**Pattern Name and classification/type**: KANBAN

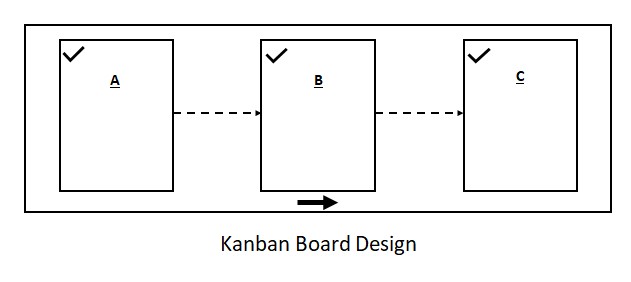
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: Using this pattern we will have a full vision of the status of the tasks of a project, handling how they evolve and monitoring them.
* **Solutions**: Firstly, cards are storaged in “to do” bucket as tasks are storaged in a backlog. Tasks will flow through the buckets as long as they are in progress or finished in real life. Kanban board will represent the status of every task at any given moment.
* **Motivation (Problem)**: Controlling the evolution of tasks.
* **Context/Applicability**: Enterprises with projects structured by steps and evolutive tasks.
* **Issues/Consequences/resulting context**: We can have multiple Kanban boards in parallel (for different projects) or concatenated boards (when a task is finished in one board, it passes to the “to do” bucket in next board).

**Pattern model:**

* **Graphic representation**:

****

* **Components description**: There are three buckets that represent three different posible status of a task: “to do”, “doing” and “done”. Cards represent process’ tasks that flow between those buckets according to its evolution in the Project context. “Doing” bucket could be extended in more than one bucket, like for example “
* **Participants description**: Each employee has to move its assigned tasks through the different buckets by the time that he/she works or finishes it. The project manager is the responsible of adding the tasks and assigning them to the different employees.
* **Collaboration/behaviour**: Regulardly, cards follow a linear flow, from left bucket (“to do”) to the right one (“done”) passing by “doing” step.

**Related Patterns**: -

**Examples/Known uses**: Scrum, factory pipeline.

**Implementation**: Lists aren’t necessary as such, if your tool doesn’t provide them. You can use columns or buckeets for storing the tasks as long as tasks can flow between those lists. Alike, tasks can be represented by tickets for example, in stead of cards.

**Instances/Sample code**: <https://trello.com/b/hSu4ioNL/kanban-template>

**Pattern Name and classification/type**: Categorized Information

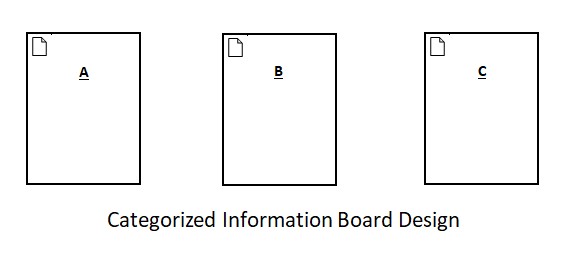
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: Using this pattern we will organize all the information of our project by the field that we need (topic, departament, employee…), helping to find it easily.
* **Solutions**: Each bucket will be used as a list of notes, where the cards will be storaged. They will separate the information according to one criteria selected by the Enterprise according to the concrete context (topic, departament…). Cards will be assigned to one and only one bucket, statically, and displaying the information depending on the tool’s features (only text, tags, date…).
* **Context/Applicability**: Static information based context or projects.
* **Motivation (problem)**: Having an organized and quick access to the information of a project.
* **Issues/Consequences/resulting context**: It’s unusual to concatenate multiple boards with this pattern, because cards will not flow between them. They would act like parallel projects without interaction between their components.

**Pattern model:**

* **Graphic representation**:



* **Components description**: Buckets will be used as containers of information. This information will be collected in the cards as text. Details and additions (use of tags, variables of each card…) will be determined by the software used.
* **Participants description**: It depends on the purpose of use of the board. For example, the project manager should manage all the cards, meanwhile an employee only can act (edit, add tags, write…) over the information of its assigned bucket.
* **Collaboration/behaviour**: Regulardly, cards doesn’t flow between buckets, so the only possibility of interacting with board’s components is editing the information of the cards or adding features as tags or dates.

**Related Patterns**: -

**Examples/Known uses**: notes clasified by topic or information structured by category.

**Implementation**: Cards of information can be as complex and detailed as the tool allows. In some workstream collaboration tools you can only write text as information, but in some other software you can use tags, dates, you can assign the card to a concrete person… Those functionalities don’t change the essence of the pattern, they only add richness to its usage.

