I propose a feature to visualize the results of table on the 'time_entries/report' page using bar chart. By visualizing the summary results in charts, we can more intuitively grasp the "time series" and "comparison result" of spent time.

The first chart shows the evolution of the total time according to the period (year, month, week, etc.), and the after second charts show the comparison of total time according to the summary axis (user, status, etc.).

History

#1 - 2020-03-12 10:37 - Go MAEDA
- Category changed from UI to Time tracking
- Target version set to Candidate for next major release

#2 - 2021-08-24 03:27 - Mischa The Evil
I think this could be a good improvement. Even more so now that we also have (similar) charts on the issues summary views introduced with #31418. Though, OTOH, I think that we should be cautious not to add too much charts to the core and not to add charts that aren't very useful for a large part of the community.

So, for example, I'd vote +1 for core integration of Chart.js ports of (some of) the charts historically provided by the [Graphs plugin](https://github.com/bradbeattie/redmine_graphs) by Brad Beattie (GitHub repository), but I'd vote -1 on such a proposal for specific charts supporting Agile/Scrum/etc. methodologies.

Besides the above mentioned considerations there is an additional potential issue that I've already reported as #35670 and more crudely as #35660 for the charts added per #31418. I think that if it's decided to integrate these changes, the mentioned issues should be handled first.

#3 - 2021-08-25 04:46 - Mischa The Evil
- Status changed from New to Needs feedback

On a second thought I start to doubt if the addition of these charts to the core is a good idea after all. To be precise, I start doubting whether the core should provide any charts at all.

The main reason for my doubt lies in the maintainability of all the code involved. To provide some crude numbers (using a patched stats rake task) for a (slightly patched) Redmine 4.1.x instance:

<table>
<thead>
<tr>
<th>Name</th>
<th>Lines</th>
<th>LOC</th>
<th>Classes</th>
<th>Methods</th>
<th>M/C</th>
<th>LOC/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllers</td>
<td>7816</td>
<td>5774</td>
<td>53</td>
<td>476</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Helpers</td>
<td>6060</td>
<td>4352</td>
<td>1</td>
<td>366</td>
<td>366</td>
<td>9</td>
</tr>
<tr>
<td>Models</td>
<td>17891</td>
<td>12748</td>
<td>95</td>
<td>1409</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>JavaScripts</td>
<td>5886</td>
<td>3167</td>
<td>0</td>
<td>204</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Libraries</td>
<td>19128</td>
<td>13813</td>
<td>151</td>
<td>1159</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Controller tests</td>
<td>29105</td>
<td>23819</td>
<td>65</td>
<td>1889</td>
<td>29</td>
<td>10</td>
</tr>
<tr>
<td>Helper tests</td>
<td>3467</td>
<td>2607</td>
<td>20</td>
<td>252</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>Model tests</td>
<td>31923</td>
<td>24395</td>
<td>146</td>
<td>2623</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>Integration tests</td>
<td>8905</td>
<td>6039</td>
<td>98</td>
<td>516</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>System tests</td>
<td>781</td>
<td>537</td>
<td>6</td>
<td>33</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>Extras tests</td>
<td>517</td>
<td>384</td>
<td>3</td>
<td>54</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>131479</td>
<td>97635</td>
<td>638</td>
<td>8981</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

Code LOC: 39854 Test LOC: 57781 Code to Test Ratio: 1:1.4
Now, compare those numbers with e.g. the same numbers for Basecamp 3 (from ~2016):

If you combine all this data and look at the respective size of the supporting development teams and the respective financial and economic capacity, you'll get an idea of what I'm talking about.

It might, as such, be a better choice to defer all charts/graphs to a single, dedicated plugin which could then include the repository graphs from the current core, the charts introduced in the core by #31418, the charts proposed in this issue and any additional graphs I referred to in my previous post.

I'm interested in hearing about what other people think about this...

1 that includes the long-time present repository graphs. It might be better to remove those completely and only provide a raw textual representation of this data.

2
diff of the patch.

diff --git a/lib/tasks/redmine.rake b/lib/tasks/redmine.rake
index db574df16..f036ac1f8 100644
--- a/lib/tasks/redmine.rake
+++ b/lib/tasks/redmine.rake
@@ -15,6 +15,9 @@
 # along with this program; if not, write to the Free Software
 # Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

+## Custom :stats task
+task :stats => "redmine:custom_stats"
+
+namespace :redmine do
+  namespace :attachments do
+    desc 'Removes uploaded files left unattached after one day.'
+    @@ -196,6 +199,30 @@ DESC
+  end
+end
+
+task :custom_stats do
+  require 'rails/code_statistics'
+
+  ::STATS_DIRECTORIES.clear
+  
+  +::STATS_DIRECTORIES << %w(Controllers app/controllers)
+  +::STATS_DIRECTORIES << %w(Helpers app/helpers)
+  +::STATS_DIRECTORIES << %w(Models app/models)
+  +# TODO: Enable once on Rails 6.1
+  +::STATS_DIRECTORIES << %w(Views app/views)
+  +::STATS_DIRECTORIES << %w(JavaScripts public/javascripts)
+  +# TODO: Enable once on Rails 6.1
+  +::STATS_DIRECTORIES << %w(Stylesheets public/stylesheets)
+  +::STATS_DIRECTORIES << %w(Libraries lib/)
+  +::STATS_DIRECTORIES << %w(Controller\ tests test\/functional)
+  +::STATS_DIRECTORIES << %w(Helper\ tests test\/helpers)
+  +::STATS_DIRECTORIES << %w(Model\ tests test\/unit)
+  +::STATS_DIRECTORIES << %w(Integration\ tests test\/integration)
+  +::STATS_DIRECTORIES << %w(System\ tests test\/system)
+  +::STATS_DIRECTORIES << %w(Extras\ tests test\/extra)
+  +
+  + CodeStatistics::TEST_TYPES << "Extras tests"
+end
+
# Load plugins' rake tasks

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Modified</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>bar-charts-in-the-spent-time-report.png</td>
<td>342 KB</td>
<td>2020-01-23</td>
<td>Takenori TAKAKI</td>
</tr>
<tr>
<td>bar-charts-in-the-spent-time-report.patch</td>
<td>5.38 KB</td>
<td>2020-01-23</td>
<td>Takenori TAKAKI</td>
</tr>
</tbody>
</table>