Subject: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Fri, 01 Nov 2019 13:17:30 GMT
View Forum Message <> Reply to Message

Hello,

As you may know U++ main version control system is SVN. We have only GitHub mirror available under https://github.com/ultimatepp/mirror/commits/master. It means that we can not use some powerful tools related to Git workflow. The main problem here is lack of Pull Request mechanism. Instead of using well defined process, we apply improvements directly mentioned on the forum or delivered within .diff files. The problem with that approach is that code review are extremely hard. Effectively, it scares away developers from making biggest contribution than several lines. Moreover, even if somebody made big change, it discourages Upp maintainers from applying this change especially when there are some comments related to the code. Personally, I remembered when I worked closely with omari on Macro Manager for TheIDE. It took some time to polish the code via discussing it on redmine... Moreover, I am pretty sure that we will be able to attract more developers to U++ itself, when we will allow for less painful and transparent contribution. At the end, I would like to add that with PR mechanism we are fully protected by unintentional changes like we have with current SVN privileges base approach.

Sometime ago Mirek mentioned that one of the problem with Git transition is the size of the Repository. In my opinion and with my current knowledge about the Git, it can be easily avoided. The main idea for that is to split main U++ repository into sub-modules like uppsr, examples, references etc. Each sub module is a small piece that doesn't weigh too much in compared with whole repository and it can be used separately. For example if you dwonload U++ you can replace your local uppsrc within the need of downloading full repository.

There is a one problem with submodules. It must be updated manually on the main repository level. However, this task can be easily automated by the script. For example we can launch script once per day that will update submodules and made the commit to the main repository. Then we will be one hundred percent sure that submodules doesn't live own life.

When we will adopt git then we can made some other improvements to the whole U++ world. The main idea I have in this area is remote package concept executed by the link to the github repository. The similar approched is used in GoLang. So, instead of copying packages from web you simply create package that points to some remote localization. Let's take into consideration upp-ccomponents developed by Oblivion. For example I would like to use MessageCtrl, so all I need to do is to add remote package that contains only link to the repository. The latest sources are downloaded automatically and you can use it directly in your project. Here is the example of .upp file modification:

```
description "A passive message ctrl and manager for U++\377";

uses
    Core,
    CtrlLib,
    https://github.com/ismail-yilmaz/terminal-ctrl.git; // <- Some modification can be saved here like branch used (master by default)
                                                        //    for example stable-11-19 etc. or include prefix to avoid
```

colissions.

file
MessageCtrl.h,
MessageCtrl.cpp,
MessageCtrl.iml,
Docs readonly separator,
src.tpp,
Info readonly separator,
ReadMe,
Copying;

The examplary usage of remote package would be:

#include <Remote/TerminalCtrl/TerminalCtrl.h> // <- All TerminalCtrl headers all avaliable from this place.

The git will be required for projects that will use above mechanism. So, the bare Upp shouldn't be affected by this dependency. However, all https requests will require additional OpenSSL dependency for TheIDE and UMK.

The above proposition aims to strengthen the whole U++ framework and increase it's popularity within developers. I think this changes should be incorporated as soon as possible and I think 2020.1 release cycle is the good moment to do them. Thanks!

Additional resources:
- https://github.com/golang/example/blob/master/hello/hello.go - usage of GitHub repo import in GoLang

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Fri, 01 Nov 2019 15:39:23 GMT
View Forum Message <> Reply to Message

Klugier wrote on Fri, 01 November 2019 14:17The problem with that approach is that code review are extremely hard.

Code reviews are always hard, but I am not sure that going GIT would make tham sigificantly easier. Besides, I believe you can do pull requests against mirror.

Quote:
[code]description "A passive message ctrl and manager for U++\377";

uses
Core,
CtrlLib,
https://github.com/ismail-yilmaz/terminal-ctrl.git; // <- Some modification can be saved here like branch used (master by default)

OK, having above capability would not hurt anything. Come back when you have that implemented

As for going GIT for core sources, I am very cautios. But you can perhaps try with bazaar first.

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Fri, 01 Nov 2019 16:07:00 GMT
View Forum Message <> Reply to Message

Hello,

Quote:
Code reviews are always hard, but I am not sure that going GIT would make tham sigificantly easier. Besides, I believe you can do pull requests against mirror.

Would the pull request made into mirror will be back-propagated to SVN?

Quote:
As for going GIT for core sources, I am very cautios. But you can perhaps try with bazaar first.

I am not sure how we can do it partially. Do you have some ideas to share?

svn://www.ultimatepp.org/upp/branches/gtk3 - is the good example why branching with SVN is expensive

Sicnerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Fri, 01 Nov 2019 16:56:43 GMT
View Forum Message <> Reply to Message

Klugier wrote on Fri, 01 November 2019 17:07Hello,

Quote:
Code reviews are always hard, but I am not sure that going GIT would make tham sigificantly easier. Besides, I believe you can do pull requests against mirror.

