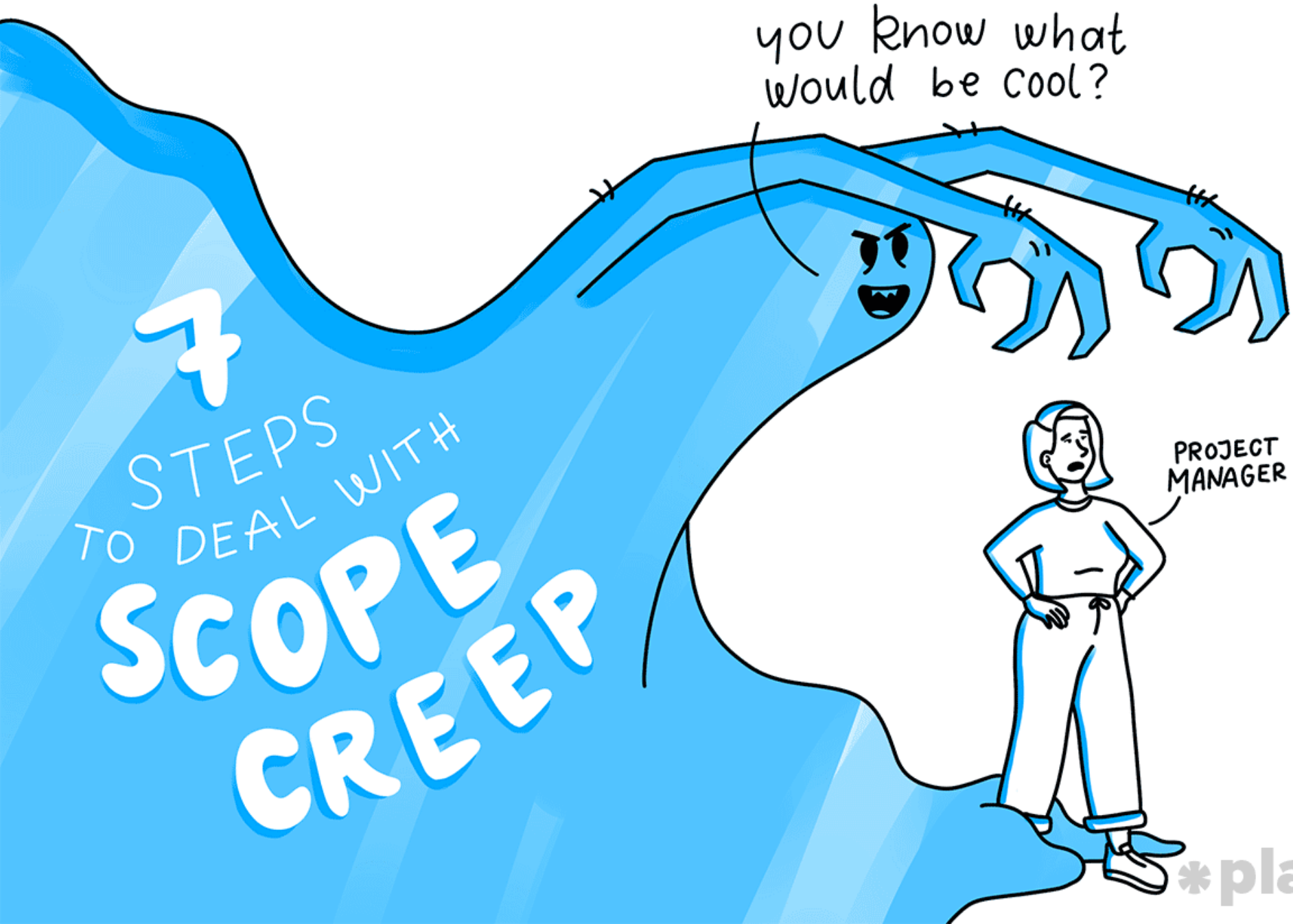
แก้ไขครั้งล่าสุดเมื่อวันที่ 5 พฤษภาคม 2568

2301361

SYSTEMS ANALYSIS AND DESIGN

11

Moving into Implementation



https://plan.io/blog/scope-creep/

How does scope creep happen in project management?

