Choosing Task Mapping Strategy

A Task Mapping strategy - is a set of rules, which define, how the issues from Jira are mapped to tasks from FreshBooks.
Different Task Mapping strategies give the ability to map issues to tasks differently:

1. **"Each Issue is a Task"**— a strategy, which makes the plugin to create a separate task for each issue (found in each of the projects [or issue filters], given in the ProjectMapping). The name of the created task is able to choose from a section **"Choose FreshBooks Task name"**. The description of the issue becomes the description of the created task. Whenever the issue is updated - the task is updated with it, accordingly. Thus:

    1. An issue created -> a task is created
    2. An issue updated -> a task is updated
    3. An issue deleted -> a task is deleted

    Also, all the worklogs, logged to the mapped issue are handled as follows:

    1. A worklog created -> a time entry created
    2. A worklog updated -> a time entry updated
    3. A worklog deleted -> a time entry deleted

    Lets prepose, that we have the following project mapping:

Jira project or filter	FreshBooks project
JiraProject	FbProject

    Also, assume that we have the following user mapping:

Jira user	FreshBooks staff
JiraUser	FbStaff

    Suppose, that we have created an issue in the **JiraProject** - **JiraIssue**.
    The plugin finds the corresponding FreshBooks project - **FbProject,** and creates a task inside it - the **FbTask**.

    The **JiraProject** has an issue - **JiraIssue** .
    Let's pretend, that **JiraUser** worklogged (made a work log) on this issue: the plugin finds this worklog,
    and creates a time entry with notes, taken from worklog description, and time, taken from worklog spent time.
    Plugin puts this time entry under the project, which corresponds to the **JiraProject** - **FbProject**,
    Under the task, which corresponds to the **JiraIssue** - **FbTask**.
    The author of this worklog is the staff, which corresponds to the author of worklog (**JiraUser**) - **FbStaff**.

2. "**Issues are a task (Accumulate Worklogs)**" — a strategy, which makes the plugin to map multiple issues to a single task with a help of Task Mapping. Worklogs come to the FreshBooks staff, corresponding to the assignee of the issue (work to which was logged). If no corresponding staff found, the default FreshBooks user is the author of time entry. There (in the task Mapping), a user defines correspondence of a Jira Issue filter to a task. Whenever an issue, fitting a filter is created - the corresponding task becomes updated. This strategy does not split worklogs for the issue by users - all the worklogs for the issue are mended together into a time entry, logged to FreshBooks under FreshBooks user, which corresponds to the assignee of the issue which worklogged.

    1. An issue created -> a task is updated
    2. An issue updated -> a task is updated
    3. An issue deleted -> a task is updated

    Also, all the worklogs, logged to the mapped issue are handled as follows:

    1. A worklog created -> if there are no more worklogs on the issue, a time entry is created, otherwise, a time entry, corresponding to these worklogs is updated
    2. A worklog updated -> a time entry updated
    3. A worklog deleted -> if there are no more worklogs on the issue, a time entry is deleted, otherwise, a time entry, corresponding to these worklogs is updated

    All the worklogs for the issue are forming one time entry. Such a time entry is created for the task (in the project), which corresponds to the filter, by which this task is found.

    Lets prepose, that we have the following task mapping:

Jira issue filter	FreshBooks task	FreshBooks project
All issues from JiraProject	FbTask	FbProject

    Also, assume that we have the following user mapping:

Jira user	FreshBooks staff
JiraUser	FbStaff

Suppose, that we have created an issue, which fits the filter **"All issues from JiraProject"** (i.e. we created an issue in the **JiraProject**) - **JiraIssue**.
The plugin finds the corresponding FreshBooks task - **FbTask**,
and updates it for Freshbooks project from the task map - **FbProject**.

The filter **"All issues from JiraProject"** has an issue, which fits it - **JiraIssue**.
Let's pretend, that **JiraUser** is the assignee of **JiraIssue**.
Any user from jira worklogged (made a work log) on this issue:
the plugin compounds time from all the worklogs in **JiraIssue**
and creates a time entry (let's name it BazTimeEntry):

- with notes, taken from **JiraIssue** summary
- the date of the time entry is computed out of dates for worklogs - the maximal out of all is taken
- author, gotten from user mapping (staff, corresponding to the **JiraIssue** assignee **JiraUser**) - **FbStaff**

This time entry is created for **BazTask** from **BazProject**.
Next time someone worklogs against **JiraIssue**,
the **BazTimeEntry** is updated with new worklog information.

3. **"Issues are a task (Process Worklogs individually)"** — a strategy, exactly the same as the "Issues are a task (Accumulate Worklogs), but split worklogs for the issue, all worklogs.

4. **"Each Issue is a Task (with changed field mapping)"** — a strategy, exactly the same as the “Each issue is a task”, but with changed IssueToTask field mapping, where Issue Summary goes to Description field of the created task.

For the strategies **"Each Issue is a Task"** and **"Each Issue is a Task (with changed field mapping)"** there is an additional function **"Choose FreshBooks Task name"**, with several options, which provides an ability to choose display name for task:

- ***Jira Issue key*** — The Issue Key goes to Task Name field.
- ***Issue summary*** — The summary from issue becomes a name.
- ***Jira Issue key + Issue summary*** — Previous two fields separated by white space.
- ***Freshbooks Custom Field value*** — The value of the FreshBooks Custom field for the Issue. If there is no Freshbooks Custom Field or Freshbooks Custom Field value the Jira Issue key will be chosen by default.