**Instances/Sample code**: <https://trello.com/templates/education/syllabus-template-MEj9Q6SK>

**Pattern Name and classification/type**: Information Lifecycle

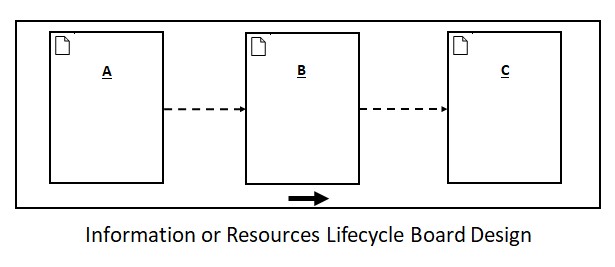
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: This pattern will allow you to view all the information of your project structured by its status while you will be able to control its evolution too.
* **Solution**: Having a full view of the evolutive information of your project, represented by cards, structured by its corresponding step in the context of the project, divided by lists.
* **Context/Applicability**: Projects with a lot of information that changes with the evolution of the processes.
* **Motivation (problem)**: Representing all needed information of a process differentiating it according to the status of the information.
* **Issues/Consequences/resulting context**: As in Kanban, we could concatenate multiple Information Lifecycle boards as a pipeline, joining many processes from different boards.

**Pattern model:**

* **Graphic representation**:



* **Components description**: Lists frequently represent steps of a whole process while cards represent information that flow between those lists.
* **Participants description**: For a lonely use, you will manage all cards and you will move it all. If you are working in a team, probably you will be only responsible of your cards (the information corresponding to your part of the whole project).
* **Collaboration/behaviour**: Cards will flow between lists as the information evolve in real life and it passes from one step (list) to another.

**Related Patterns**: -

**Examples/Known uses**: An information pipeline or a BPMN.

**Implementation**: The use of this pattern is quite similar to Kanban, but representing information instead of tasks. We can represent almost any type of information with this board, with labels, images or tags, depending on the features of the specific software that we are using.

**Instances/Sample code**: <https://trello.com/templates/sales/bike-repair-pipeline-9clNtU6v>

**Pattern Name and classification/type**: Process Tasks

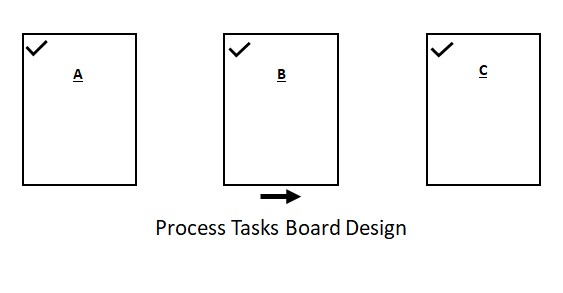
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: Using this pattern, we will have a list of tasks for completing a whole project.
* **Solution**: We will divide tasks by steps as we separate the steps to follow to make a cooking recipe.
* **Context/Applicability**: Tasks based projects divided by steps (when we want to determine the tasks of each step).
* **Motivation (problem)**: Representing all the necessary tasks to complete a process structuring them step by step.
* **Issues/Consequences/resulting context**: With this pattern we are not representing the evolution of the tasks of a project as in kanban. Here we are giving all the steps to follow (tasks) to complete a process but without having a direct relationship with how we are doing those tasks in real life.

**Pattern model:**

* **Graphic representation**:



* **Components description**: Cards will represent tasks and lists will represent the different steps in which we divide the whole project.
* **Participants description**: As tasks do not flow between lists, it is not necessary human interaction with the board. We will add tasks one time, and they will not need changes. Furthermore, we can add more cards (or delete one) according to our needs during the process, but it is not allowed to change cards (in this case, we are using this board as a kanban, we are introducing card flow).
* **Collaboration/behaviour**: Cards will not move between lists, so the only possible change is about adding information to the card (adding labels, tasks or assignments to a task).

**Related Patterns**: -

**Examples/Known uses**: Waterfall engineering process or a recipe.

**Implementation**: Use it like guide with steps to follow, specify the tasks as much as possible (low level description, dividing it as much as possible) and do not skip any task or step for completing the project.