Would the pull request made into mirror will be back-propagated to SVN?

Automagically, no. But I am rather addressing the fact that this is already an option and it does not seem like there are many pull requests so far....

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Fri, 01 Nov 2019 20:57:37 GMT
View Forum Message <> Reply to Message

Hello,

FYI, We decided that we will make some initial git repository. The first repository I created is uppsrc submodule https://github.com/ultimatepp/uppsrc. You can simply replace your current uppsrc with this repository:

cd uppdir
rm -r -f uppsrc
git clone https://github.com/ultimatepp/uppsrc.git

and update it regularly with 'git pull origin master' command. I will try to sync it once in a week or two. In the future I plan to run this process on server...

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mr_ped on Sat, 02 Nov 2019 08:03:29 GMT
View Forum Message <> Reply to Message

I'm sort of confused, what's the issue with full "mirror"? It's currently ~360MB, that's nowhere close to the largest git repositories I'm using. I tried to clone it just now, and it took about 3-4min.

Splitting it to smaller ones would be probably of some benefit already, but it didn't feel like having the big repo is already prohibitive.

And the amount of files/etc... is still well within what git can handle easily.
(thinking about it, doesn't the windows git has some issues sometimes with repositories over

certain count of files, because the OS is failing to report the modified files correctly? Then again, does splitting the repos into parts help with that? And is Mirek still on windows machine as daily driver??? ... anyway, in linux repository of the "mirror" size doesn't have any issues AFAIK, it can manage 100x larger ones)

Klugier: about splitting the repo ... you are losing the the commit history, which is sometimes valuable info (you can annotate each line in source and see when some bug/fix was introduced and what other parts of source were modified). That's why I am asking what's the issue with one big git like "mirror" is, as I don't "see it".

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Sat, 02 Nov 2019 17:56:06 GMT
View Forum Message <> Reply to Message

Hello mr_pred,

Thank you for your feedback! I decided to delete these two new repositories uppsrc and example. The reason for that is that we should think what is the best approach of handling the git move. As I told in the previous post submodules gave us some overhead and it doesn't allow to modify submodule A and B in the same commit. For example if you want to modify uppsrc and fix example in the same time. It is not possible with this approach when we assume that uppsrc and examples are different sub-modules.

I list the size of the mirror repository (It weights around 710 Mb which is quite huge number):

```
8,5M    ./upptst
3,5M    ./examples
71M     ./uppbox
5,5M    ./autotest
91M     ./uppsrc
5,0M    ./rainbow
362M     ./.git
1,2M    ./tutorial
153M     ./bazaar
4,3M    ./reference
88K     ./autotst
9,9M    ./benchmarks
1,1M    ./archive
714M     .
```

For me the reasonable move is to create new repository without the history of the mirror (.git directory weights around 300 MB) and make decision of removing some code from there. I think bazzar is the good candidate of moving to separate sub-module especially that there is no dependency to the bazaar from other modules. All other directories will be part of the repository without sub-modules. I am aware that we will lose the history, but we will reduce the size of main repo by 300Mb+, which will give us quite reasonable repo size. So, here is the strategy:

8640 ./upptst
3516 ./examples
72592 ./uppbox // <- UppWeb weights around 62 Mb, so I think it is good candidate for submodule with the link
5544 ./autotest
92692 ./uppsrc
5068 ./rainbow
369812 ./.git
1148 ./tutorial
155900 ./bazaar // <- Will be moved to separate repository without the link...
4388 ./reference
88 ./autotst // <- Do we need it anyway, it is similar to autotest, should be unified in new repository
10060 ./benchmarks
1052 ./archive // <- I think it can be removed from new repo
730508 .

And then if you want to contribute, all you need to do is clone this repository update uppsrc directory and other assemblies as you want. After that you can simply create branch make changes and create pull request to the main repository.

Mirek, please let me know what do you think about this updated strategy. I can create the preview, but maintaining it for the long time doesn't have much sense.

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mr_ped on Sat, 02 Nov 2019 20:44:13 GMT
View Forum Message <> Reply to Message

"(It weights around 710 Mb which is quite huge number)"

Oh, you are not Android developer, are you?

Then it makes some sense, sometimes I forget how numb I've become to the file sizes since pretty much every work project on my disk is 200MB-1.5GB, and Android SDK is 50GB, so ~1GB for full framework seems to me very reasonable, and my computer can cope with it nicely, as it must cope with lot more bullshit daily.

