tl;dr: When comparing versions, the current_version is set to the core's latest migration instead of the plugin's latest.

Steps to reproduce:

1. Install any plugin with migrations, for example:

   cd redmine/plugins
   git clone https://github.com/ixti/redmine_tags.git
   bundle install
   rake redmine:plugins:migrate NAME=redmine_tags

2. Undo the last migration of that plugin:

   rake redmine:plugins:migrate NAME=redmine_tags VERSION=20151209153801

3. Migrate to the latest version again, but specifying the version this time:

   rake redmine:plugins:migrate NAME=redmine_tags VERSION=20180122193833

Expected behaviour: the 20180122193833 migration is executed.
Actual behaviour: the 20180122193833 is not executed.

This is because redmine core's latest migration is 20180923091603 which is greater than the plugin's 20180122193833.

The attached fix defines a `migrate` method for the `Redmine::Plugin::MigrationContext` class that makes sure to pass along the current plugin's latest migration. Using the default migrate method defined in activerecord executes a `current_version` that looks for the most current migration in the list of all migrations (core and plugins).

Note: I have not run any tests with this patch.
Although... I am not sure how to add a test that checks for this exact error.

#2 - 2019-03-31 07:06 - Go MAEDA
- Category set to Plugin API
- Target version set to Candidate for next major release

#3 - 2019-04-17 17:49 - Martijn Vernooij
- File redmine-4.0.2.plugin-upgrade.patch added

I noticed the same issue (with 4.0.2 though) and 'fixed' it another way.

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001-Fixes-plugin-migration-with-specific-version.patch</td>
<td>1.75 KB</td>
<td>2019-03-28</td>
<td>Anonymous</td>
</tr>
<tr>
<td>redmine-4.0.2.plugin-upgrade.patch</td>
<td>1.47 KB</td>
<td>2019-04-17</td>
<td>Martijn Vernooij</td>
</tr>
</tbody>
</table>