Gantt - Board synchronization

Introduction

Modules in our app are parts of a bigger project portfolio management powerhouse. Therefore they must be integrated in many aspects. The purpose of this documentation piece is here to explain in which way our Gantt module synchronizes with the Board.

There are several aspects to the Board and the Gantt integration:

- updating the task period as an effect of moving tasks between iterations
- updating the task period as an effect of changes in Jira Sprints
- visualization of iterations on the Gantt chart
- visualization of favorite iterations as markers on the Gantt chart
- links sharing.

Updating the task period as an effect of moving tasks between iterations

This type of synchronization can be turned on in the Program Configuration:

Board configuration 🔗

Synchronization with Jira

Board can work in 3 different modes: manual (no synchronization with Jira Agile sprints), small team (synchronizes bi-directionally with Jira Agile Sprints) and large team (syncs with Jira Sprints also, but recommended for adopters of The Scaled Agile Framework® - available exclusively in BigPicture Enterprise).

Task period synchronization

Board module can synchronize the task period. Whenever this option is enabled and we move a task on Board, the plugin will automatically adjust and recalculate start and end dates of that moved task to those of the corresponding cadences.

<table>
<thead>
<tr>
<th>Synchronization mode</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simple drag and drop. No synchronization with Jira Agile sprints</td>
</tr>
</tbody>
</table>

Once turned off, the movement between iterations never results in a task period update.

Once turned on, the task period is adjusted to iteration’s period if a user performs an action on the Board. A movement between iterations which is an effect of a movement between matched Jira sprints (Small or Large Team Synchronization Mode) will not result in the task period update until you set it accordingly (read more: Updating the task period as an effect of changes in Jira Sprints).

If task’s End Date is outside iteration period, that task’s assignment is considered as incorrect. It is indicated by the red calendar on the task card (visible only when Warnings are enabled).
A special dialog box appears whenever the user clicks on the red calendar info. User can choose between two ways of resolving such a conflict:

1. To adjust the period of a task to match the current iteration.
2. To move a task to the iteration which period contains task’s end date. This option can be disabled if there is no such an iteration. Please note that such an action executes an actual movement between iterations so if the task period synchronization is turned on, then the task period will be
adjusted to perfectly fit that iteration right after a manual adjustment was performed.

What do we mean by "adjusting an iteration date"? There are several scenarios which we should go through and they are described below.

Precise alignment

You are provided with two strategies of adjustment task period. The first strategy called "precise alignment" assumes, that you work in a fully agile manner and task duration is not significant as long as the team declares to complete the task within an iteration. Once you check that option, the application will always align task dates (start and end dates) with iteration dates by synching task period.

**Task period synchronization**

Board module can synchronize the task period. Whenever this option is enabled and we move a task on Board, the plugin will automatically adjust and recalculate start and end dates of that moved task to those of the corresponding cadences.

- Task period synchronization
- Synchronize with Jira changes

**Adjustment strategy**

- **Precise alignment**
  Task dates (start and end dates) will be aligned with cadence dates by synching its period
- **Smart adjustment**
  Task dates (start and/or end date) will be always in-between cadence period (trying to avoid changing task duration)

**BEFORE THE ADJUSTMENT**

Task period before the adjustment does not influence the result of the adjustment.

**AFTER THE ADJUSTMENT**

No matter task period was, after the adjustment task period always aligns precisely with the period of iteration it is assigned to at the moment.

<table>
<thead>
<tr>
<th>r, 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 17 18 19 20 21 22 23 24 25 26 27 28 29 30</td>
</tr>
<tr>
<td>Cadence 1</td>
</tr>
<tr>
<td>EVII-13 - New sword</td>
</tr>
</tbody>
</table>

Smart adjustment

The second adjustment strategy called "smart adjustment" assumes that base task duration is intended. It means that application adjusts start and/or end date of the task so that they are always in-between iteration period, but it tries to avoid changing task duration. Take a glance at the following scenarios to understand how it works and select that option if you wish to take advantage of that algorithm.

1) Whole task period is contained in the iteration period   do nothing.
2) Whole task period is located before the iteration period (Task period is not longer than an iteration period) task will be moved to start at the beginning of the iteration and the period length will remain unchanged.

