**Risk Management**

Risk Management is the process of identifying, evaluating and prioritizing risks that could impact cost, timeline or quality.

Why is it important to think about Risk Management when determining Cost Estimation?

**Common Risks**

**Scope Creep:** Uncontrolled changes or continuous growth in a project’s scope

**Technical Debt:** Costs associated with delayed or suboptimal coding practices, continually refactoring code

**Resource Availability:** Risks due to unavailability of key personnel or tools

Single Point of failures

**Mitigation**

**Contingency Budget:** Setting aside a portion of the budget for unforeseen expenses

**Regular Audits/Demos:** Conducting periodic project reviews to catch issues early

**Risk Sessions:** Holding sessions to identify a plan for potential risks

**Quality Assurance Costs**

Quality Assurance Costs involve the expenses related to ensuring the software meets quality standards before release

**Manual Testing Costs:** Costs for manual testers, test case creation and execution

**Initial Setup:** Writing test cases, preparing test environment

**First Cycle:** manual execution of test cases

**Long-Term Maintenance:** 5 cycles ongoing manual testing effort

**Automated Costs:** Initial setup, tool licenses, and maintenance of automated test scripts

**Initial Setup:** Setting up automation framework, writing scripts

**First Cycle:** Running automated tests, fixing initial issues

**Long-Term Maintenance:** 5 cycles maintaining scripts, re-running automated tests

**Regression Testing:**  Ongoing costs to ensure new code doesn’t break existing functionality

Discuss professor’s point about not hiring testers immediately, waiting a month or so to start hiring Testers

**Short-Term vs. Long-Term Costs:** while automation has high upfront costs, it reduces long-term testing expenses

**Improved Efficiency:** Automation can reduce testing time and increase coverage

**Return on Investment (ROI) of Automation:** Demonstrate the return of investment over multiple project cycles of automation vs manual testing

**Total Cost of Ownership**

Total Cost of Ownership (TCO) includes all costs associated with a software product over its entire lifecycle, including development, deployment, maintenance, and end-of-life disposal

Highlight that the TCO provides a more comprehensive view of the financial impact than initial development costs alone

**Initial Development Costs:** Include labor, tools, installation and setup, and infrastructure

**Maintenance and Support:**  Ongoing costs for updates, bug fixes, and technical support

**Training and Onboarding:** Costs for educating users and new developers on the system

Show how these costs accumulate over time

**End of Life:** Data Migration, Decommissioning, System Replacement, Compliance and Disposal

**Efficient Design:** Invest in a scalable, maintainable software to reduce future costs

**Outsourcing Non-Critical Components:** Reduce TCO by outsourcing maintenance of non-core elements

**Regular Reviews and Updates:** Having regular demos can keep the team up to date and make sure all parties/stakeholders are on the same page. It is also important to ensure cyber security is coming to the demos so the don’t blind side you at the end of the project, when all of your code is insecurly developed with no cyber security protections.

Show pie chart of breakdown by major categories (Development, maintenance, training, etc)