7 ways to avoid scope creep and keep your project on track

› 1. Know your project goals from the start

› 2. Get serious about documenting requirements

› 3. Use project management software to keep everyone on track

› 4. Create a change control process

› 5. Set (and stick to) a clear schedule

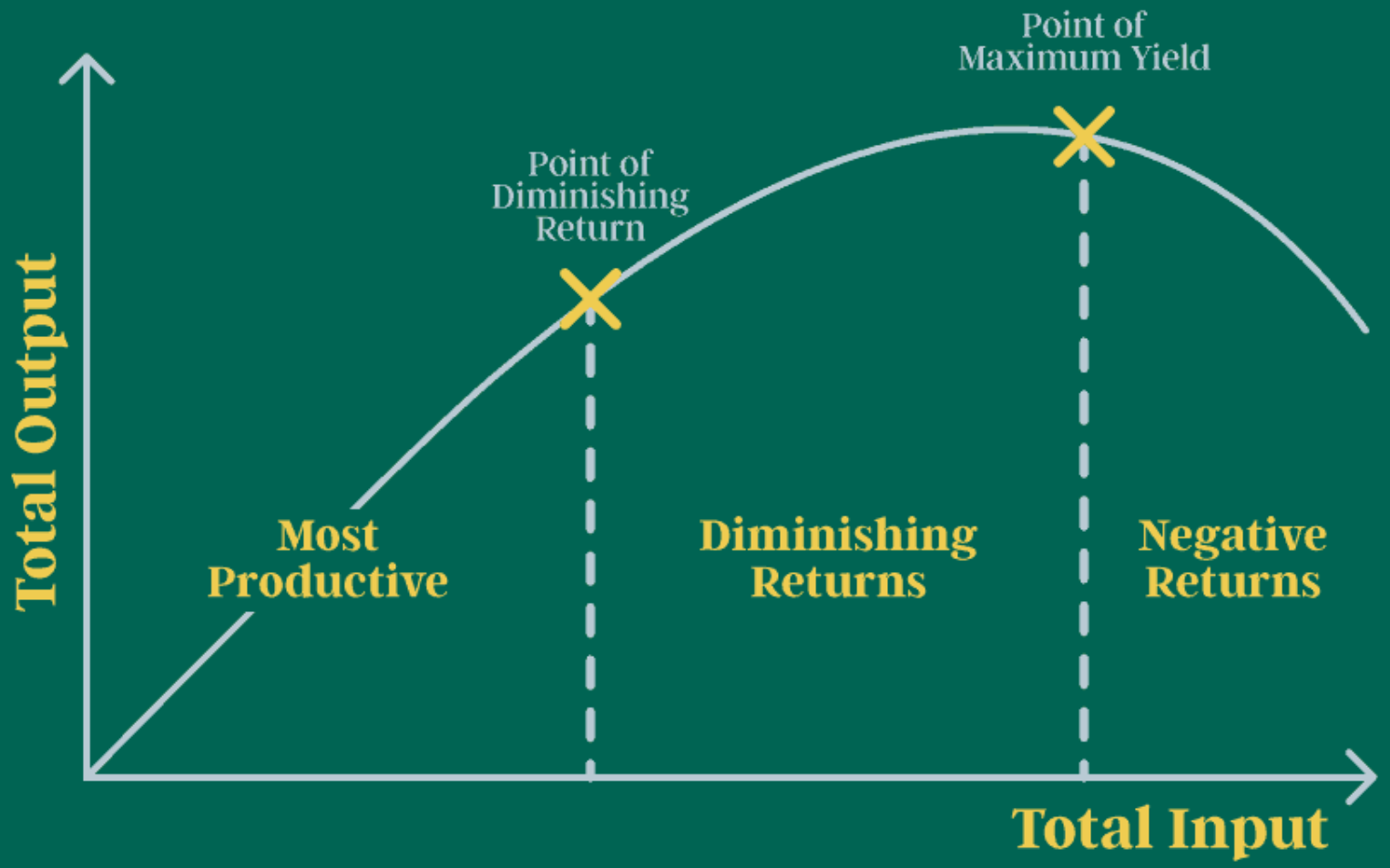
› 6. Learn the proper ways to communicate with stakeholders and your team

› 7. Protect your team against “Gold plating”

If you give in to every stakeholder demand,
you won't be facing scope creep.
You'll be facing a full-on scope eruption.