**Instances/Sample code**: <https://trello.com/templates/support/building-a-customer-feedback-program-C2ZIBftB>

**Pattern Name and classification/type**: Categorized Tasks

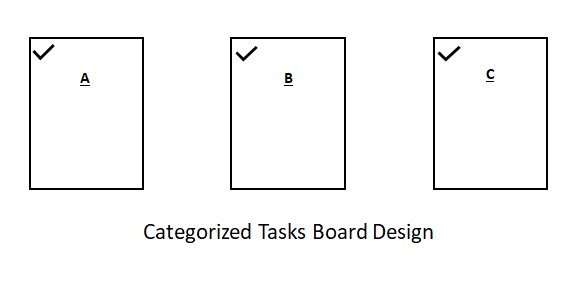
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: Using this pattern we will organize tasks by topic, normally for lonely use (not for team projects).
* **Solution:** We will structure our board according to the different topics of our specific problem, dividing the board in lists (one for each topic) and we will store the tasks of each topic in their corresponding list.
* **Context/Applicability**: Topics or steps mutually exclusive and tasks without evolution control (as in a kanban for example).
* **Motivation (problem)**: Representing tasks without relationship between them and structuring them with a meaning (topics for example).
* **Issues/Consequences/resulting context**:If there is some relationship between the lists in which we are dividing our problem, we are using Process Tasks board. Here the different columns of our board are unrelated.

**Pattern model:**

* **Graphic representation**:

****

* **Components description**: Lists represent topics or excluding steps and cards represent tasks that belong to one and only one list.
* **Participants description**: It is frequently designed for lonely usage, where the only person that uses the board can add, delete and edit lists and cards as he/she needs.
* **Collaboration/behaviour**: As cards do not evolve or flow between lists, it is not necessary human intervention with the board once created, beyond editing them if necessary.

**Related Patterns**: -

**Examples/Known uses**: Getting Things Done (GTD) – A productivity methodology

**Implementation**: It is really important to choose an appropriate division of lists (they have to be exlusives between them) and specify low level tasks for easing their realization in real life.

**Instances/Sample code**: <https://trello.com/templates/productivity/8-creative-habits-L6SJcYm6>

**Pattern Name and classification/type**: Assigned Information

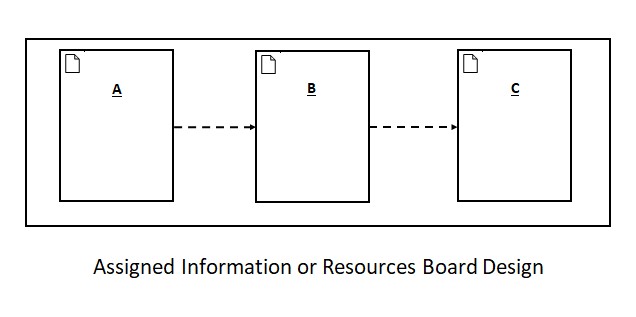
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: Using this pattern we will have a view of the information of the project at each moment. This information can evolve, change its status and move to another list.
* **Solution**: We will store the information in lists and we will move it through the different lists according to the needs of the specific problem, determining the meaning of the card flow in that context (because there is not an established precedence between lists).
* **Context/Applicability**: Projects with non-static information and without an established order/precedence between lists (topics or phases for example).
* **Motivation (problem)**: Representing information divided in not specifically ordered lists (the flow between the different lists is not determined a priori as if we have ordered steps of a project where the requirements phase is always before development phase).
* **Issues/Consequences/resulting context**: We can concatenate different Assigned Information boards as an information pipeline.

**Pattern model:**

* **Graphic representation**:



* **Components description**: Cards represent information that can flow between lists. Lists are divisions without a specific relationship order between them (for example, the different departments of an enterprise do not have an a priori established movement order).
* **Participants description**: We can have two scenarios: somebody moves all board’s cards (for example, the boss is responsible to move each worker to its assigned department) or each person move its own cards.
* **Collaboration/behaviour**: Cards can flow between lists without a priori restrictions (they should be specified by the context of the problem)

**Related Patterns**: -

**Examples/Known uses**: Representing an organizational enterprise structure with employees and departments where the workers can change their deparment.

