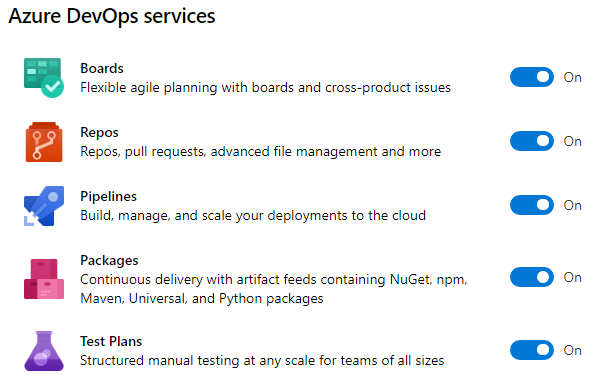
# Chapter 4: Tracking Work

Now that you’ve looked at the capabilities of the professional DevOps environment and a mix of tools that can be a part of it, we’ll drill down into each product within the Azure DevOps family and set it up in the proper way. You’ll certainly want to customize the configuration, but your suggested configuration works great in 80% of the cases. If you’ve already read the book “The Phoenix Project” by Kim, Spafford, and Behr, you’ll recognize the principles we implement in this chapter. You might want to create a new project so that you can test different configurations as you read. Once you have your Azure DevOps project created, take a glance at your project settings and select the products that you’d like enabled.

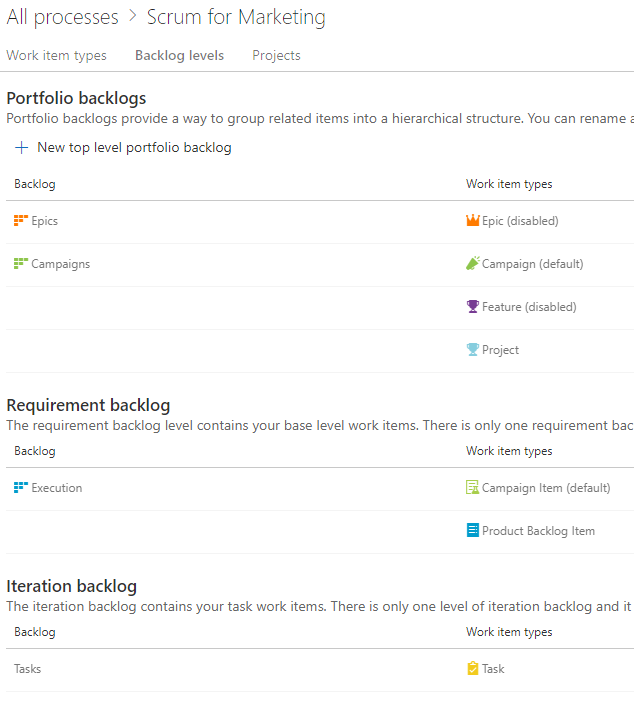
Below, you can see that I have all of the products enabled. For your team, you’ll want to equip them with the Visual Studio Enterprise subscription (formerly called MSDN Premium) so that they have licensing for all of the products. You’ll need them. Packages is the first one you’ll miss if you are using a free or lower license. And as we move through the book, you’ll make use of all the products in the Azure DevOps family.



*Figure: You can enable or disable any of the products in the Azure DevOps family.*

## Change your process template

The title of this section is not “choose your process template.” You will do that, but your organization has a workflow, and you must capture it and make the process template faithfully model that workflow. Within the first way of DevOps is the principle of “make work visible”. Azure Boards is the tool of choice for modeling the shape of your work. Azure Boards uses Work Items to track a unit of work. A work item can be of any type and has a status as well as any number of other fields you’d like. As you think about your hierarchy of work, don’t immediately start creating work items using the built-in sample hierarchy. Instead, think about the work that you already do and the parent-child relationships between some of the types of work. For example, in a marketing department, the structure in the figure below may be appropriate.



