

IT 323 Case Study Analysis Activity: Investigating Real-World Software Problems Using Scientific Thinking

Instructions:

1. **Form small groups (4-5 members).**
 2. **Find a real-world software problem** that has been documented in industry reports, academic papers, news articles, online case studies, or any real-world problems.
 3. **Analyze the problem** using the four steps of **scientific thinking in software engineering**:
 - **Characterize** (Identify and define the problem)
 - **Hypothesize** (Suggest a solution)
 - **Predict** (Estimate the expected results)
 - **Experiment** (Design an approach to test the solution)
 4. **Prepare a case study report** and present your findings to the class.
-

Case Study Analysis Guidelines:

1. Characterize (Identify the Problem)

- Describe the **real-world software issue** you have chosen.
- Provide background information (What is the system? Who are the users? What is the impact of the problem?).
- Collect **data or evidence** from reliable sources (e.g., articles, performance metrics, user complaints).
- Identify the potential causes of the problem.

Example:

- **Case:** An e-commerce platform experiences frequent website crashes during flash sales.
- **Characterization:** The site cannot handle a high volume of concurrent users, leading to downtime and revenue loss.

2. Hypothesize (Suggest a Solution)

- Based on your problem characterization, propose a **hypothesis** about the root cause of the issue.
- Suggest **possible solutions** that could resolve the problem.
- Provide **technical reasoning** for your hypothesis.

Example:

- **Hypothesis:** "The website crashes because the database queries are inefficient under high load. Optimizing database indexing and implementing load balancing should improve performance."

3. Predict (Estimate the Outcome)

- What improvements do you expect if your hypothesis is correct?
- Define measurable metrics (e.g., **page load time reduced by 30%, error rate decreased by 50%**).
- Explain how you will evaluate the success of the solution.

Example:

- **Prediction:** "If we optimize queries and add caching, response time will decrease from 5 seconds to under 2 seconds, and server uptime will improve by 90% during peak traffic."

4. Experiment (Test the Hypothesis)

- Design an **experiment** to test the proposed solution in a **controlled environment**.
- Identify the **tools, frameworks, or techniques** you will use.
- Describe the **steps of implementation** and how you will collect performance data.
- Discuss possible challenges and alternative solutions if the experiment does not yield the expected results.

Example:

- **Experiment Plan:**
 - Apply database indexing and caching on a test server.
 - Simulate high-traffic conditions using load testing tools (e.g., JMeter, Gatling).
 - Measure and compare response times before and after optimization.

- Roll out changes gradually and monitor real-world performance.
-

Deliverables:

Each group must submit:

1. **A detailed case study report** covering all four steps.
2. **Software solution presentation.**
3. **A PowerPoint presentation** summarizing their findings.

The report and presentation should include:

- ✓ **A clear problem statement**
- ✓ **Evidence-based analysis** of the issue
- ✓ **A structured hypothesis and prediction**
- ✓ **A well-defined experimental plan**