To take control of your project's scope, you're going to have to learn to say no sometimes. Even to your boss, manager, or an important project stakeholder. Saying no to people in power is never easy. But it's the best way to protect the quality of a project, and that's exactly how you should think about it.

What is Diminishing Returns



LifeHack

PRACTICAL TIP 11-1 Avoiding Classic Implementation Mistakes

In previous chapters, we discussed classic mistakes and how to avoid them. Here, we summarize four classic mistakes in the implementation phase:

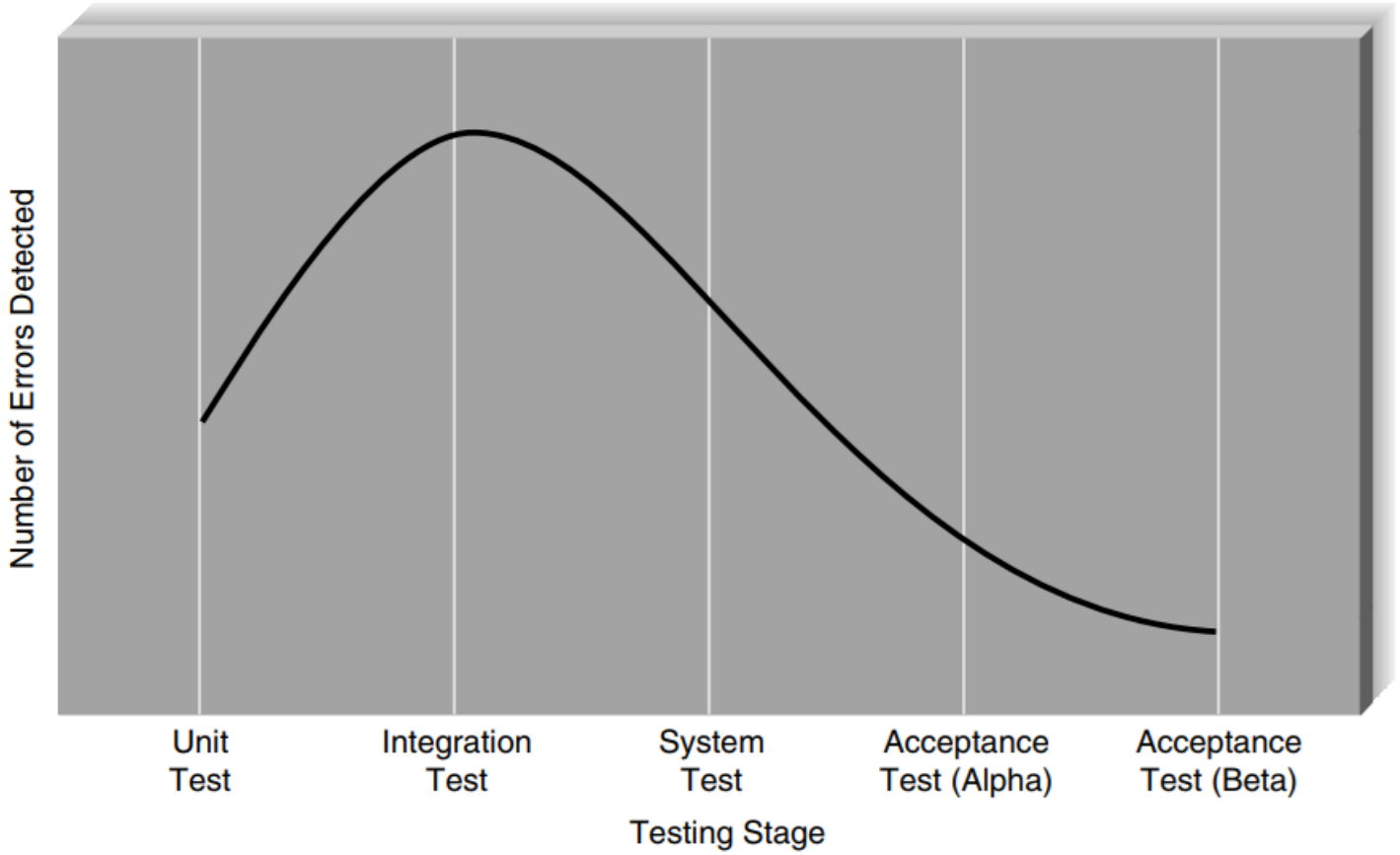
1. **Research-oriented development:** Using state-of-the-art technology requires research-oriented development that explores the new technology, because "bleeding-edge" tools and techniques are not well understood, are not well documented, and do not function exactly as promised.
   Solution: If you use state-of-the-art technology, you should significantly increase the project's time and cost estimates even if (some experts would say *especially if*) such technologies claim to reduce time and effort.
2. **Using low-cost personnel:** You get what you pay for. The lowest-cost consultant or staff member is significantly less productive than the best staff. Several studies have shown that the best programmers produce software six to eight times faster than the least productive (yet cost only 50–100% more).
   Solution: If cost is a critical issue, assign the best, most expensive personnel; never assign entry-level personnel to save costs.
3. **Lack of code control:** On large projects, programmers must coordinate changes to the program source code (so that two programmers don't try to change the same program simultaneously and one doesn't overwrite the other's changes). Although manual procedures appear to work (e.g., sending e-mail notes to others when you work on a program to tell them not to work on that program), mistakes are inevitable.
   Solution: Use a source code library that requires programmers to check out programs and prohibits others from working on them at the same time.
4. **Inadequate testing:** The number-one reason for project failure during implementation is ad hoc testing—in which programmers and analysts test the system without formal test plans.
   Solution: Always allocate sufficient time in the project plan for formal testing.