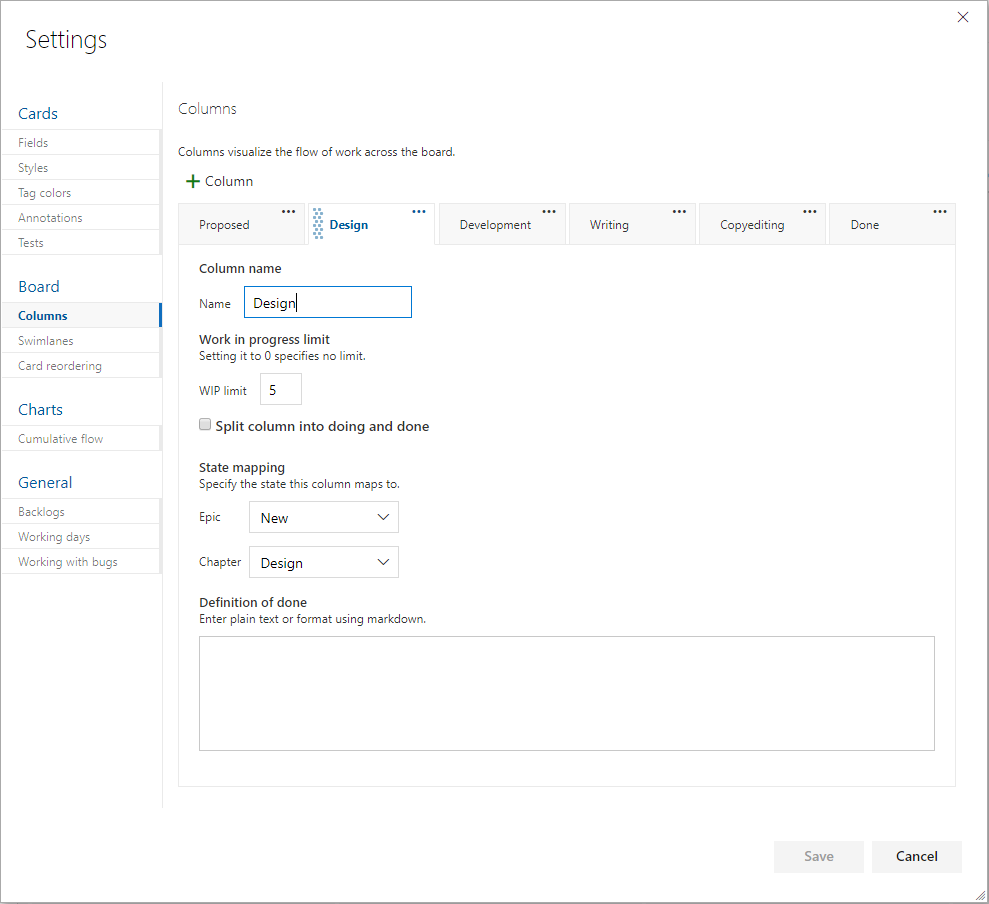
*Figure: A marketing department has Campaigns that are broken down into individual items.*

This marketing department has decided that they only need three levels of work. A Campaign can have multiple Campaign Items or Product Backlog Items. A Campaign Item and a Product Backlog Item can have multiple Tasks. At the top level, they can track at the Campaigns level or the Execution level. An individual iteration or sprint is tracked with Tasks. You can have any number of higher-level portfolio backlogs if you need higher levels of groupings. Even while the built-in process template includes **Epic > Feature > Product Backlog Item**, you’ll quickly outgrow this because it won’t match your organization. You need to disable most of the built-in work item types and create your own so that you can name them and put only the fields and the progression of statuses that make sense in your teams’ environments.

You may think of the following work types to get the creative juices flowing in order to capture the model of your organization’s world. Notice that I didn’t say “design the model.” Your model already exists. You need to capture the nouns and the verbs of your existing reality and make Azure Boards represent what’s already there. If you capture the wrong model, it won’t fit, and your co-workers will have a hard time tracking their work because it just won’t make sense. So, consider the following types:

* Business initiatives
* Marketable features
* Plannable work to budget, schedule, and fund
* Individual tasks to get done

This becomes the foundation of your usage of Azure Boards going forward. You’d never think of starting a new application with the Northwind or AdventureWorks database schema. Those tables were chosen by someone else. That model just doesn’t fit the nature of the data you’re trying to store. In this same way, the schema of the built-in process templates won’t fit your organization. You need to load your own model. Once you have your model, you need to specify the process of each major entity (work item). For example, if you were writing an article or a book, you might create a Chapter work item and specify the status progression on the Kanban board like that shown below.



