REFLECTION

Today I will be discussing with you: Silver Bullet while using Cherry Picking and Scientific Method, Google Copy Repo and Make Se More Ethical as information to back up certain things or give examples . “Silver Bullet” by Frank Brookes discusses the monster known as Software engineering. When I was growing up my sibling use to babysit me while my parents were at work. We would always watch anime because my brother went to Japan with my father for several years and that where he was introduced to anime, loved it and wanted to introduce it to us. I remember the first anime I ever watch was アンパンマン and ドラゴンボール. The plot was the same, the evil villain reck havoc on the city, planet, and hero friends etc. And everyone would wait for the hero to save the day to make thing easier for everyone and to end everyone suffering. One watches horror movies like Blade, Underworld etc. Where vampire reach havoc on the world and as you are getting bit by a vampire you just think within your head that you wish you had a silver bullet to kill this vampire and make thing easier for you and end the suffering. The point I am trying to make, and the point Brooke made is that there is no silver bullet in software engineering. We as developer might think the solution is nontechnical and straightforward but software can become a disaster like a tsunami or as frightening as a vampire or werewolf. From: miscommunication, to complex structure, flawed product, unaffordable budget etc. I think we as developer or as programmer wished we had a silver bullet like hardware has and reduce productivity and production with reduction of cost. As you read “Silver Bullet”, Brooke categories software difficulties into two groups: essential difficulties which discusses the difficulties that was inherent by the software itself which are problem that unavoidable due to the complex nature of the software. Then accidental difficulties which is the result of one overcomplicating things for themselves and making things harder. Which I think relates to the “Cherry Picking and Scientific Method” article. Cherry Picking is a very misleading and untrustworthy method which relate to making thing more difficult for yourself. Cherry Picking does not provide all the test cases for the experiment and it hard to find pattern in data so I would categories this as accidental difficulties. Then Brook goes on discussing the source of essential difficulties and why there will never be a silver bullet. They are listed as: complexity, conformity, changeability, and invisibility. A example of one of these essential difficulties is google code repo. The monolithic model that google made surely makes it easier to comprehend the complexity of the codebase, since there is no coercion of repository boundaries between dependencies. However, as the scale increases, code discovery can become more difficult. Which many problems that arise in software engineering is the nonlinear increase in size. So, what can we do to fix these problems? We should reuse, incremental development and nurture your software engineers. I think nurturing our software engineers and make sure they follow the right path and ethics. Software developers are a huge part of today society. They are the one protecting our infrastructure. They write 1000 of lines of code today to model today society and impact people live. They protect and secure our information from hackers. Software engineering is in everything: banking, surveillance, drones, security, cars and etc.. They truly impact the world .