I have undertaken the following using both Agile and Waterfall methodologies

**Software development processes**

**Identifying requirements**

Worked with end users and product owners to identify what the existing system (manually based / IT based) does and does not do.  Identify what these two sets of people would like the system to do - both essential and optional requirements. Then, away from the meeting(s) with end users and product owners, analyse the existing system for yourself both from an end user perspective as well as (from a programmers point of view) the existing code base should one exist.  Attempt to identify how easily the essential / optional requirements can be incorporated into the existing system.  Create business spec

Examples where I've done this:

rewriting a database containing student placement data

creating an IT based timesheet system at Capita from the ground up

working with end users and product owners at Xpress to develop new functionality as well as maintaining existing functionality

**Working from requirements (in business or technical specs)**

Read and comprehend business and technical specs describing (in the case of the former) business requirements from an end user/product owner point of view.  Note anything that you need further information about and speak to the writer of the spec, either by email, phone or in person. Analyse the functionality in any existing software and identify how you are going to incorporate the required functionality. From this, write the technical spec.

In the case of working from a technical spec, read and comprehend how the Senior Programmer / Technical Architect required functionality to be implemented.  As per the technical spec, note anything that you need further information about and speak to the writer of the spec, either by email, phone or in person.

Examples where I've done this:

enhancing / maintaining functionality in products such as Capita Appointments

/ SmartStream Reconciliations (SSR) or Transaction Lifecycle Management (TLM) / Capita Contractor.

**Code**

Having read the tech spec, analyse the existing functionality and identify how you are going to incorporate the required functionality - hopefully what you think should be done in the way of technical implementation should be the same as outlined by the writer of the tech spec.  If you have any queries / suggestions for improvement then ask - interaction between the coder and the author of the tech spec should be encouraged regardless of whether you are in an Agile / Waterfall environment

Examples where I've done this:

SmartStream Reconciliations (SSR) or Transaction Lifecycle Management (TLM)

**Test/Document**

Unit test the functionality you have coded - ensure that the new functionality works and also that existing functionality is not impaired by your changes.  Thoroughly test the aforementioned functionality especially at places where the user is asked to make a choice or where the user enters data for some validation to take place.  Write documentation from the end users point of view, not from a technical perspective.

Examples where I've done this:

Throughout my IT career

**Maintaining large systems**

Developed / maintained what I thought were large systems at SmartStream, only to be dwarfed by the software products I latterly encountered, namely Capita Housing, PARIS (at NewGen Innovation) and the Register and Management products at Xpress Software.  These were software products that contained a huge amount of functionality and it was necessary when, for example, amending a function called by mainly different processes to ensure that no existing functionality was impacted by your changes.  When working with such products I made an effort to concentrate my efforts on familiarising myself with one major part of functionality rather than employing a scattergun approach looking at several large areas functionality at once.

**Working with developers**

Worked with developers who were Junior to me, Senior to me as well as on the same level.  Always endeavour to help people where possible if they need information and not be afraid to ask if needed to find out something from someone with more knowledge of the technologies used or the product itself.  Do this for other people in different teams such as QA.  Endeavour to find out answers for yourself using the likes of Google before you ask a question of someone.  I have typically worked in individual teams of around six people and there could be as many as five/six different teams within a company.