

Strategic plan [Editor]

Base

Information architecture

Current state and problem

App bar

Toolbar

Content area

Right panel

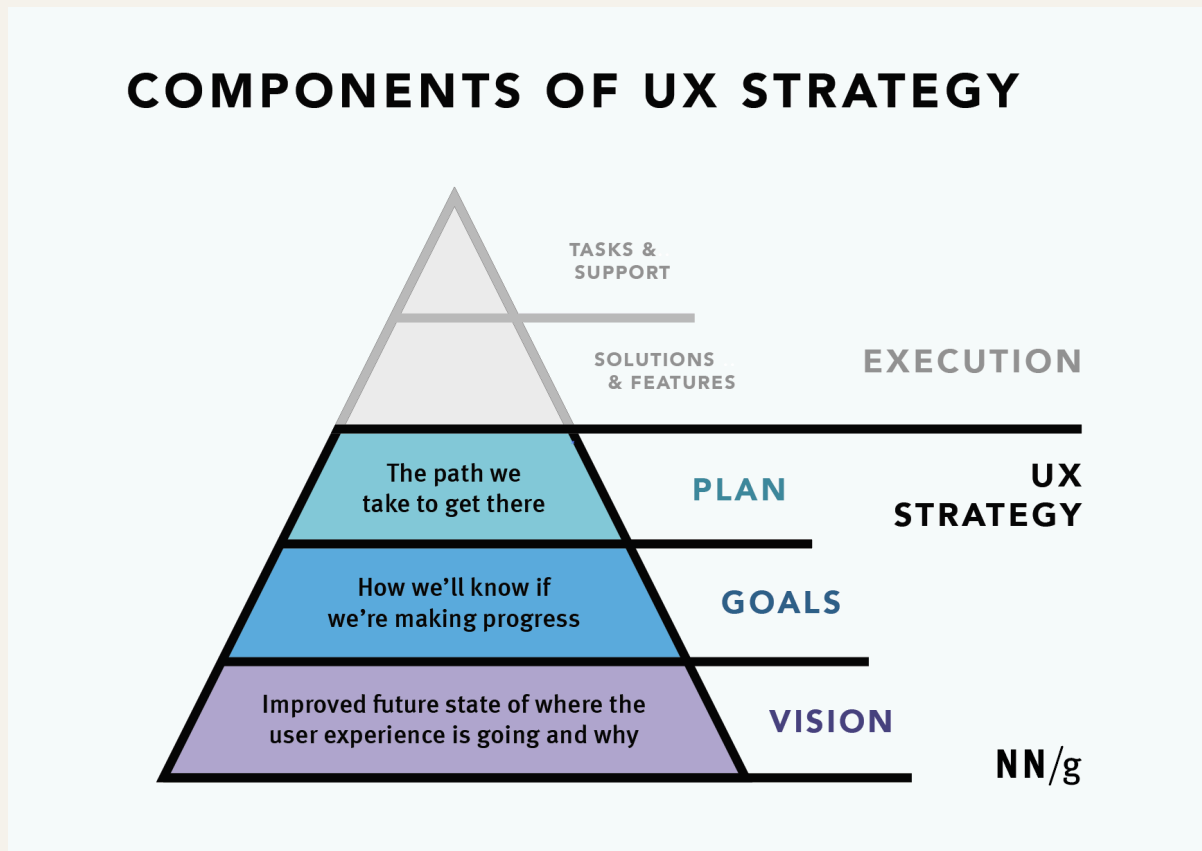
General approach

Solutions

Questions

Base

Creating the UX strategy for the PandaDoc's Editor, we need to cover certain understanding levels to move forward.



We start with **Vision** and describe **Goals**. According to those inputs, we're able to create a **Plan**. In addition, there are plenty of **heuristic issues** we need to consider. We are talking about design/technical debt and information architecture problems.

Speaking about the **Vision**, it's important to clarify the **primary directions** we have and want to improve.

WYSIWYG editor	There are plenty of options for advanced document formatting we might want to add.
Imported files conversion to PandaDoc native format	Users upload local files for further editing in PandaDoc
Operations with uploaded PDF files	Users import PDF files, making modifications if needed
Template generator based on user needs	The approach is when the system proposes necessary templates according to user requests
Workflow settings	Signing order, approval flow, etc.

How does our current planning approach look?

1. Metrics definition we would like to increase.
2. We compose short-term goals that will help us to achieve the desired result.
3. We arrange the prioritisation of the tasks according to user requests, available data, and estimated effort (not always).

The current state of the **Editor** reflects all consequences of the feature-based approach. This module tries to cover dozens of scenarios lacking the main direction.

As a result, we are constantly facing the following problems:

- Solutions developed by different teams aren't aligned.
- All changes try to be the most prominent thing on the screen because it's new, and we desperately want users to notice it.
- Solving particular interface problems, you never know how exactly the module is intended to expand.

