# TargetProcess + Mercurial Plugin

```
Functional requirements and scenarios.

Usage Scenarios

Profile setup
Users Mapping
Commit code
View diff
Protocols
Logging
Non-functional Requirements
```

Mercurial Plugin is very similar to Git or Subversion plugin <a href="http://help.targetprocess.com/">http://help.targetprocess.com/</a> integration api/git. It extracts commits from repository, bind these commits to TargetProcess User Stories/Bugs/Tasks, execute some actions like add comments, add time records, change state of User Story, Bug or Task.

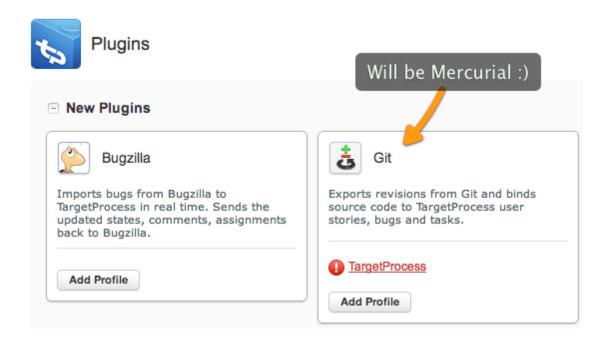
## Functional requirements and scenarios.

## **Usage Scenarios**

**Profile setup** 

Given Plugins section in TargetProcess
Then I see Mercurial Plugin with Mercurial logo, description, Add Profile button

[TBD - Design]



Given Plugins section in TargetProcess When I click Add Profile button in Mercurial plugin Then I see a plugin profile with fields

- Profile Name
- Repository path
- Login
- Password
- Import Start Date
- Users Mapping

## Git Integration

Exports revisions from Git and binds source code to TargetProcess user stories, bugs and tasks.

Profile Name
Should be unique. Can't be changed later.
TargetProcess
Repository Settings
Enter a full path to the repository Examples
git://127.0.0.1/trunk
Login
non
Password
•••
Export all the revisions starting from 02/01/2012
G Check Connection

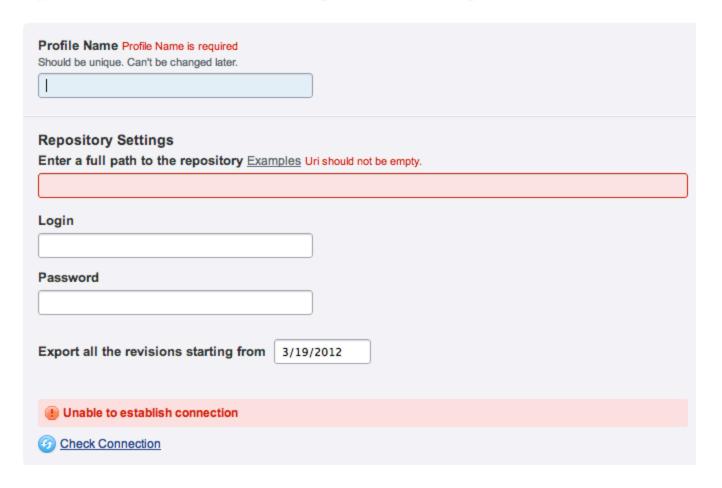
Given user filled all required fields When user clicks Check Connection link Then connection is checked And green success message is displayed

[TBD - screenshot]

Given all fields left blank When user clicks Check Connection OR Save profile Then required fields are highlighted And error message is displayed

#### Git Integration

Exports revisions from Git and binds source code to TargetProcess user stories, bugs and tasks.



#### **Users Mapping**

note: Mercurial user format: Mr. Johnson < johnson@smith.com>

Given Users Mapping section in Mercurial plugin profile

When I type Mercurial User email <a href="mailto:valja@somedomainwithlongname.com">valja@somedomainwithlongname.com</a>

And TargetProcess user email valja@pupkina.com

Then plugin should recognize <u>valja@somedomainwithlongname.com</u> Mercurial user and map it to <u>valja@pupkina.com</u> TP user

### User Mapping Enter user names here (Example: John Smith -> John Smith) Git User TargetProcess User maksim.linich@targetprocess.com Maksim Linich Andrey Щёткин shotkin@targetprocess.com ryabtsev@targetprocess.com Sergey Ryabtsev oleg@targetprocess.com Oleg Seriaga truhtanov@targetprocess.com Sergey Truhtanov alla@targetprocess.com Alla Pogotskaya abramchuk@targetprocess.com Vladimir Abramchuk Anna Shnyukova anna.dev@targetprocess.com Alex.Fomin@targetprocess.com Alex Fomin shitkin@targetprocess.com Aleksey Shitkin nadia.bulynia@targetprocess.com Nadia Bulynia sergey@targetprocess.com Sergey Demyachonak Uladzimir Piatryka vlad@targetprocess.com andrey.vaskovskiy@targetprocess.com Andrey Vaskovskiy alex@targetprocess.com Alex Tsayun eugene@targetprocess.com Eugene Khasenevich

Add more users

#### Commit code

Given Mercurial plugin is setup
When I commit source code (1 file A) and type commit message
#234 state:done time:3:0 comment:this bug is fixed
And navigate to Source tab for Bug with ID = 234
Then I see new revision that contains file A
And state of the bug is Done
And new time record is added with 3 hrs spent and 0 hrs remaining
And new comment added "this big is fixed"