Adapted from: *Rapid Development*, Redmond, WA: Microsoft Press, 1996, pp. 29–50, by Steve McConnell.

Test Planning

- Unit Tests
- Integration Tests
- System Tests
- Acceptance Test (Alpha)
- Acceptance Test (Beta)

user interface, use scenario, data flow, system interface testing



ศัพท์เทคนิคที่ควรทราบ stubs, hardcoded

Stage	Types of Tests	Test Plan Source	When to Use	Notes
Unit Testing	**Black-box testing:** treats program as black box.	Program specifications	For normal unit testing	The tester focuses on whether the unit meets the requirements stated in the program specifications.
	White-box testing: looks inside the program to test its major elements.	Program source code	When complexity is high	By looking inside the unit to review the code itself, the tester may discover errors or assumptions not immediately obvious to someone treating the unit as a black box.

Integration Testing	**User interface testing:** The tester tests each interface function.	Interface design	For normal integration testing	Testing is done by moving through each and every menu item in the interface either in a top-down or bottom-up manner.
	Use scenario testing: The tester tests each use scenario.	Use scenario	When the user interface is important	Testing is done by moving through each use scenario to ensure that it works correctly. Use scenario testing is usually combined with user interface testing because it does not test all interfaces.
	Data flow testing: Tests each process in a step-by-step fashion.	Physical DFDs	When the system performs data processing	The entire system begins as a set of stubs. Each unit is added in turn, and the results of the unit are compared with the correct result from the test data; when a unit passes, the next unit is added and the test is rerun.
	System interface testing: tests the exchange of data with other systems.	Physical DFDs	When the system exchanges data	Because data transfers between systems are often automated and not monitored directly by the users, it is critical to design tests to ensure that they are being done correctly.