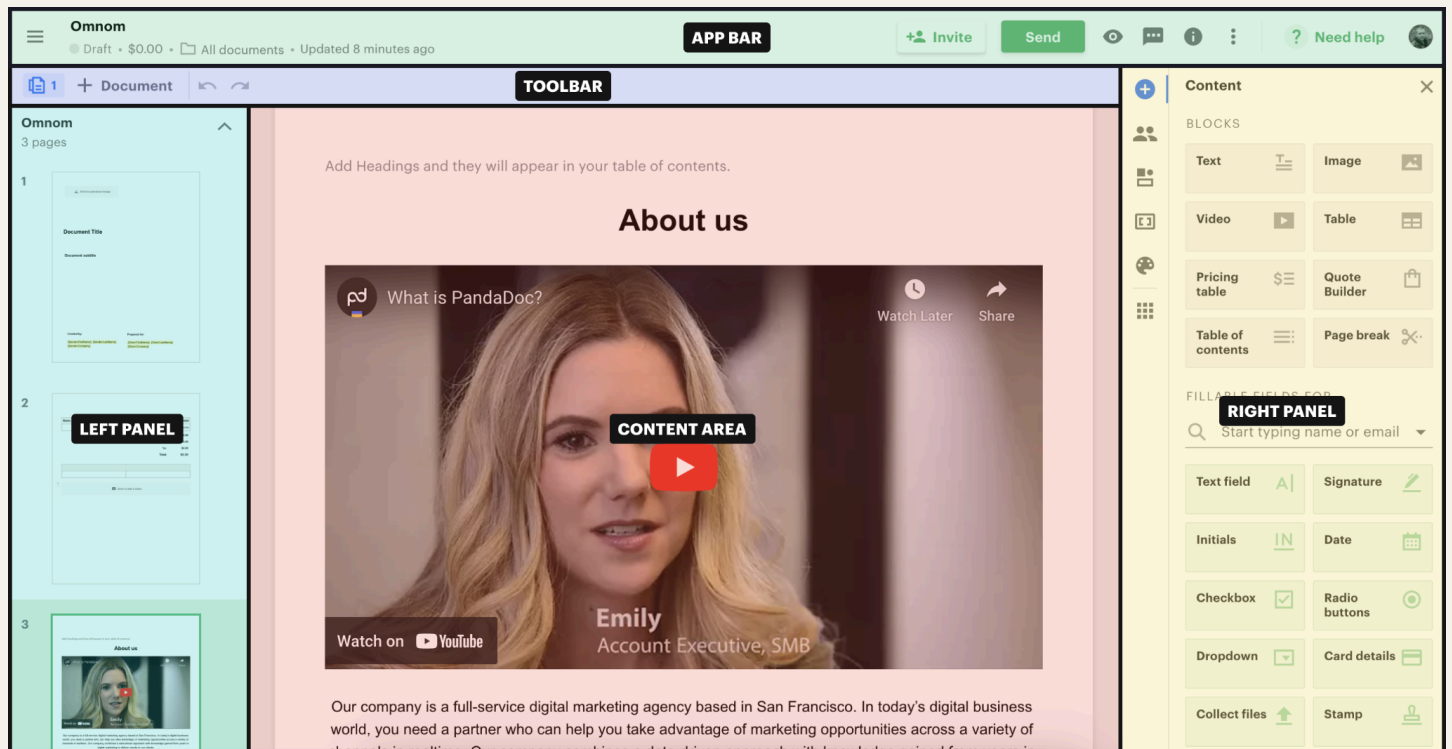
To fix that, we can start with two steps:

1. Define existing architectural problems.
2. Describe the basic principles we should consider when making any interface changes.

Information architecture

Current state and problem

Let's set up our vocabulary of detailed document view sections.



The **Application bar (App bar)** contains general information about the document, primary actions, and some settings.

The **Toolbar** is mainly used for Bundle creation, Undo/Redo actions, and textual block properties (when they are selected).

The **Content area** is the actual document users create.

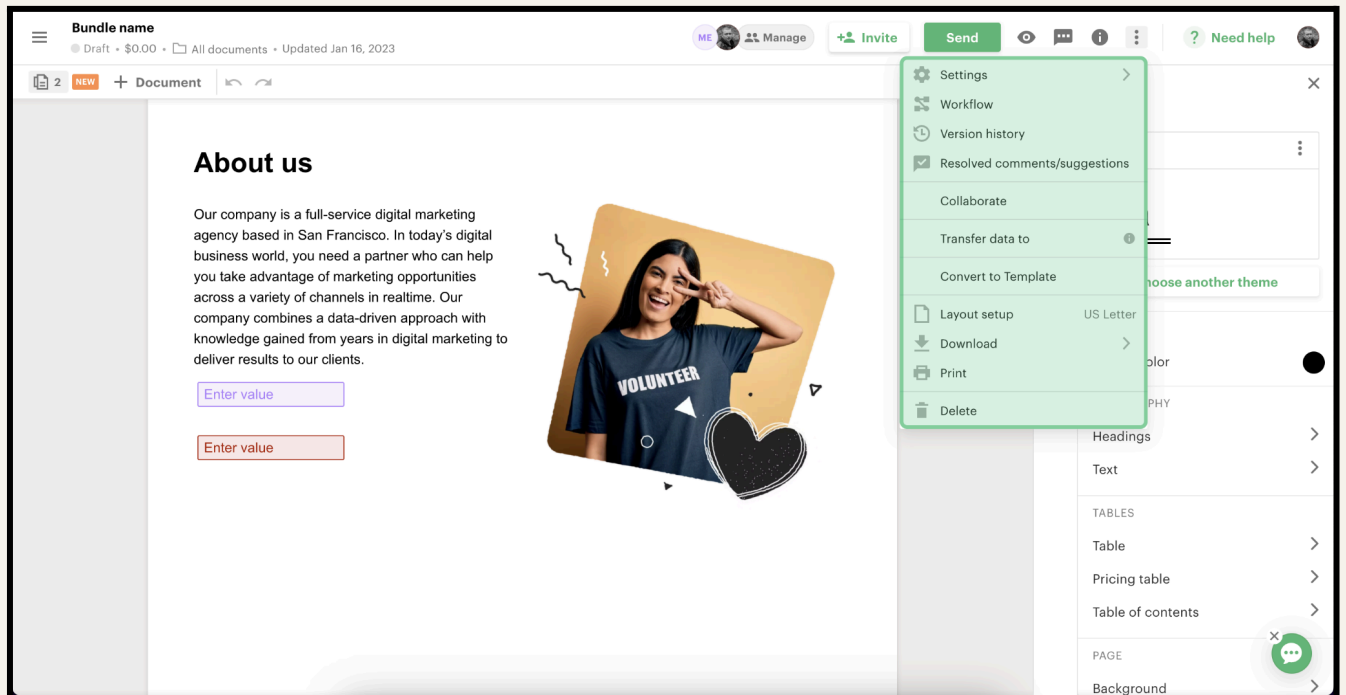
The **Right panel** depends on the selected item within the toolbar and contains relevant properties for block management, recipients, workflow, and overall document settings.

The **Left panel** contains a page preview.

We will highlight some prominent issues from different areas and show their connection. The numbering of problems doesn't correlate with their importance.