#### Detailed commit messages format:

#ID (or id:ID) state:fixed	Change defect state
#ID (or id:ID) comment: text of your comment here comment: this is a second multi-line comment row1 row2	Add comments to defect
#ID (or id:ID) time:1	Add time spent only
#ID (or id:ID) time:1:12	Update time spent and time remaining
#ID (or id:ID), #ID (or id:ID) comment: will not be fixed state: invalid time:0.5	Update several defects at a time

Given Mercurial plugin is setup When I commit source code (1 file A) and type commit message "state:done time:3:0 comment:this bug is fixed" Then this commit is ignored by plugin

Given Mercurial plugin is setup

And there are 2 branches in repository (A and B)

When I commit source code (1 file X) into branch A OR branch B

And type commit message "#234"

And navigate to Source tab for Bug with ID = 234

Then I see new revision that contains file X

Given user with email <a href="mailto:valja@pupkina.com">valja@pupkina.com</a> in Mercurial is mapped to V.Pupkina TP user When Plugin discovers a commit with a user that has <a href="mailto:valja@pupkina.com">valja@pupkina.com</a> email

And commits contains message "#ID:23 state:fixed comm:xxx" Then this commits binds to an entity with ID 23 And state of this entity changed to Fixed And comment from V.Pupkina user is added with content "xxx"

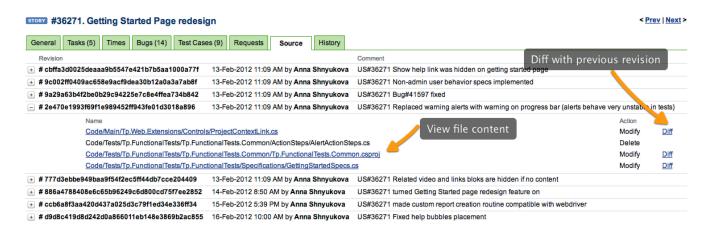
Given user with email notfound <a href="mailto:opupkina.com">opupkina.com</a> in Mercurial is NOT mapped to any TP user When Plugin discovers a commit with a user that has notfound <a href="mailto:opupkina.com">opupkina.com</a> email And commits contains message "#ID:23 state:fixed comm:xxx" Then this commits binds to an entity with ID 23 And state of this entity DOES NOT changed to Fixed And comment from V.Pupkina user is NOT added with content "xxx"

#### View diff

Source tab is already exist in TP as well as all UI for Mashups, it will be required to create Diff/ View mashup based on Git mashup and provide required commands in plugin BL side.

Given Source tab in a user story
When I click Diff on a file A inside revision X
Then I see two versions of file A in a popup window (X and previous revision)
And changes are highlightes (red, green, yellow)

Given Source tab in a user story
When I click Diff on a file A inside revision X
Then I see content of file A in revision X in a popup window



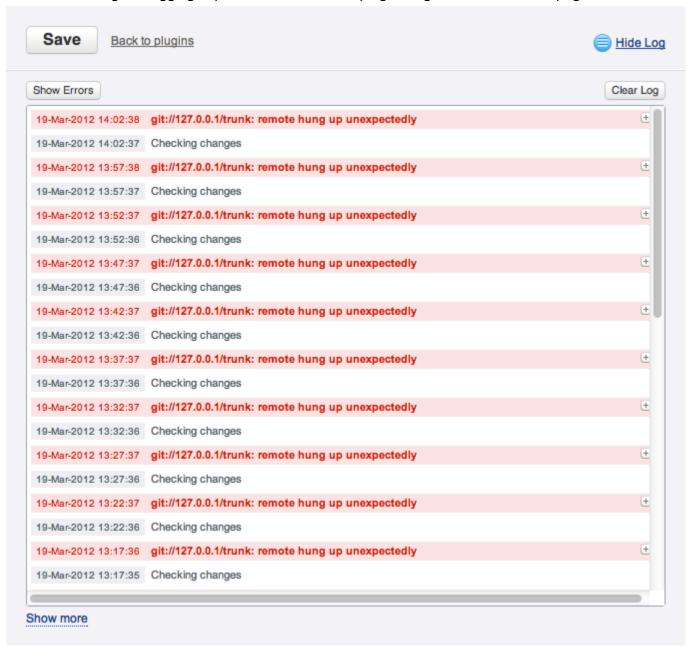
#### **Protocols**

3 connection protocols should be supported:

- HTTP
- HTTPS
- Windows fileshare (\\server\folder)

## Logging

There should be good logging implemented. Check Git plugin. Log is visible in Profile page.



So far we've used Mercurial in a \_very\_ basic manner, but we've talked about developing our strategies with it as soon as we start putting code in a production state.

We're not using branches much so far. Just a few user stories have been separated into their own branches, everything else is done in the default branch.

# **Non-functional Requirements**

- .NET lib for Mercurial <a href="http://mercurialnet.codeplex.com/">http://mercurialnet.codeplex.com/</a> (maybe there is something better, I am not sure).
- Mercurial Plugin should be able to import 100,000 revisions without hangs and crashes.
- Test Suite should be created to ensure that Mercurial plugin works correctly. Scenarios will be provided.