8-1 Discussion: Implementation and Maintenance

I think for larger companies it would be best to use different teams for programming and testing their software. This allows the different teams to have a more focused approach to their specific tasks and allows there to be separation to avoid over-encumbering the team members. They should, however, work closely with each other and provide feedback where needed or offer suggestions if they feel something is lacking in either process. Having fresh eyes never hurts the process.

I work at a company that has newly integrated an automation system for filling orders for stores and shipping them to the right place. One issue we run into frequently is the system getting bogged down, it starts to run very slowly interfering with how the system responds to the users. This requires the company to contact “tier 2”, which is basically a supporting group of people to look at the system and figure out why we are experiencing these issues. This usually requires us to schedule a CR and Bounce to erase unnecessary data and clear up data storage. This could be improved by having a more frequently scheduled Bounce and CR to avoid these problems in the first place.

For the DriverPass system, I would suggest adding a “Report a Problem” link to the system. This will allow the user to enter their contact information as well as give a full description of what issues they may be experiencing. A confirmation notification should be sent to the user letting them know that the request has been received and is being transferred to the correct personnel. This request will be sent to the IT members, and they can work to correct the issues while maintaining communication with the user, ensuring that issues are being resolved and what timeframe they can expect the issue to be fixed.

A good team member in system development is someone who can communicate properly and explain what is needed. A good team member works well with others and takes responsibility for their part in the development process. They share their experiences and use already proven, well-working code in their development (not reinventing the wheel). They work together to fix discrepancies in a timely manner. Good documentation is vital in ensuring that the developers and other designers on the team can implement the designs. Having clear and concise documentation helps guide the developers in what the system is expected to do and how to meet the client’s needs and requirements. Skipping a requirement or leaving out a base class can be costly to the development of the system.

This sequence diagram demonstrates the scheduling process for the DriverPass system. It shows that the system will check the availability of a car and instructor for the desired date and time for the student. If both are available, the appointment will be scheduled and posted on the dashboard for the user to see. If one or the other is not available the customer will be given the opportunity to try a different date and/or time.