3) Whole task period is after the iteration period (Task period is not longer than iteration period) task will be moved to end at the end of the iteration, period length will remain unchanged.
4) Whole task period is located before the iteration period (Task period is longer than the iteration period). task will be moved to start at the beginning of the iteration and end at the end of the iteration.
5) Whole task period is located after the iteration period (Task period which is longer than iteration period) task will be moved to start at the beginning of the iteration and end at the end of the iteration.

BEFORE THE ADJUSTMENT

<table>
<thead>
<tr>
<th>9</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BEFORE THE ADJUSTMENT

<table>
<thead>
<tr>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AFTER THE ADJUSTMENT

6) Whole iteration period is contained within the task period task will be fitted in, meaning it will be moved to start at the beginning of the iteration and to end at the end of the iteration.

BEFORE THE ADJUSTMENT
7) Part of task period is located before the iteration period and part of it overlaps iteration period. (Task period is not longer than the iteration period) the task will be moved to start at the beginning of the iteration, and the period length will remain unchanged.
8) Part of the task period overlaps iteration period, part is after (Task period is not longer than the iteration period) task will be moved to end at the end of the iteration, period length will remain unchanged.
9) Part of a task period overlaps iteration period, and part of it is outside of the iteration (Task period is longer than iteration period) task will be fitted in, meaning it will be moved to start at the beginning of the iteration and end at the end of the iteration.
10) Part of a task period overlaps iteration period, and a part of it is located outside (Task period is longer than the iteration period) task will be moved to start at the beginning of the iteration and end at the end of the iteration.
Corner cases

Under some circumstances, Gantt's Task Mode may be shifted whenever changes are made in the Board module. For example whenever we choose Data -> Task Mode -> Auto, it appears to work as intended. But then whenever we move a Parent to another iteration on the Board - Subtasks do not stay within the Start and End Date of the Parent. Also if you look at the Subtask's Task Mode you will notice that the system automatically changes it back to the Manual instead of keeping it in the Auto setting.
Updating the task period as an effect of changes in Jira Sprints

Once you turn on 'Task period synchronization', you are allowed to decide whether you want update task period based on changes performed out of BigPicture (eg. by modifying value of "Sprint" field on issue detail page).
Task period synchronization

Board module can synchronize the task period. Whenever this option is enabled and we move a task on Board, the plugin will automatically adjust and recalculate start and end dates of that moved task to those of the corresponding cadences.

- **Synchronize with Jira changes**

Adjustment strategy

- **Precise alignment**
  - Task dates (start and end dates) will be aligned with cadence dates by syncing its period.
- **Smart adjustment**
  - Task dates (start and/or end date) will always be in-between cadence period (trying to avoid changing task duration).

Save  Discard

It means, that when you check an option 'Synchronize with Jira changes', every movement of the task between Sprints in Jira (in Small or Large Team Synchronization Mode) results in the task period update (in line with the rules mentioned above).

Visualization of iterations on the Gantt chart

Please keep in mind that moving tasks on the Gantt chart will never move tasks between iterations on the Board. Iterations on the Gantt are only a visualization and not the drop area.

Visualization of 'Favourite' iterations as a marker on the Gantt chart

If the user marks an iteration as a 'Favourite' (small star in the iteration header), then it will be visible among markers on the Gantt chart.
Sharing links

Board uses the same linking system as Gantt does. Therefore, whenever one creates a link on the Board, it will be automatically visible on the Gantt chart and vice versa.

Modules in our app are parts of a bigger project portfolio management powerhouse. Therefore they are must be integrated in many aspects. The purpose of this documentation piece is here to explain in which way our Gantt module synchronizes with the Board.

HIGHLIGHTS

- Introduction
- Updating the task period as an effect of moving tasks between iterations
  - Precise alignment
  - Smart adjustment
  - Corner cases
- Updating the task period as an effect of changes in Jira Sprints
- Visualization of iterations on the Gantt chart
- Visualization of 'Favourite' iterations as a marker on the Gantt chart
- Sharing links