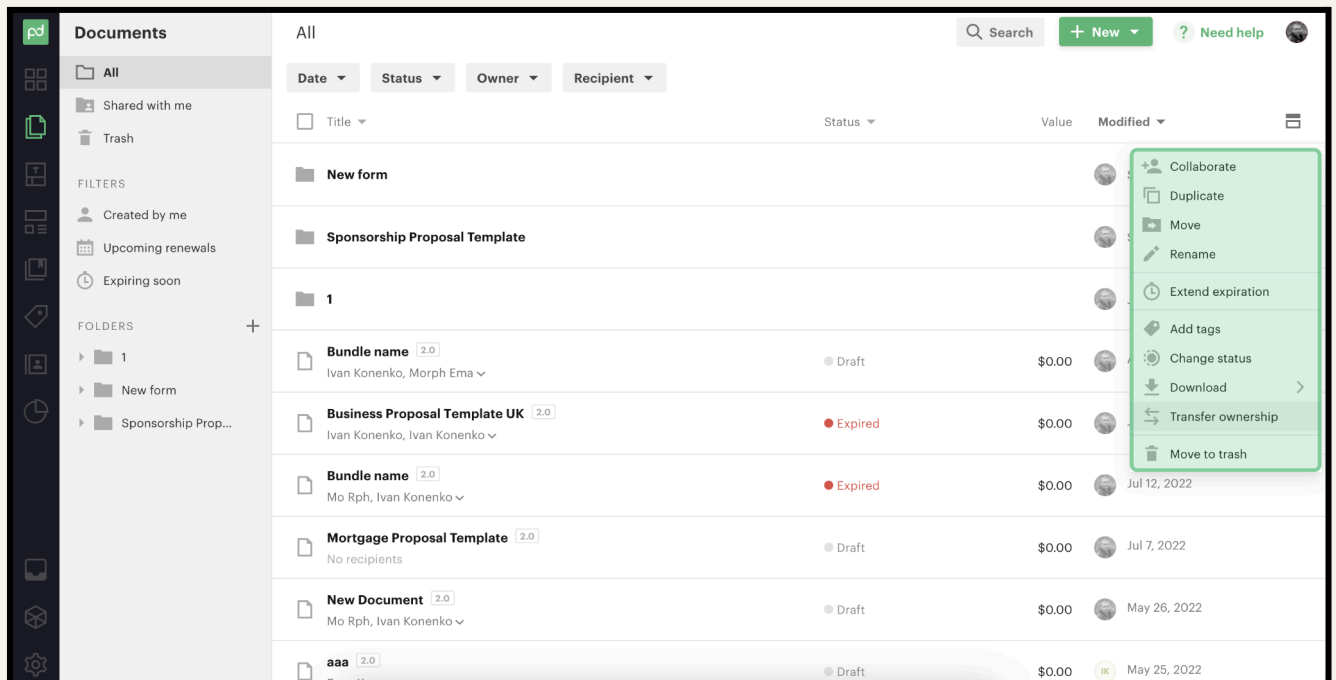
App bar

1. The contextual menu contains inconsistent deprioritized options.



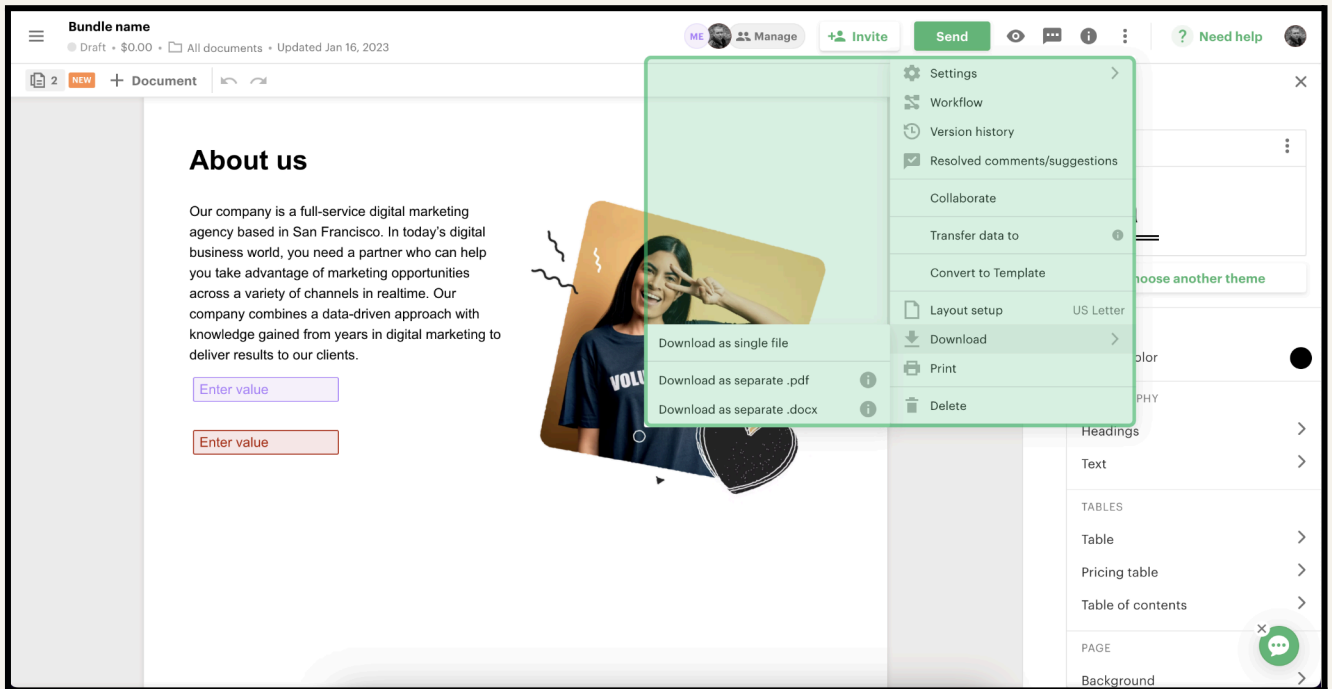
Several times I've heard team members call it the graveyard of actions due to the massive amount of scattered not-so-popular-but-still-kind-important options.