But yeah, now I see it may be somewhat "big" for people not having to deal with the stuff like me... and not having android-dev grade HW, but just some regular powerful PC. Then it may be a bit of struggle I guess.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Sat, 02 Nov 2019 21:14:22 GMT
View Forum Message <> Reply to Message

Hello,

I am actually the Android developer For more details you can scan my Linkedin profile... Moreover, I developed Android builder for TheIDE in 2015 as my master thesis.

As I said before we should keep the current repository layout, but we can improved it future size by reducing some components that are not required in this particular repo. It is always better to have less than more. If we can do such optimization I do not see the reason to not do this. The only thing we will lose is the repository history, but I think we can live with it. There are some situation when you need, but in such case you can still use SVN...

Please notice that some folks still doesn't have huge hard drives. I personally have 500 Gb SDD, but for me every 1 - 0,5 GB counts and If I can save that space why do not do this? I know that some folks have 250 Gb or even 120 Gb, so for them this 710 Mb have bigger value. It all depends on the observer situation. To be clear reducing repository size is something that Mirek mentioned to me when I talked with him about the git transition, so I am really cautious here.

The overall goal is the git migration with all benefits it gave us. However, I am sure that we need the good plan for the migration and optimizing repository size is one of the thing that we should consider during transition.

I really appropriate your feedback, because it shows me that my initial plan is not good as I thought. Thanks!

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Sun, 03 Nov 2019 07:21:11 GMT
View Forum Message <> Reply to Message

In current model, theide is open to changes from many developers, documentation can be updated by anybody, Core/SSH is maintained by its author, bazaar is open, sandbox is open.

How would you like to handle this? Would all these become PR based for me to integrate?

What would happen to svn://www.ultimatepp.org/upp/bin/win32 ?

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Sun, 03 Nov 2019 07:22:48 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 02 November 2019 18:56
88 ./autotst // <- Do we need it anyway, it is similar to autotest, should be unified in new repository

It is package to test autotest.

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Fri, 15 Nov 2019 16:52:22 GMT
View Forum Message <> Reply to Message

Why not keep this whole thing simple, just remove the word "mirror" from the mirror?

I mean, GIT is decentralized. There is no problem with svn subtree being part of that, it basically acts as another remote. That way I can keep most of infrastructure and do not need to reinvent things.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Sat, 16 Nov 2019 09:01:51 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I think it is good point. We can migrate to git as a main development tool for some heavy changes that will require pull requests and we can still use SVN compatibility layer from GitHub https://help.github.com/en/github/importing-your-projects-to -github/support-for-subversion-clients. That's seems grate to me. Did you test this approach?

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Sat, 16 Nov 2019 09:14:07 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 16 November 2019 10:01Hello Mirek,

I think it is good point. We can migrate to git as a main development tool for some heavy changes that will require pull requests and we can still use SVN compatibility layer from GitHub https://help.github.com/en/github/importing-your-projects-to -github/support-for-subversion-clients. That's seems grate to me. Did you test this approach?

Sincerely,
Klugier

No need to use GitHub anything for this. I will keep mirroring cron running to push any changes from svn to git and use that repo (or other local, does not matter) to push any changes from git to svn.

That svn->git path works is obvious from the fact that mirror exists No reason why git->svn should not work, that is why git svn integration tools were build for... (basically to allow git developers use git in organizations with central svn repo, which is sort of exactly our case).

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Sat, 16 Nov 2019 09:17:14 GMT
View Forum Message <> Reply to Message

All that said, I think that perhaps forking 'theide' (or creating something based on it and/or compatible with .upp) with different project management and leader might be a good idea too. I can see that one could develop faster without my dictatorship

---

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Sat, 16 Nov 2019 15:03:22 GMT
View Forum Message <> Reply to Message

Hello,

Let's do this Git -> SVN synchronization and rename mirror to simply ultimatepp. I think all sides will be happy. The git users will have the opportunity to easily collaborate in the projects. The contributors will have the opportunity to provide more advances fixes that can be easily verify on the code review level by repo administrators. On the other hand, svn is still there, so you can easily modify the documentation or website. There are minimal chances that something will go wrong for example synchronization is too slow and there are conflicts. How often will the git repository be with sync with SVN?

I think, we can test the synchronization by simply modify README.md (I wanted to modify it 2 years ago with several attempts without success to this day). Please let me know when the synchronization will be enable, so I will create first exemplary pull request.

I am not sure about forking is the good idea for me - I do not have so much time to reasonably maintain fork. I can live with your dictatorship

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Sat, 16 Nov 2019 18:02:25 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 16 November 2019 16:03How often will the git repository be with sync with SVN?

Currently cronjob runs each 5 minutes (usually does nothing). I could probably make it run on svn commit, but IMO not worth the complication.

