

UMass Boston CS410
Homework 1
Posted Tuesday, 8 September 2026
Due Monday, 14 September 2026

J. Holly DeBlois

1 General Information

Homework must be typed and converted to Portable Document Format (PDF), see <https://en.wikipedia.org/wiki/PDF>. We use the Linux servers of the Computer Science Department to collect homework. Homework submitted by email will be sent back. To submit your homework, prepare one PDF file called hw1.pdf — the filename must be exactly hw1.pdf, otherwise it will not be collected. Upload the file to the cs410 folder linked to your home directory on the CS Linux server. If you have trouble with uploading, email operator@cs.umb.edu for immediate help and copy the instructor at jane.deblois@umb.edu. The questions in this homework are based on the agile principles at <https://www.agilemanifesto.org> and the reading in Essential Scrum by Kenneth S. Rubin: Figure 1.3 on p7 and Chapters 2-4.

2 Looking at Agile Principles in the Agile Manifesto, 2001 and Figure 1.3 - Cynefin framework, Essential Scrum p7, which is from Snowden and Boone, 2007

Please explore the Agile Manifesto website first and then look at Figure 1.3. Keep in mind that many failures of software projects motivated the individuals who contributed to the Manifesto. Simply put, large software projects are very difficult to build.

a. List the four items that explain what the signers have come to value. Each is a two-sided preference statement where the left side is preferred to the right side. You may paraphrase but be sure there are two parts to each of the four items.

b. Then add at least one question that you have regarding what is meant by one of the preference statements. Note: All eight left and right sides are part of the software work we do.

c. Cynefin describes development environments as Complex, Complicated, Simple, Chaotic, Disorder and Interrupt-Driven Work. Note: These environments can co-exist or morph into each other. Rubin suggests that when folks sense that the environment is Disorder, they tend to show a personal preference for the phase-based, sequential approach that works well in Simple environments. Instead, Rubin suggests you try to break the situation into parts that you can assign to one of the other domains so you get out of Disorder. Please write one sentence that characterizes the kinds of development environments that you have worked in in the past.

3 Looking at Chapter 2 - Scrum Framework

Please read Chapter 2 to become familiar with the terminology we will be using. Note that Rubin states the "Scrum framework is like the foundation and walls of a building." (p13) Your task will be to "customize inside the structure of Scrum, adding features and fixtures until you have a process that works for you." (p13)

a. Read the roles - Product Owner, ScrumMaster and Developer. In cs410, everyone is a Developer and we take turns being ProductOwner or ScrumMaster. On early projects our timeframe is 1-2 classes. For the long project, each sprint is about a week. Write a statement that explains whether you prefer the ProductOwner role or the ScrumMaster role and why.

b. Suppose we have a software project that has 5 parts; we list these features on the Product Backlog as what the client wants. We figure out one part that has the most business value and make the goal of the first sprint building code for that. Now look at Figure 2.8 Sprint Planning (p21). The last step is drawing the Sprint backlog which is a spreadsheet of tasks for each feature. We have one feature. Figure 2.9 Sprint backlog (p22) shows an example of breaking features into (sub)tasks with effort estimates. Figure 2.10 Sprint Execution (p23) shows the work being done by the team. What is missing is how to decide who does what!

That's what you have to get good at figuring out quickly since the sprint planning is a short part of the sprint. Please write a paragraph about what questions (give at least 5) that you and your team might ask and discuss to get this figured out.

c. You would actually need a spreadsheet that shows not only each subtask of the feature being built, but also how these subtasks are further broken out to be assigned to each developer individually or in pairs, etc. Select an imaginary project with minimum three features and three subtasks for each. Select the one to be built in the first sprint. Create a drawing of an imagined spreadsheet assigning parts of each subtask to each of three people on the team. All are developers, one is also ProductOwner and one is also ScrumMaster. Include your drawing in your pdf.

Figure 2.12 Sprint results (potentially shippable product increment) (p25) shows the output increment of the product that gets built in this sprint. It is called "Potentially shippable product increment" because it pleases the client and is software that runs.

4 Looking at Chapter 3 - Agile Principles

Consider the information in Table 3.4 (p59). Check that you understand the contrast in columns 2 and 3 for each of the topics. Then select the three topics for which the agile principle surprises you most. For each of the three, write a sentence explaining what surprised you and where you think the principle might be useful. Be sure to identify each one you choose by including the topic and the agile principle sentence in your sentence.

5 Looking at Chapter 4 - Sprints

Please reread the Definition of Done information in Table 4.1 - Example Definition-of-Done Checklist (p74). There are 17 checkboxes for accomplishing the 7 factors listed. Please write your own definition of done checklist for your individual project: building your own website on the cs server. Use the same 7 categories. Put as many checkboxes as you think you need.

Note 1: In the project, you are to use the testing to drive your design, so you will produce what is called test-driven code. The tests will not need to be automated, just written in simple sentences. You can define the tests on the checklist by a name or by actually writing what to type for the test.

Note 2: Keep in mind that your write-up of design, end-user documentation and test plans and reports will need to use as few words as possible! It is the running product and its code that need most attention. Your website can be very simple, but doneness will have to be accomplished fully and carefully using the checklist you write here.