**Implementation**:

**Instances/Sample code**: <https://trello.com/templates/personal/book-clubs-tP0ROvp8>

**Pattern Name and classification/type**: Assigned Tasks

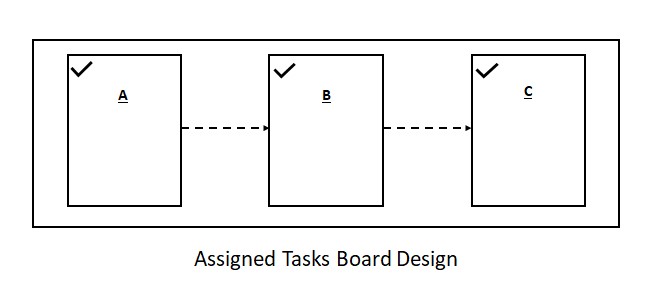
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: This pattern is mainly intended for organizing the task of a team, assigning them to each employee/group member (other uses can fit with this pattern, we only propose an example for explaining it)
* **Solution**: Each list will represent a team member (or an unordered phase, without an established precedence with each other) and tasks will be represented by cards.
* **Context/Applicability**: Team projects when we want to focus on the assignment (not on the status of the task as in kanban) or when the evolution of the task is not a priori established (as in kanban, where we have a predetermined and ordered flow)
* **Motivation (problem)**: Organizing tasks without a pre-established flow (not a pipeline or an ordered sequence)
* **Issues/Consequences/resulting context**:

**Pattern model:**

* **Graphic representation**:



* **Components description**: Lists store tasks (cards) without order or precedence relationship between the lists, allowing the flow of tasks between them.
* **Participants description**: The project manager can distribute the tasks between his/her employees (moving cards from one list (employee) to another) or each employee is responsible of managing its own tasks and moving them to the next employee when finishes.
* **Collaboration/behaviour**: Cards can flow between lists without an a priori established order (because there is not order precedence between lists). Tasks might not follow always the same path.

**Related Patterns**: -

**Examples/Known uses**: Enterprise task assignment to employees (when tasks are not finished by one employee and they have to pass by more than one person)

**Implementation**: It is highly recommended to have some instantiation rules that specify the flow of cards in the board in your specific problem context. Just because there is not order precedence between lists does not mean that the card flow is free and unrestricted; you must specify it.

**Instances/Sample code**: <https://trello.com/templates/team-management/team-tasks---5-things-workflow-DQDXYzN7> (in this specific example, card flow between lists does not have much sense, the idea is moving task from one employee to another, assigning them team tasks, not his personal objectives)

**Pattern Name and classification/type**: Ordered Information

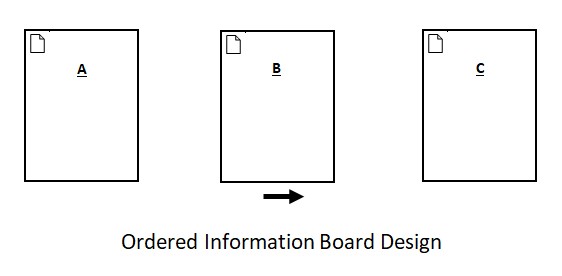
**Also known as/alias**: -

**Pattern definition:**

* **Overview/Description**: Using this pattern we will have a full vision of the static information of the project ordered by categories or steps (related between them by a specific precedence).
* **Solution**: This pattern will be used similar to Process Tasks pattern, but storing information instead of tasks, where cards are in one and only one list which is related with the others (order or precedence relationship).
* **Context/Applicability**: Projects with many information and related clasifications of it (order, sequenced stages…) or calendar usage (lists represent temporal divisions as days, weeks or months).
* **Motivation (problem)**: Clasify ordered, related and static information.
* **Issues/Consequences/resulting context**: In this pattern we do not have isolated notes by topic, as in categorized information. Here, the different lists have a precedence or order relationship, although cards do not flow between them.

**Pattern model:**

* **Graphic representation**:



* **Components description**: Information will be represented in cards, storaged in lists that can represent order, precedence, days, weeks, different stages of a project…
* **Participants description**: Somebody adds the information to lists, and as it will not evolve (flow is not allowed), human intervention will be unnecessary once created.
* **Collaboration/behaviour**: Cards do not flow between lists, so we use this pattern like Process Tasks pattern, but storing information instead of tasks.

**Related Patterns**: -

**Examples/Known uses**: Calendar

**Implementation**: This pattern does not allow card flow, so if you are going to use it for a calendar, for example, when you place a card for a due date, you will not be able to move it to another day (if you do not do it in time or if you reorganize your agenda).

**Instances/Sample code**: <https://trello.com/templates/marketing/annual-email-marketing-calendar-VCaIAYFR>