Quote:
I think, we can test the synchronization by simply modify README.md (I wanted to modify it 2 years ago with several attempts without success to this day). Please let me know when the synchronization will be enable, so I will create first exemplary pull request.

Sure, let us try.

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Sat, 16 Nov 2019 19:49:46 GMT
View Forum Message <> Reply to Message

Hello,

Here is the initial PR related to changes in README.md -
https://github.com/ultimatepp/ultimatepp/pull/7

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by koldo on Sun, 16 Feb 2020 10:00:29 GMT
View Forum Message <> Reply to Message

Is the Github U++ fork official? Should we follow from know on U++ Github issues and other GitHub stuff?
I have been just "invited" to do it, even though I am not in the U++ GitHub authors list.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by Klugier on Sun, 16 Feb 2020 17:25:57 GMT
View Forum Message <> Reply to Message

Hello koldo,

Right know GitHub acts as a official repository, however it is still the mirror. The point we did is that we want to encourage people to create their pull request from their and support the U++ project. I think Mirek right know is the only person who can reapply PR directly to U++ mirror. You can not click "Merge" button as you can on regular git repository.

In your case https://github.com/ultimatepp/ultimatepp/pull/11 you can just copied code from PR and commit is as regular commit to upp SVN repository and write that the PR can be closed. Personally, I would like to see better workflow here, but this is what we have now.

Sincerely,
Klugier

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by koldo on Sun, 16 Feb 2020 20:14:57 GMT
View Forum Message <> Reply to Message

Hello Klugier

I think what you're saying might cause some misunderstanding: AFAIK https://github.com/ultimatepp is not the official U++ repository, but the official U++ mirror in GitHub .

Otherwise a few more things would have to be modified.

I mean, it is great to be in GitHub since it's really trendy now, but in this case we should ask ourselves how to work from now on. The GitHub workflow offers several features, but many of them might not be very efficient in our project.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Sun, 16 Feb 2020 22:36:09 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 16 February 2020 21:14Hello Klugier

I think what you're saying might cause some misunderstanding: AFAIK https://github.com/ultimatepp is not the official U++ repository, but the official U++ mirror in GitHub .

How do you define mirror in distributed versioning system?

Quote:
Otherwise a few more things would have to be modified.

There would be no practical difference.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by koldo on Mon, 17 Feb 2020 08:11:46 GMT
View Forum Message <> Reply to Message

Hi Mirek

In plain text, the repository is the first place where changes are uploaded and a mirror is just a copy done regularly.
Is it U++ GitHub mirror anything different?

I think that standard GitHub Workflow (create branches, add commits, open pull requests, merge) is rather different than in U++.
Are we going to adopt this system, or are we going to continue with the current system?

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Mon, 17 Feb 2020 08:54:47 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 17 February 2020 09:11Hi Mirek

In plain text, the repository is the first place where changes are uploaded and a mirror is just a copy done regularly.
Is it U++ GitHub mirror anything different?

So all github repos are mirrors, as the first place where changes are uploaded is your local repository.

What I want to say here that 'mirror' does not have meaning with git. You have upstream and

downstream. U++ github us definitely upstream.

Quote:
I think that standard GitHub Workflow (create branches, add commits, open pull requests, merge) is rather different than in U++.
Are we going to adopt this system, or are we going to continue with the current system?

This system is already adopted. The only difference is that I am not using automatic merging of pull requests in github, but I would not do that anyway - I want to check it first in local repo. But that is behind the scenes.

(A problem is that I lag accepting pulls - sorry about that, too busy lately working on chameleon / macos).

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by koldo on Mon, 17 Feb 2020 10:06:24 GMT
View Forum Message <> Reply to Message

Mirek, I think we agree. I simply dislike with a fait accompli policy.

You know that if Klugier and you decide to change the workflow, that's fine with me. All I ask is not to receive private messages rudely demanding me to do things that have not been communicated to the community.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Mon, 17 Feb 2020 10:30:40 GMT
View Forum Message <> Reply to Message

Ah, sorry, I have missed the start of this thread. Yes, I agree that bazaar pull requests are sort of problem - this exactly the problem that prevented me from going git - access rights. But for what is worth, I have merged that XTEST pull request...

Mirek

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by koldo on Mon, 17 Feb 2020 10:53:24 GMT
View Forum Message <> Reply to Message

Thank you Mirek. I have uploaded the changes to SVN.

Subject: Re: Proposal: Transition from SVN to GIT within 2020.1 release cycle
Posted by mirek on Mon, 17 Feb 2020 11:02:02 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 17 February 2020 11:53Thank you Mirek. I have uploaded the changes to SVN.

Thats a mistake It got there by my accepting the pull.

Mirek