Driving Design with TFD

Dung Le Hoang

NASH TECH

The Power to Innovate

AGENDA

- Benefits to Design
- Singletons
- Coupling
- Concern Separation
- Dependency Inversion

CHANGE DESIGN WITHOUT FEAR

Writing tests first mean writing testable code

Writing testable code happens to result in well-designed code

SINGLETONS

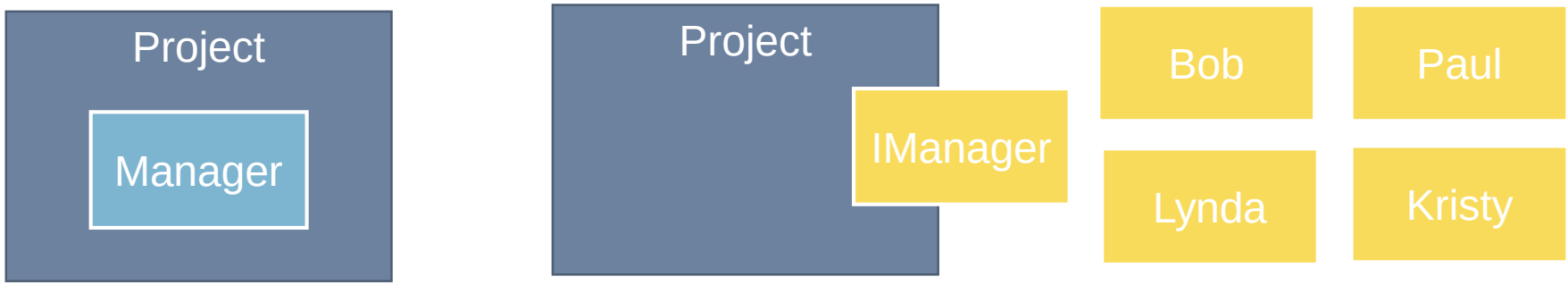
- Often provide global access to resources
  - Hiding implementation “details”
  - Actually hiding other dependencies
- The singleton pattern itself violates Single Responsibility Principle
  - The functionality of the class
  - The creation and management of the singleton instance
- Promote tight coupling
  - Unable to supply alternative implementations
- Stat is a problem for singletons as long as the program is running

COUPLING

- Given two lines of code, A and B, they are coupled when B must change behavior only because A changed
- Limits ability for design to change

DEPENDENCY INVERSION

- High-level modules should not depend on low-level modules. Both should depend on abstractions.
- Abstraction should not depend upon details. Details should depend upon abstractions.



SEPARATION OF CONCERNS

- A classes has one and only one purpose

Clear intent for the next programmer

Simpler code implementation

Fewer dependencies

Fewer defects

SUMMARY

- Benefits to Design
- Singletons
- Coupling
- Concern Separation
- Dependency Inversion