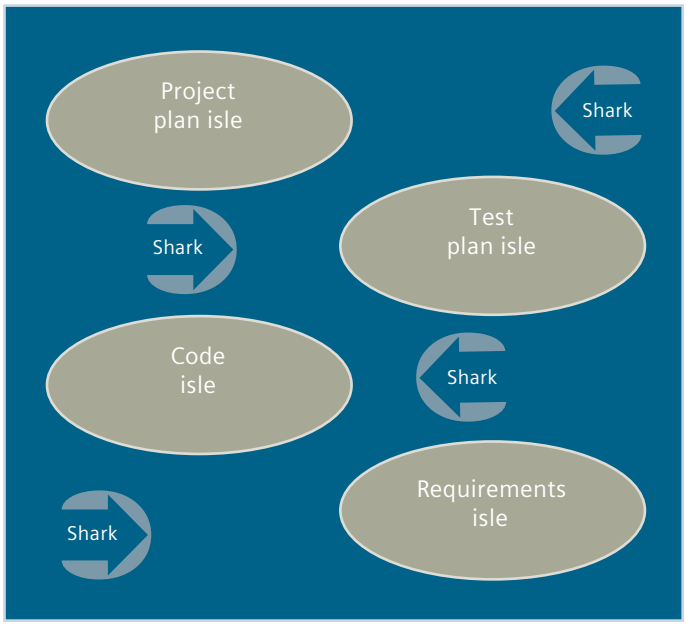
Thu Dec 19 14:03:37 UTC 20

ID-Title	Link Role	Suspect	Status
* DP-313 - DrivePilot shall easily engage operations while the vehicle is at rest...			
DP-314 - DrivePilot may not be engaged while the vehicle is under manual control.* provid...			
DP-315 - DrivePilot shall be easy to operate without extensive training.			
DP-316 - Before any user may engage DrivePilot on public roads, that user must successful...			
~~DP-463~~ - DrivePilot requires confirmation of a test drive	implements		Verified as Done
~~DP-460~~ - Require Tutorial before first run	implements		Verified as Done
DP-224 - A tutorial is required before the first run.	verifies		Active
DP-317 - Drive Pilot warnings on special conditions			
DP-454 - Drive Pilot warnings on special conditions	implements		Reopened
DP-473 - DrivePilot will disengage when User shouts "Stop"	verifies		Active
DP-474 - DrivePilot will disengage when Brake is manually engaged	verifies		Active

When data is in SVN, it can be reported in useful ways.
When developing software applications, Subversion can be the repository for every kind of information you need for your development process.

5. Unify your development organization

All too often, application and product development organizations end up with different functions in different, semi-isolated organizational islands. For example, one island conducts risk analysis and sends a document with their results to the island that develops requirements. On that island, there may be sub-islands for functional requirements and system requirements. The output of the requirements island (typically one or more specification documents, often versioned in SVN) goes to the island that develops the application. That island may have a sub-island that develops and conducts tests (or that may be yet another autonomous isle).



A typical "islands" approach to application development

From a corporate management viewpoint, it can make sense to manage these islands separately. But in practical terms of getting product out the door and revenues flowing in, this kind of separation usually makes development more difficult for all concerned.

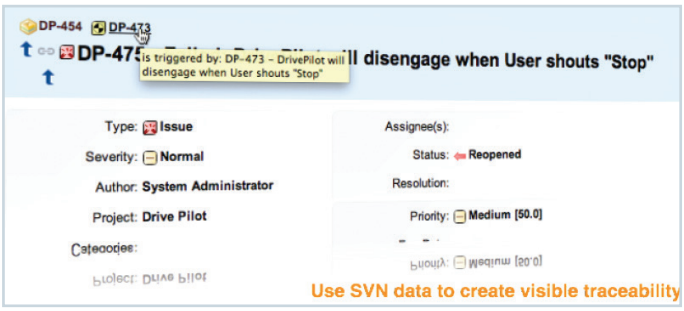
Companies can both store and track every kind of granular development artifact, along with its change history, its relationships to other artifacts, the process it is part of, and so on in Subversion.

Envision the possibilities of a common development platform and tool set that could be used to get all those various silos working together much more closely and efficiently. The same platform would enable the risk management unit to create risk analysis specifications containing granular risk artifacts that define a hazard, failure modes according to failure modes and effects analysis (FMEA), etc. – all stored in Subversion. These can be accessed via a web-based platform, making the risk analyses (and the current head revision thereof) accessible to the requirements management unit.

The requirements unit develops the various functional and system requirements specifications, containing granular requirement artifacts (even subatomic types like electrical requirements or hardware requirements) and stores them in SVN. Any of these can be linked via platform tools to any risk artifact and the links stored in Subversion to achieve traceability.

You can probably envision the extensions yourself: requirement artifacts linked to test case artifacts, test result artifacts that link to the test cases tested, and issue artifacts resulting from test failures, to cite a very few in the huge realm of possibilities.

Every unit creates its own brand of artifacts, and can link them to artifacts created and managed by other units. The platform could easily present visualizations and reports. Users could query for information in real time. Managers could always see the actual status and progress of projects, with data pulled from the head revisions in the underlying SVN repository. It should be clear by now that Subversion's potential goes far beyond simple source code version control.



Traceability created with data tracked in Subversion

Implementing advanced functions with Subversion

How can development teams actually implement advanced functions with Subversion? All Polarion solutions from Siemens PLM Software for requirements management, quality assurance and application lifecycle management, plus Polarion vertical solutions for automotive, medical devices, and Agile software development use Subversion in the ways discussed above, and in other ways you may never have imagined possible. Polarion can even do it in the cloud.

You can merge any existing SVN repo with the one integrated into Polarion, or you can continue to use one or more SVN repos to manage your code base, and let Polarion's integrated repo handle everything else, including establishing traceability to revisions in your external source code repo(s).

How to learn more
If you're intrigued by the potential for your development organization, then we highly recommend that you explore Polarion solutions on our web site at https://polarion.plm.automation.siemens.com. As you explore, remember one thing: it's all done with Subversion.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

www.siemens.com/plm

55672-A3 7/16 F