SWED Übung 9

Aufgabe 1)

The number of defects left in your software at delivery has a direct impact on product support:

* More defects mean users are more likely to encounter issues, leading to a higher volume of support tickets, emails, and calls.
* Support costs increase because the team must spend more time investigating, reproducing, and resolving reported problems.
* Customer satisfaction drops as users face more bugs, which can harm your product’s reputation.
* Development resources are often redirected from new features to fixing urgent issues, slowing down progress.

Aufgabe 2)

Pro:

* Deep code knowledge: Developers know the code best and can target complex logic
* Faster feedback: Bugs are found and fixed quickly during development
* Responsibility: Encourages ownership and code quality
* Efficiency: Immediate fixes avoid long bug-report cycles
* Better coverage: Can test hard-to-reach code paths

Cons:

* Bias/blind spots: Developers may overlook their own mistakes
* Confirmation bias: Tendency to test only what they think works
* Limited perspective: May miss real-world usage scenarios
* Lack of testing expertise: May not know all testing techniques
* Time pressure: May skip thorough testing to meet deadlines

Aufgabe 3)

Valid & reachable HTTPS URL

* Example Input: <https://www.google.com>
* Expected Result: Monitoring works, detects changes

Valid & reachable HTTP URL

* Example Input: <http://www.google.com>
* Expected Result: Monitoring works, detects changes

Invalid protocol

* Example Input: [htpps://www.google.com](https://www.google.com)
* Expected Result: Should not crash, returns false

Invalid format

* Example Input: [not](https://www.google.com) a url
* Expected Result: Should not crash, returns false

Empty string

* Example Input: “”
* Expected Result: Should not crash, returns false

Unreachable but valid URL

* Example Input: <https://sinnlos.domain.abc>
* Expected Result: Should not crash, returns false

Aufgabe 4)

Regression testing is the process of re-running existing tests after changes (like bug fixes or new features) to ensure that previously working functionality has not been broken. The main goal is to detect if new code changes have introduced unexpected errors (regressions) in parts of the software that used to work correctly.

When: After every code change, especially bug fixes and new features.

Why: To maintain software stability and prevent old bugs from reappearing.