**Discussion Topics: TDD, BDD, Andon Cord**

Select one of the following and find at least two articles on the topic. Address the subtopics for your selection.

1. TDD
   * What is TDD?
   * What are the advantages and disadvantages of TDD?
   * Why was TDD created?
2. BDD
   * What is BDD?
   * How is it used? Why is it used?
   * Why do some developers prefer BDD over TDD?
3. Andon Cord
   * What is and Andon Cord? Where is the term come from?
   * When would you pull an Andron Cord?
   * When would it NOT be a good idea to pull an Andon Cord?

For my post, I will discuss the Andon Cord and the articles I found about it. The first article I found was on the Medium website from 2020, written by Farida Aliyeva. The second article I found was by Peter Waterhouse on the DevOps website.

The origin of the Andon Cord comes from the founder of the Toyota Systems Corporation, Sakichi Toyoda (Aliyeva, 2020). Sakichi Toyoda is credited with the Japanese industrial revolution (Aliyeva, 2020). It is a Jidoka quality method where a system is stopped when a problem occurs and is immediately worked on to resolve it (Aliyeva, 2020). The point of stopping the process when an error is found is that it will not go unresolved or trickle down into the rest of the process (Aliyeva, 2020). Anyone can and should pull the Andon Cord if a problem is discovered so that production stops to tackle the problem (Aliyeva, 2020). When employees are encouraged and feel positive reinforcement for reporting a problem, they are more likely to pull the cord (Aliyeva, 2020). Aliyeva suggests that the Andon Cord can be used in the software development process through automated testing so the team can see errors along the way (Aliyeva, 2020). Any failed tests or errors in code should be resolved before moving forward (Aliyeva, 2020).

The Andon Cord should be thoughtful since the rest of production halts when pulled (Waterhouse, 2016). Having a cord to pull fosters a better environment for all and makes mistakes not feel like something to sweep under the rug (Waterhouse, 2016). The Andon Cord in software development can also help against defects and security issues (Waterhouse, 2016). Having a customer service cord can improve their experience and the system's overall performance (Waterhouse, 2016).

So, the best time to pull an Andron Cord is when you witness an obvious problem. It would not be a good time to pull an Andon Cord if there is no problem. Production should not stop just because; make sure the reason you are pulling the plug is legitimate.

**References**

Aliyeva, F. (2020, November 22). *Toyota’s Andon Cord and its implementation in Software Development.* Medium. https://farida-alv04.medium.com/toyotas-andon-cord-and-its-implementation-in-software-development-a84674201dbc

Waterhouse, P. (2016, November 7). *You’re Not Doing DevOps if You Can’t Pull the Cord*. DevOps.com. https://devops.com/youre-not-devops-cant-pull-cord/

***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., CST**.
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., CST**.
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)**

Hey, Joe! I really enjoyed reading your post for the module this week. You were thorough with the definition, the advantages and disadvantages, and why it was created. I would agree that TDD is a seemingly weird approach. Usually, it makes the most sense to aim towards error-free code to begin with, rather than creating something set up to fail. TDD also improves collaboration efforts with reviews by developers and those performing the tests. Doing all these tests means the code developed is much cleaner than without them. Requirements are highlighted in unit tests, so it helps meet the mark.

Hi, Brian! I think you did a fantastic job on your post for this week. I also chose to write about the Andon cord, so I enjoyed hearing your thoughts. I think it is intriguing how an idea for the Toyota Systems Corporation can seep into DevOps. As an employee, there have been times when I have messed up and initially felt hesitant to admit my mistake. So, having a set principle that mistakes should be reported makes it easier for employees to come forward, so I think having an Andon Cord should be implemented in most, if not all, production.

Hey, Megan! I think your discussion post for this module was very well said. You perfectly touched on all the aspects of the Andon Cord. All your suggestions for times when an Andon Cord should be pulled make sense. In DevOps, it could also be pulled when code is not running at all, when it is running but there is a logic error, when the code is obsolete, or when only parts of it are functioning. Depending on the severity of the problem, halted production can be the best route to take before the mistake continues to travel into further steps.