System Testing	**Requirements testing**: tests whether original business requirements are met.	System design, unit tests, and integration tests	For normal system testing	This test ensures that changes made because of integration testing did not create new errors. Testers often pretend to be uninformed users and perform improper actions to ensure that the system is immune to invalid actions (e.g., adding blank records).
	Usability testing: tests how convenient the system is to use.	Interface design and use scenarios	When user interface is important	This test is often done by analysts with experience in how users think and in good interface design. This test sometimes uses the formal usability testing procedures discussed in Chapter 8.
	Security testing: tests disaster recovery and unauthorized access.	Infrastructure design	When the system is important	Security testing is a complex task, usually done by an infrastructure analyst assigned to the project. In extreme cases, a professional firm may be hired.
	Performance testing: examines the ability to perform under high loads.	System proposal and infrastruc-ture design	When the system is important	High volumes of transactions are generated and given to the system. This test is often done using special-purpose testing software.
	Documentation testing: tests the accuracy of the documentation.	Help system, procedures, tutorials	For normal system testing	Analysts spot-check or check every item on every page in all documentation to ensure that the documentation items and examples work properly.

Acceptance Testing	**Alpha testing:** conducted by users to ensure that they accept the system.	System tests	For normal acceptance testing	Alpha tests often repeat previous tests but are conducted by users themselves to ensure that they accept the system.
	Beta testing: uses real data, not test data.	No plan	When the system is important	Users closely monitor the system for errors or useful improvements

Developing Documentation

แบ่งประเภทเอกสารตามผู้ใช้

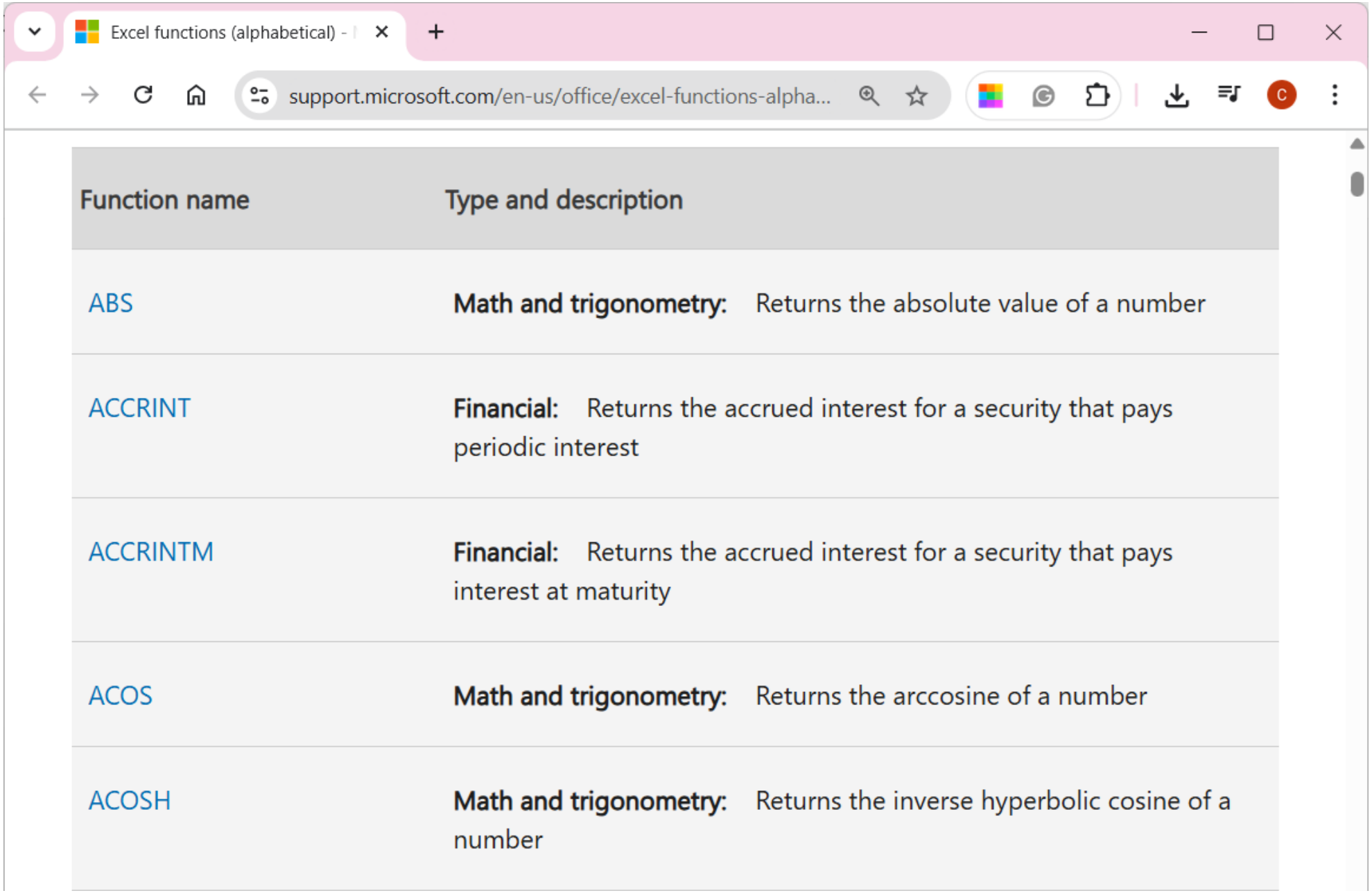
- System Documentation
- User documentation

