**Discussion Question: Bug Tracking**

At the moment George Everett, the senior developer, is working on bug tracking. He understands the importance of documenting code bugs and/or security bugs, but is not sure how that would actually work. Your job this week is to provide him with an option for documenting the bugs. At the least, they'd need a date of discovery, a description, a level of severity assigned, and who was assigned to fix the bug, if the level of severity warrants it.

Consider how you would like to have the data available if you were on his team. Is there software available for this task? Would a whiteboard make more sense? A spreadsheet? Something else? In your recommendation, provide reasons why this would be the optimal choice. You also need to provide a graphic example with test data to support your recommendation.

In your responses to other students, provide an opinion as to the effectiveness of their recommendation.

When developing and managing software, the documentation of code bugs, security bugs, or any other errors is vital to fighting against vulnerabilities. As George Everett is aware, bugs are run into when developing code. When bugs are found, an attacker can sometimes exploit them, which is why they must be remedied while less severe ones are less of a priority (Conklin & Shoemaker, 2022).

To start on the basics, all bug documenting should include the date of discovery, a description, a level of severity assigned, and who was assigned to fix the bug, if the level of severity warrants it.

If I were a part of the team, I would prefer using a designated software. If that were not an option, then using a spreadsheet would be my second suggestion. There is software available for bug tracking. Some options are ACCELQ Manual, Jira, Bugzilla, Mantis, Redmine, Zoho Bug Tracker, ClickUp, GitHub, Trello, and HP ALM/QC (N, 2025). Based on the tools presented, the ACCELQ Manual seems like the best choice if I choose a bug tracking software. It can link defects for failed test runs, allows for dynamic testing, includes detailed reports, and tracks runs across environments (N, 2025). It is also the only tool that does not have any negatives listed.

One option that may seem like a route to go in would be using a whiteboard. This is a simple way to track bugs and the basic information that should always be documented, as mentioned previously. But this seems like the worst solution since it can not be accessed remotely and is less official.

Another option is a spreadsheet. All the required bug tracking information can be included in a spreadsheet like Google Docs so others can see live updates to it. However, a spreadsheet is not as intuitive as a tool designed for bug tracking.

Here is a graphic that breaks down the top 10 bug tracking tools and further elaborates on why I think ACCELQ Manual is the best option.

A white table with text on it

AI-generated content may be incorrect.

(N, 2025)

**References**

Conklin, WM. A., & Shoemaker, D. P. (2022). *CSSLP Certified Secure Software Lifecycle Professional: Exam Guide.* McGraw-Hill Education.

N, C. M. (2025, March 24). *Top 10 Bug Tracking Tools In 2025*. ACCELQ. https://www.accelq.com/blog/bug-tracking-tools/

***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)**

Hey, Arely. I think you did a great job on your post for this week! Your solution to George Everett of using a shared spreadsheet seems like a good route. The visual example you created is a great way of showing how the spreadsheet could look. However, I do think that a bug tracking tool might be even better than a shared spreadsheet. It is not as easy to catch little errors in a spreadsheet compared to a tool dedicated to helping with this process. Of course, the issue of costs comes with it, but I think it is worth implementing.

Hi, Colton! I thought your post for this week was very insightful. I agree that tracking bugs is essential to maintaining quality and transparency. Tools are a great way to help with bug tracking, and it is also what I decided on for my post. I think GitHub issues is a great repository to utilize for bug tracking! We have used and learned much about GitHub throughout our courses, so using it here also looks pretty beneficial. I like how you included a screenshot of the test issue you created using GitHub. It demonstrates how easy it can be utilized in real life.

Hello, Samir! I think you did an excellent job on your post for this module. Implementing bug tracking software is the best way to handle and manage bugs, especially compared to a basic spreadsheet or whiteboard. Software development teams will not always be located in the same area. Also, if someone cannot get into a physical office, they would not be able to see the content needed on the whiteboard or update it in real time, like software or a spreadsheet can. I like how you created your graphical assignment preview in your post. It really shows your understanding of the subject.