2. This context menu isn't aligned with the 3-dots menu from the lister. Having the same menu items for entities with different views is vital.

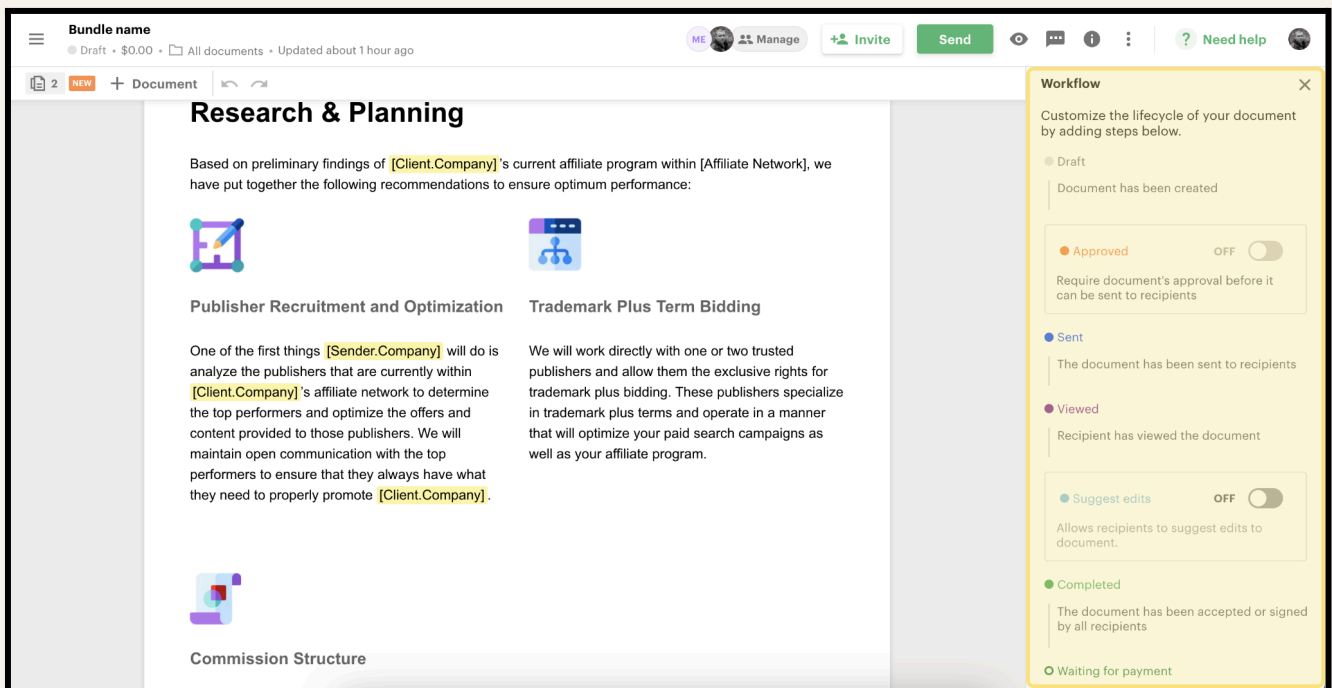


For example, we can transfer ownership, change status, or manage tags in Lister, but for some reason, not on the detailed document view.

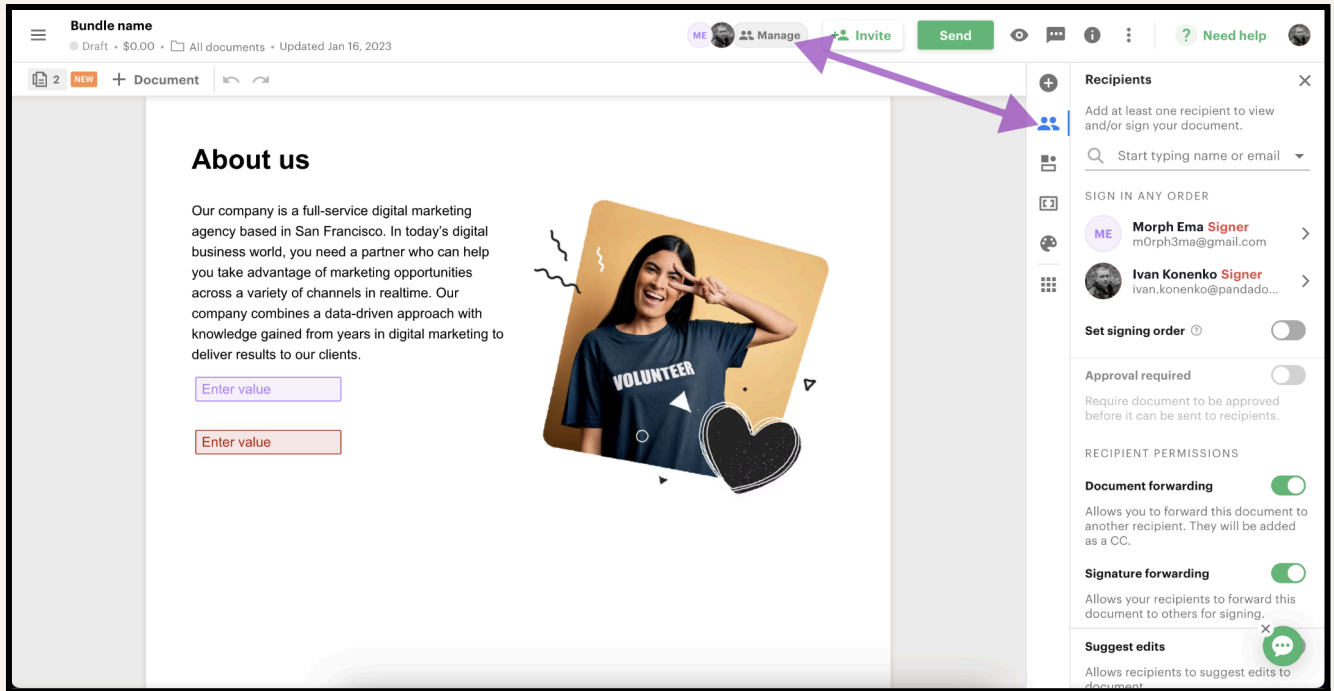
3. Such a trivial thing as document downloading is hidden in the third [!] level of the context. It looks like an attempt to hide the ability to download documents due to the poor formatting of exported files. Rarely used Workflow settings are on the same level.