แบ่งประเภทเอกสารตามเนื้อหา

- Reference documents
- Procedures manuals
- Tutorials

3 hours per page (single-spaced)

Reference Documents



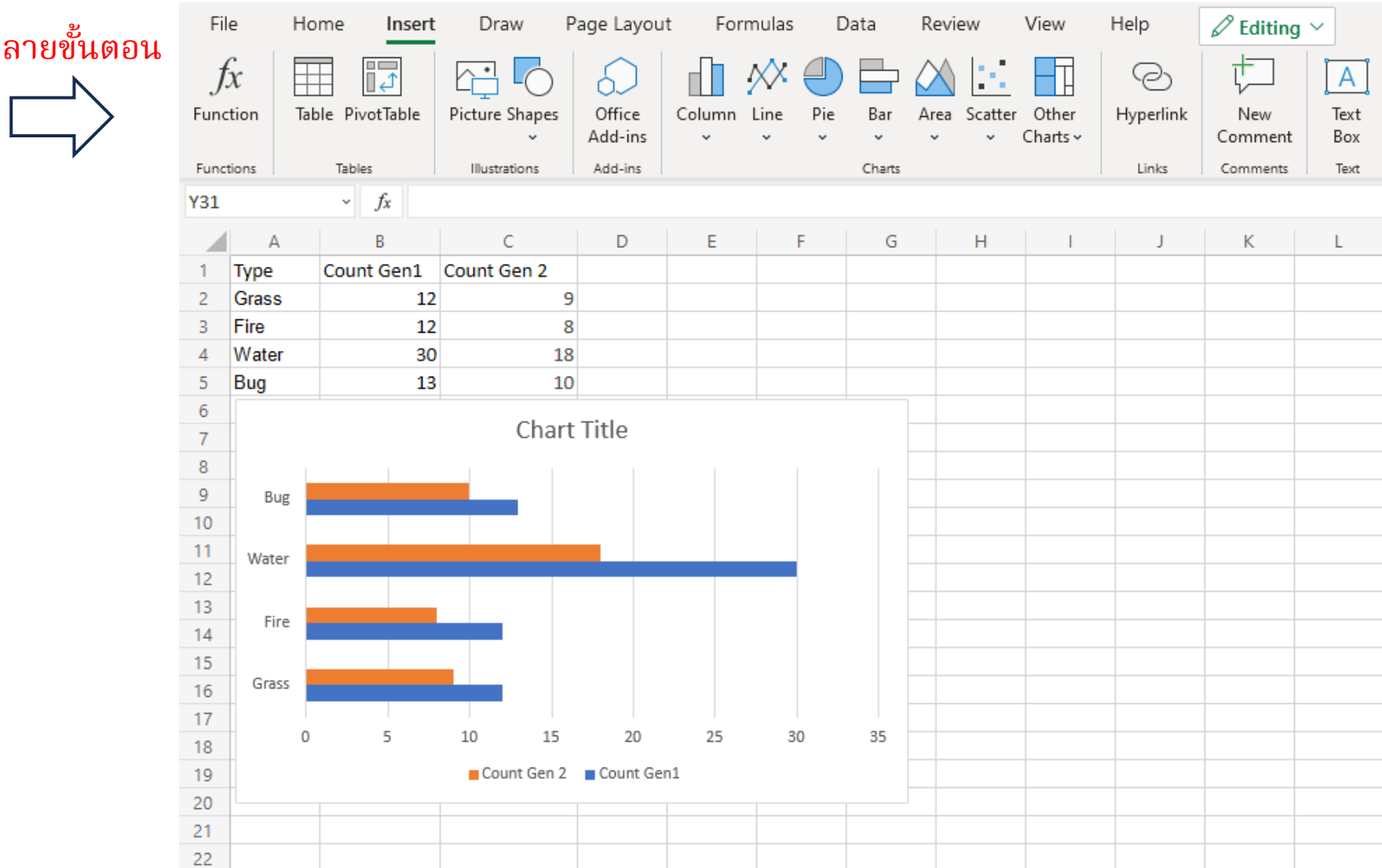
Function name	Type and description
ABS	**Math and trigonometry:** Returns the absolute value of a number
ACCRINT	**Financial:** Returns the accrued interest for a security that pays periodic interest
ACCRINTM	**Financial:** Returns the accrued interest for a security that pays interest at maturity
ACOS	**Math and trigonometry:** Returns the arccosine of a number
ACOSH	**Math and trigonometry:** Returns the inverse hyperbolic cosine of a number

