Week Five: Software Evolution & Architecture

Date: November 13th

🎯 The 80:20 Rule

- Focus on discovering **unknown unknowns** early
- ✅ Effective design anticipates problems before they surface

💡 Interpretation: 80:20 Rule

Software engineering is not about removing every risk.
The goal is to identify the most significant risks as early as possible.
Addressing them sooner prevents costly rework later in the project.

🏗️ MVVM Architecture

- **Model–View–ViewModel** separates concerns:
    - Model → data and business logic
    - View → UI representation
    - ViewModel → mediator between model and view

💡 Interpretation: MVVM

Separating responsibilities makes systems easier to test and maintain.
Logic stays in the Model and ViewModel, while the View focuses only on presentation.
This division allows teams to work in parallel without interfering with each other.

🐞 Debugging Philosophy

Good software engineering isn't just finding bugs — it's designing code so bugs **reveal themselves**.

💡 Interpretation: Debugging

Well-structured systems expose problems instead of hiding them.
Clear error handling and strong typing make failures visible, so they can be fixed quickly.
This reduces time spent searching for hidden issues.

😃 Happy Engineers = Successful Engineers

The most successful software projects are built by teams that enjoy their work.

💡 Interpretation: Happiness

Satisfied engineers produce higher-quality work.
When frustration is reduced through good design and clear processes, teams are more productive and creative.
Positive environments lead to better long-term outcomes.

🐾 HW4 Example: Animal Shelter API

Applying Week 5 Principles:

- MVVM
  - **Model** → Database tables: Animals, Adoptions, Users
  - **View** → Frontend UI: animal listings, adoption forms, user registration
  - **ViewModel** → API endpoints: `/animals`, `/adoptions`, `/users`
    This separation ensures the UI does not directly manipulate the database, making the system easier to test and extend.

🐾 HW4 Example: Animal Shelter API

Applying Week 5 Principles:

- **Debugging Philosophy**
  Errors surface clearly through endpoints. For instance, if an adoption request references a non-existent animal, the API returns a clear error message. This design makes problems visible instead of hidden.