TTL number, grouped by types (.docx, .pdf)



4. We have different-looking controls in the header and the right toolbar that do the same.

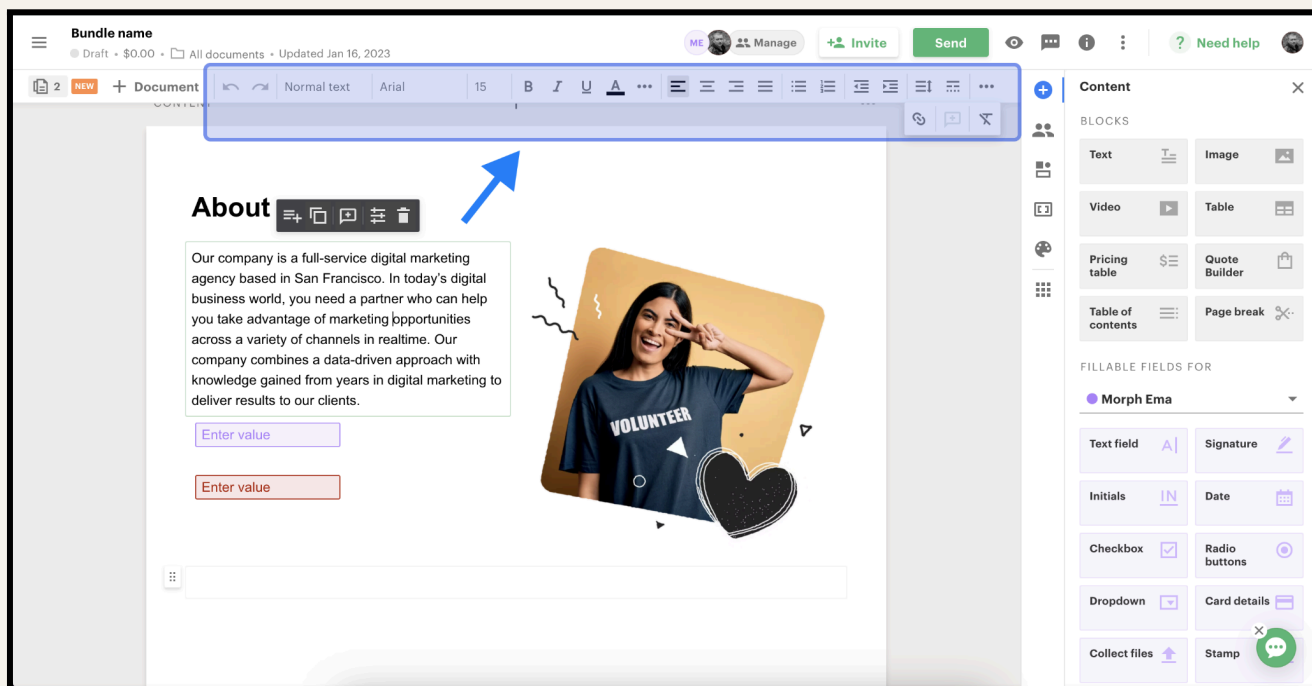


There are several cognitive issues we observed during user interviews:

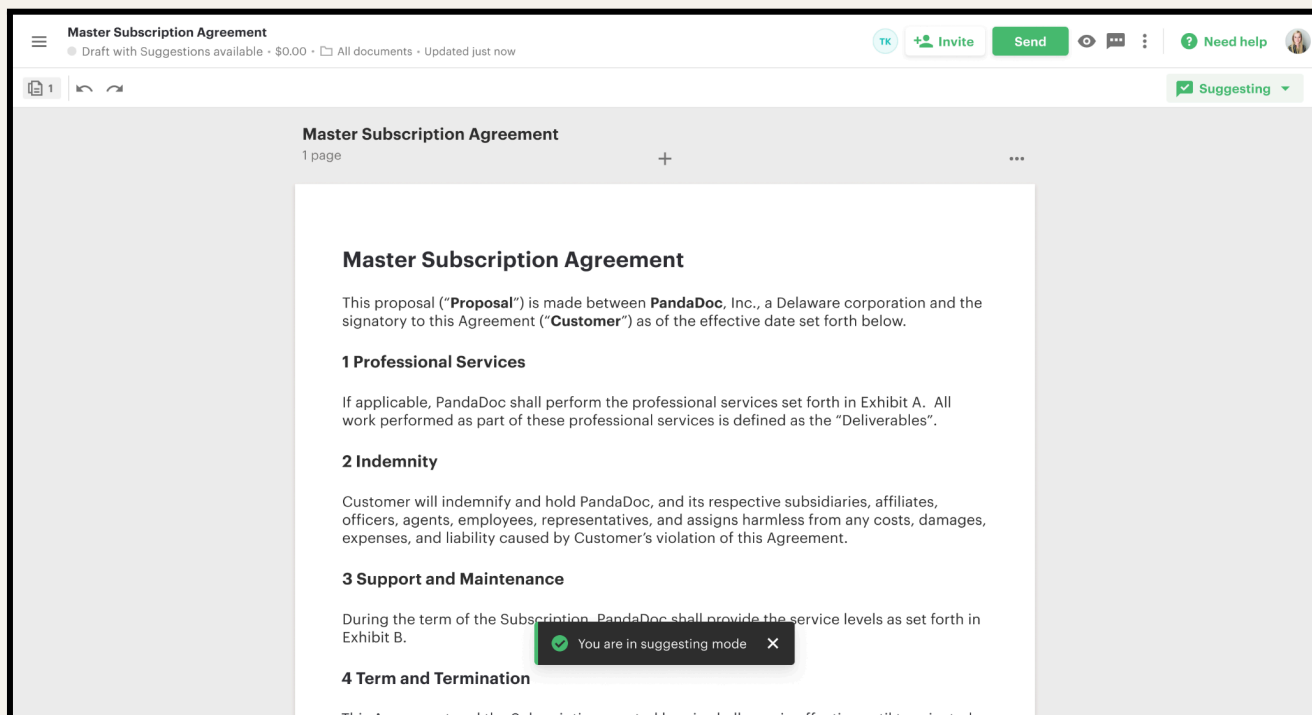
- It's difficult to distinguish the entity type we open (Document, Template, CLI).
- The right side of the header is overwhelmed with controls "Manage", "Invite", "Share", and icons.
- Avatars representing document actors are often misinterpreted as the "being in the document right now" pattern.

