**Q1. Why does it take so long to get software finished?**  
**A1.** Software development involves several complex stages—requirements gathering, design, implementation, testing, and deployment. Requirements may change mid-way, bugs may arise, or integration with other systems may take longer than expected. This unpredictability, combined with the creative and iterative nature of software design, leads to delays.

**Q2. Why are development costs so high?**  
**A2.** High development costs stem from the need for skilled professionals, time spent on testing and debugging, iterative changes, and tools or infrastructure. Additionally, more time is now spent on security, user experience, and regulatory compliance, which increases overall cost.

**Q3. Why can't we find all errors before we give the software to our customers?**  
**A3.** It’s nearly impossible to test every scenario, especially in large, complex systems. Users may interact with the software in unexpected ways or use it in environments developers couldn’t simulate. Limitations in testing resources and time also mean some issues go undetected until real-world use.

**Q4. Why do we spend so much time and effort maintaining existing programs?**  
**A4.** Maintenance includes fixing bugs, adding new features, adapting to new platforms, and improving performance. As technology and user expectations evolve, existing systems must be updated to stay relevant, secure, and functional. Over time, the cost of maintaining aging systems often exceeds initial development.

**Q5. Why do we continue to have difficulty in measuring progress, as software is being developed and maintained?**  
**A5.** Software progress is hard to quantify because development is not always linear. Unexpected bugs, requirement changes, and integration challenges can slow or reverse progress. Unlike physical products, the “visibility” of progress in software is abstract, making it harder to track.