**Discussion Topics: Pragmatic Programming**

In this module's discussion board assignment, answer the following questions:

1. Using the ***Pragmatic Programmer*** reading assignment, select one (1) topic and complete the following:
   * Why did you select this topic?
   * Summarize the main points (in your own words) of that topic in three or four sentences.
   * Find at least one additional resource (video, book, article, website, etc.) that supports your summary. Write an additional 1-2 sentences. Include a link to that resource.

After reading topics 11-15, I decided to focus on topic 11: reversibility. I chose this topic because sometimes it can be easy to fall into the mindset that there is exclusively one single way to accomplish a task. Often, I am intrigued when I watch others perform tasks, even of a more mundane nature, because you can learn a new way of doing something, and sometimes more efficiently.

The *Pragmatic Programmer* touches on reversibility as a developer. It is essential to look at things differently. We must take a step back and realize we can take more than one path. There is rarely a single solution to a task, and if you think there is only one, you are likely not thinking big enough. As developers, we may decide on one database or vendor only to realize we need to adapt our plans because our original choices no longer work out for one reason or another (Thomas & Hunt, 2024/2020). We can change our course by remembering that at the start and during our progress, we may need to change how we are going about things. This helps when we have to make a change because there was planning. Being prepared for change makes the change less overwhelming. Understanding that things are subject to change helps prepare developers for the reality of coding and an ever-changing world (Thomas & Hunt, 2024/2020). Flexibility should surpass developing code and factor in other attributes leading to project adjustments (Thomas & Hunt, 2024/2020).

Ziaur Rahman also touches on reversibility in his 2019 article. Changes may occur throughout a project, but the primary way to increase reversibility, making it a more straightforward process, is by documenting every step along the way (Rahman, 2019). This helps to avoid going through the same steps again if changes need to be made and makes it easier to redo or undo a step.

**References**

Rahman, Z. (2019, August 22). *Reversibility — the missing key Engineering Principle of Software Development*. Medium. https://medium.com/@m.ziaur.rahman/reversibility-the-missing-key-engineering-principle-of-software-development-5e5ed49681c5

Thomas, D., & Hunt, A. (2020). *The Pragmatic Programmer: your journey to mastery*. Addison-Wesley. (Original work published 2024)

***Before you submit your thread, put your name in the subject line.***

**Assignment Requirements and Grading:**

1. An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CT**.
2. For the initial post to be considered substantive, it should be at least 250 words in length and fully cover the topics being presented. Single-sentence definitions or responses will not be awarded points.
3. Submit your post by clicking on the **Assignment Link** above, then **Create Thread**. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
4. A minimum of three (3) responses, **to the original threads of other students**, of 100-200 words each are due by **Sunday, 11:59 p.m., CT**.
5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf).

**(50 points)**

Arely, you did a nice job on your post this week! It was interesting reading how you plan to implement this in the future and how it stuck out to you since it already has elements you enjoy doing! You are correct that prototyping saves companies money. I could see people worried that it is a waste of time, but having a battle plan before executing a project can be essential to its success. Sometimes, a good prototype is all you need to win over a client! Developing a program can be time-consuming, especially if it does not meet stakeholder expectations. Taking the time to create a prototype can save time if the client is unhappy from the start.

Adrian, I really liked reading more about your personal experience and how it related to your chosen topic! I cannot agree with you enough about the importance of prototyping. Diving right into a project can lead to many mistakes, causing frustration for the developer and potentially other vital stakeholders. It can help to counter some of the issues that might occur when first releasing the project for testing since obstacles are carefully thought out. Of course, even with a prototype, errors, and issues that need to be addressed are likely to be encountered, but having a plan can make this less overwhelming.

John, you did a fantastic job on your post! Working in customer service and for a roofing company, I often see how customers can get upset if a project takes longer than estimated. This is the same for customers expecting scheduling calls and communication. When you are open about how long it will take or even overestimate, it helps to avoid upset customers. Even for my coursework, resources, and work tasks, I often estimate how long everything will take me. Similar to what we read, overestimating is the best choice to help avoid easing expectations. Just like with customers, it yields more positive results when you can accomplish a task faster than you estimated.