Toolbar

1. The expensive real estate of the top toolbar works exclusively for the properties of text blocks. Any other element of PandaDoc doesn't use this panel.

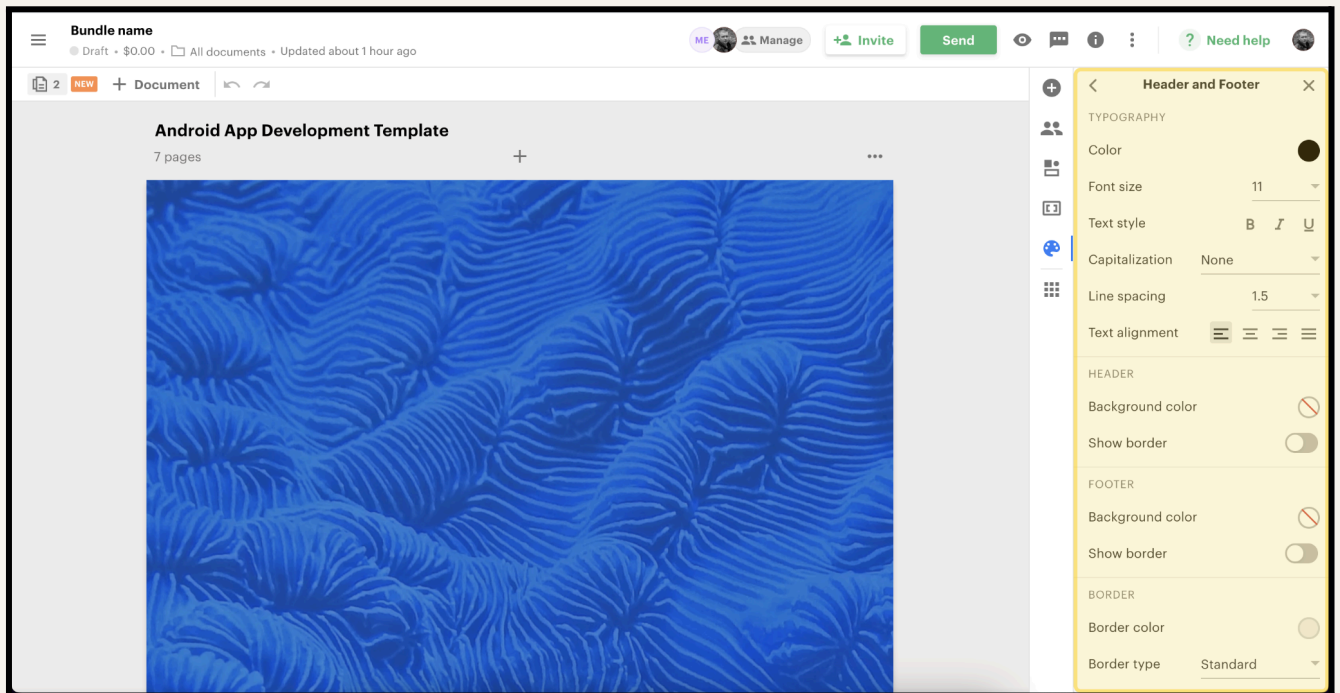


The problem becomes more prominent considering the upcoming mode selector.