*Figure: The columns all map to a state of a work item, and each can be assigned a definition of Done.*

By determining ahead of time what the process is to take a certain level of work item from creation to done, you organize your team. Each state, or board lane, should be owned by a type of role. For example, if you have a stakeholder designated as the person who’ll give the go ahead on the sketch of a screen before it’s developed, that stakeholder should have a column where they own the work within it. Each work item is represented by a visual card on the Kanban board, and the cards in their column are theirs to work. If the stakeholder does nothing, cards pile up in that column, and nothing is developed because of the bottleneck in that column. A dashboard report can bring this to light on a daily basis so that no column has too much work in it. The stakeholder’s job would be to either approve the sketch of the screen or initiate a conversation to fix it. In no case would you want a bad screen to be coded. That would be worse. By creating a good number of columns, mapped to the states of the work item, you can move the work through a known process where every column has a type of role responsible for performing a known set of work and then forwarding the work in process WIP to the next column. From a quality control perspective, every person starting on work has the obligation for inspecting the WIP to see if the work is ready for them yet. If something is missing, you stop the line and get it corrected before propagating the error further downstream.

For the purposes of software teams, the level of backlog that is prepopulated with Product Backlog Items in the case of the Scrum process template or User Stories in the case of the Agile process template, is the appropriate level for doing branches and pull requests as well as designed test cases, as you’ll see a bit later in the article. Iterations or sprints can be planned with work items from this level. Then, tasks can be organically created, completed or destroyed day by day. It’s often good to make plans based on the lowest backlog level and then break those down into tasks as needed on an ad-hoc basis during the sprint.

## Types of work items

Depending on the process template you choose when you create your project, you’ll start with a pre-defined set of work item types, statuses and swimlanes in your boards. You should change these because there are only three process templates built-in, and they are all very basic. Don’t expect to use them without customization except for very simple projects. You have three process templates to choose from when starting a project. If you have already created a project, and you want to choose a different process template, you are out of luck. Create a new project. If that ship sailed long ago, don’t fret. You can morph any of the project templates into just about anything you want.

The choices for project template are CMMI, Agile, and Scrum. The scrum template is probably the most widely used, and it is the template that is maintained the most. If you don’t know the difference between these, **choose the Scrum template**[[1]](#footnote-1) and modify it from there.

You will see some similarities and differences in the built-in process templates, but they all share more than they differ. The following table illustrates the configurations of the templates.

|  |  |  |  |
| --- | --- | --- | --- |
| Backlogs | CMMI | Agile | Scrum |
| Portfolio | Epic Feature | Epic Feature | Epic Feature |
| Requirements | Requirement | User Story | Product Backlog Item |
| Iteration | Task | Task | Task |
| Others | Bug Issue Change Request Review Risk | Bug Issue | Bug Impediment |

*Table: Built-in process templates come with a set of work item types that are meant to be customized*

You can see how similar the process templates are, and you should examine each one to gain some ideas because each work item is configured with a certain number of fields, and the fields of each is likely not going to fit your needs. As with a database schema if you go forward with tables and columns that are not used, your data set ends up with many null values. This causes confusion with reporting. If you are not going to use a field, customize your template and remove it or hide it from a work item. Simple is better.

You may think that the above processes are so similar that it doesn’t matter which one you start with, but the Requirements level work item type will probably help you make your decision. Here are the fields in this key work item type out of the box.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CMMI | Agile | Scrum |
| Name | Requirement | User Story | Product Backlog Item |
| Main Section | Description (multi-line text) | Description (multi-line text) | Description (multi-line text) |
| Secondary Section | Impact Assessment (multi-line text) | Acceptance Criteria (multi-line text) | Acceptance Criteria (multi-line text) |
| Development Section | Development (links) | Development (links) | Development (links) |
| Related Work Section | Related Work (links) | Related Work (links) | Related Work (links) |
| Planning Section | Planning | Planning | Details |
| Classification Section | Classification | Classification | n/a |
| Other | Effort   * Size * Priority * Triage * Blocked * Committed * Story Points * Priority * Risk * Priority * Effort * Business Value * Value area * Type * Value area * Value area * Original Estimate   Schedule   * Start Date * Finish Date   Build and test   * Integrated In * User Acceptance Test   Subject Matter Experts   * Subject matter expert 1 * Subject matter expert 2 * Subject matter expert 3 | n/a | n/a |

*Table: Structure of the main work item type per process template*

As you can see, the process templates start to diverge at this point. You can hide fields of the built-in work item types, but you can’t remove them. It’s a cleaner work tracking data model to add custom fields rather than hide most of the built-in fields.

## Customizing your process

1. Pit of success: start new process templates inheriting from Scrum [↑](#footnote-ref-1)