Procedure Manuals

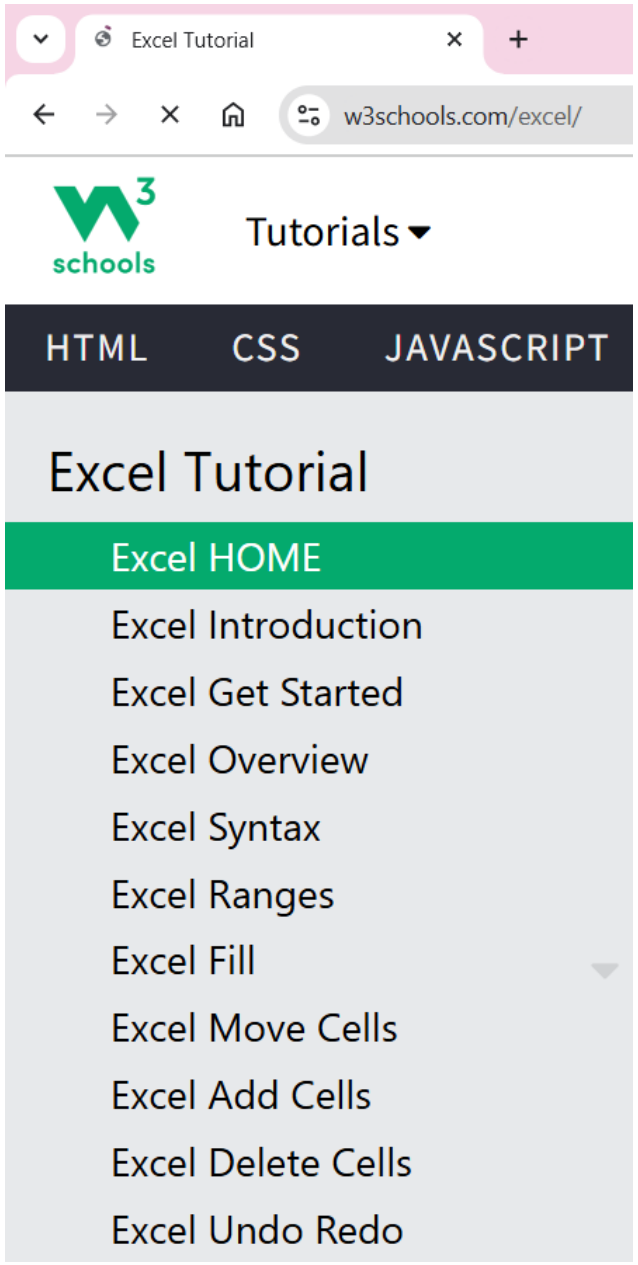
สร้าง clustered bar chart

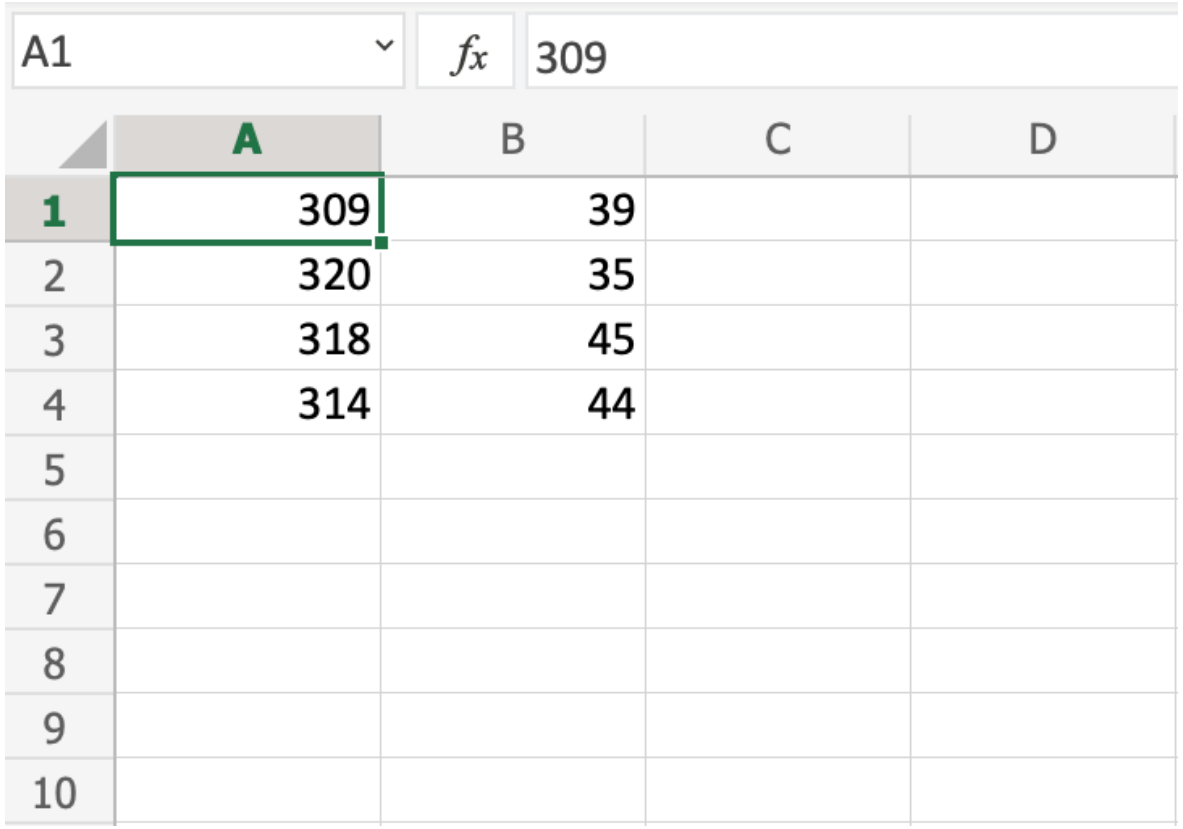
	A	B	C	D
1	Type	Count Gen1	Count Gen 2	
2	Grass	12	9	
3	Fire	12	8	
4	Water	30	18	
5	Bug	13	10	
6				

มีหลายขั้นตอน



Tutorials


Excel Tutorial
w3schools.com/excel/
Tutorials
HTML
CSS
JAVASCRIPT
Excel Tutorial
Excel HOME
Excel Introduction
Excel Get Started
Excel Overview
Excel Syntax
Excel Ranges
Excel Fill
Excel Move Cells
Excel Add Cells
Excel Delete Cells
Excel Undo Redo


A1
fx
309
A
B
C
D
1 309 39
2 320 35
3 318 45
4 314 44
5
6
7
8
9
10