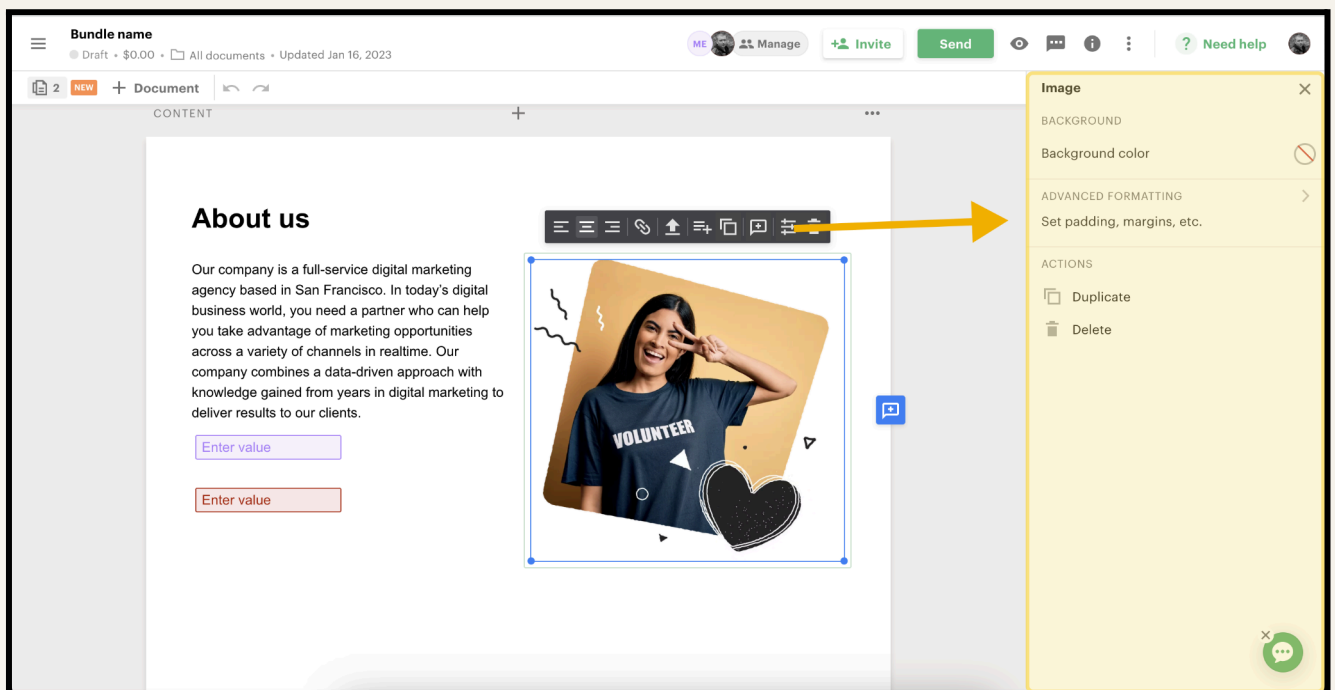


Content area

1. Floating toolbar items and detailed property lists are inconsistent for different content types. This is a huge and deep topic, but it's the same issue on a high level.

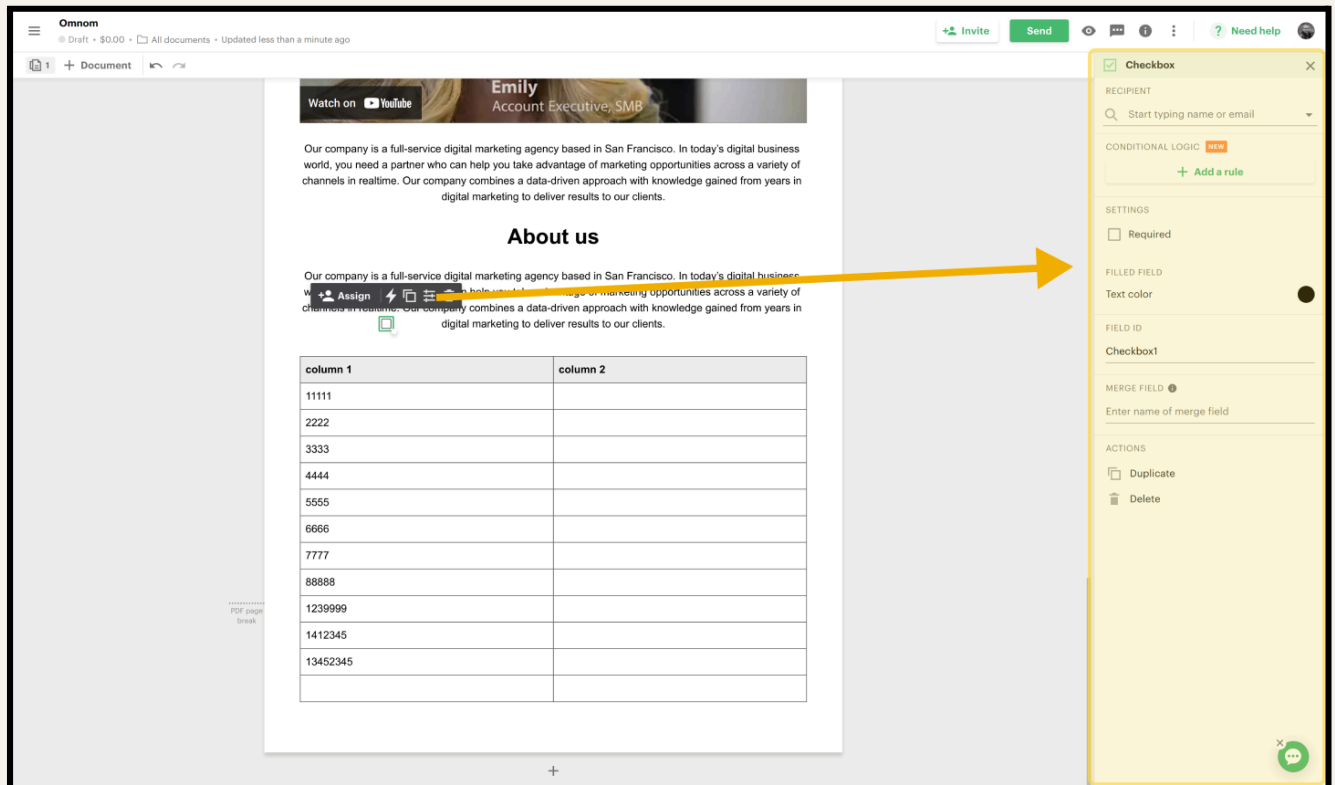


2. We don't show any properties in the floating panel using the top toolbar for the **text block**. "Detailed" properties contain several additional settings and duplicated actions. We don't need to open additional settings to customize this block type.



At the same time, full textual properties in the right panel are available only for the general document styles.

We show most of the properties in the floating panel for the **picture** block. "Detailed" properties in the right panel contain several additional settings and duplicated actions.



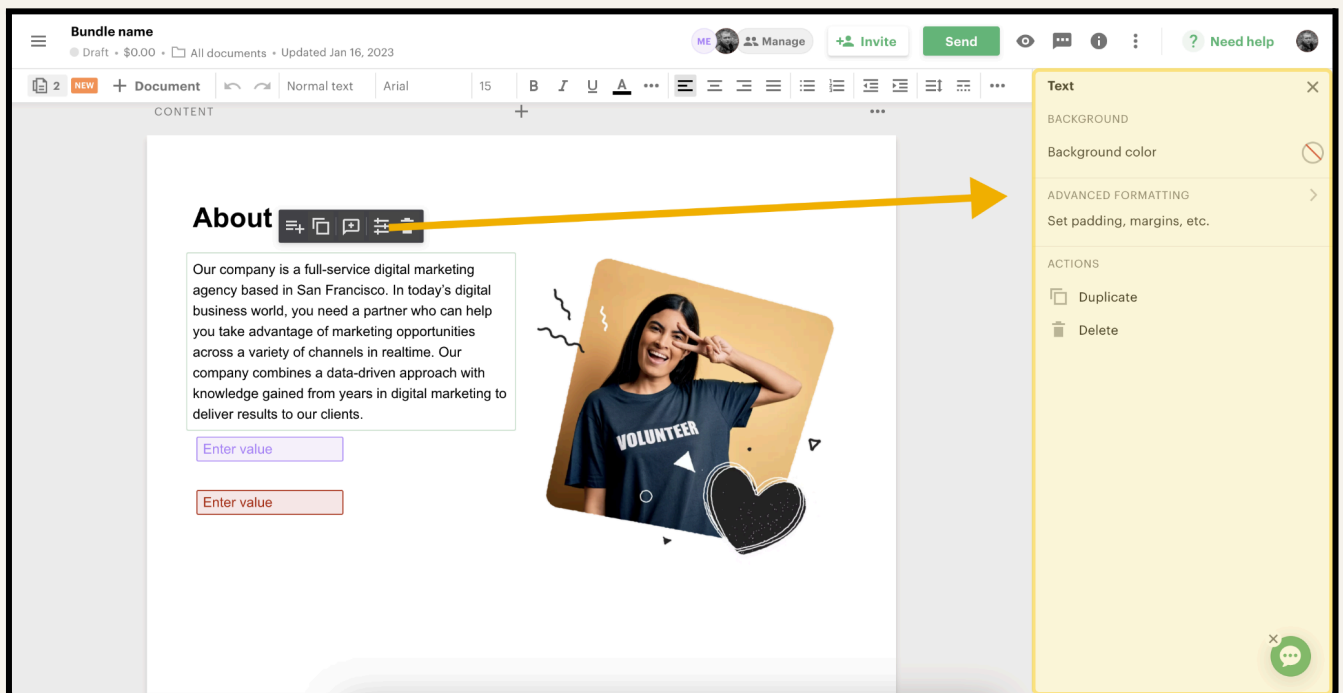
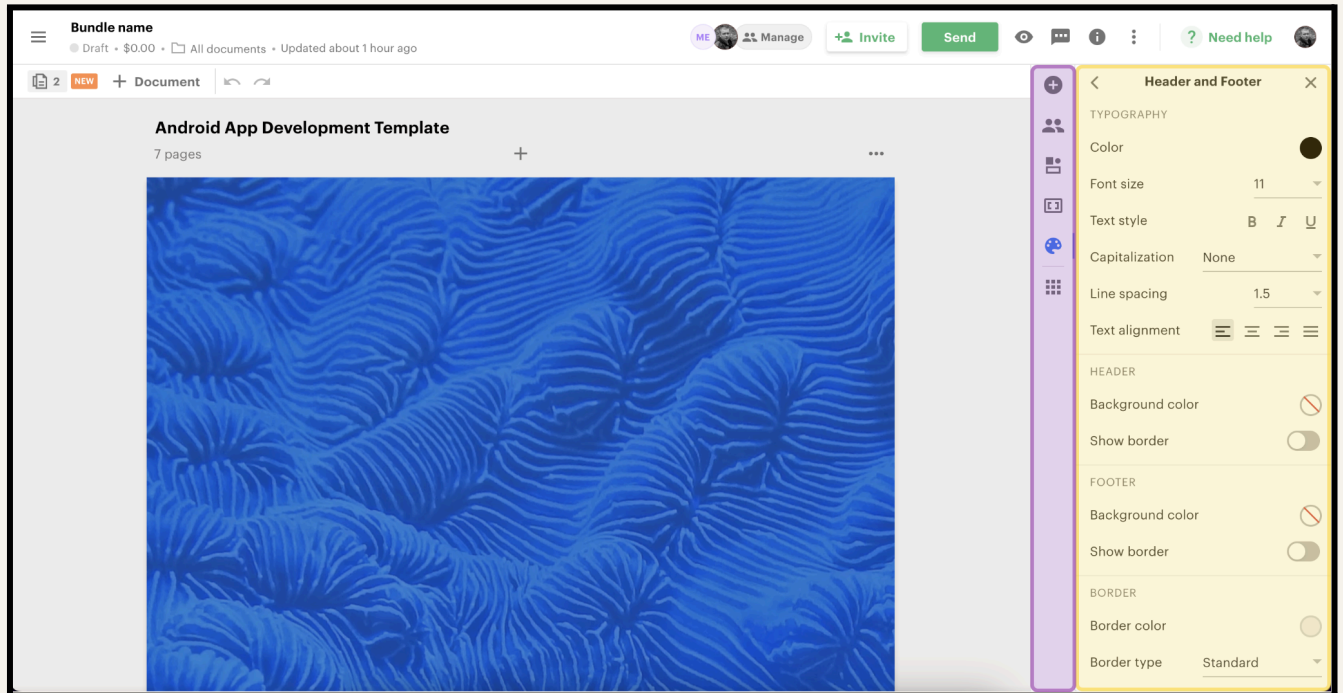
The **checkbox** contains the main action in the toolbar and all the settings in the right panel. The same issue is with the **Dropdown**.

There are no internal rules for how we use the floating panel, top toolbar and right panel for different block types.

3. The right panel has one more problem with large screens. For example, its state changes, but the user's eye does not notice it. It doesn't look critical, but it definitely came up many times in the research.

Right panel

1. The right panel and property drawer are entirely different things. It looks like Internal navigation within these components is critical. Unfortunately, users are constantly confused about recognising the difference.



General approach

- We demand that users remember where specific actions are placed.
- Users never know where to find block properties because it depends on the block's type.
- We don't have a consistent set of main actions available for documents in different views (list and detailed views)
- Some document and recipient settings are duplicated for no proper reason.

Solutions

- Eliminate unnecessary duplication of settings by consolidating them and providing a clear explanation of the relationship between the duplicated settings.
 - Rearrange the top toolbar and context menu from the header.
 - Unify context menu items with the list view options set.
 - Unify toolbar behaviour for different block types.
 - Unify the right panel behaviour for different block types.
- Develop a consistent set of main actions that are always available, regardless of the view, to help users quickly perform common tasks.
- Provide a consistent location for block properties regardless of block type, or use a contextual menu to allow users to access properties directly from the block.
- Move the most rarely used items to the settings tab of the DocInfo panel.