

Proof of Concept

Lecturer: Ngo Huy Bien
Software Engineering Department
Faculty of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
nhbien@fit.hcmus.edu.vn

Objectives

- To create a *proof of concept*.



References

1. Karl Wieggers and Joy Beatty (2013).
Software Requirements. Microsoft Press.



Can You Really Do Something?



Previous Similar Projects



A Difficult Feature Implementation





Proof of Concept (I)

- A *proof of concept* or a *proof of principle* is realization of a certain method or idea(s) to demonstrate its feasibility, or a demonstration in principle, whose purpose is to verify that some concept or theory has the potential of being used.

The purpose is to verify that some concept or theory is probably capable of being useful.



Proof Of Concept (II) [1]

- A *proof of concept*, also known as a vertical prototype, implements a slice of application functionality from the user interface through all the technical services layers.
- A proof-of-concept prototype *works like the real system* is supposed to work because it touches on all levels of the system implementation.
- To make the results meaningful, such prototypes are constructed by using *production tools* in a production-like operating environment.



When?

- Develop a proof of concept
 - when you're uncertain whether a proposed *architectural approach* is feasible and sound, or
 - when you want to optimize *algorithms*, evaluate a proposed *database schema*, confirm the soundness of a *cloud solution*, or *test critical timing requirements*.



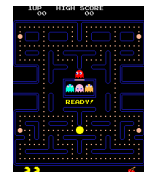
Create a Proof of Concept

- Concrete *difficult* problem
- Input – Output
- Solution (Theory/Design)
 - Intuition
 - Formulas
 - Transformation steps
- Implementation (Practice)
 - Libraries
 - Concrete results



Example Question and Answers

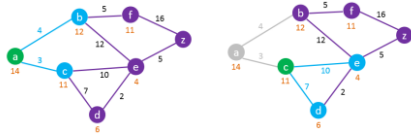
- How to display a background?
- How to display the Pacman?
- How to make the Pacman respond to user inputs?
- How to make the Pacman constrained to within the walls?
- How to display a ghost?
- How to make the ghost move automatically?
- How to detect some collision?
- How to display some dots?
- How to make the Pacman eat dots?
- How to find out that the level is complete?



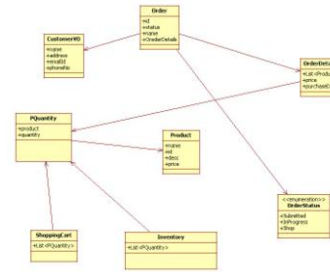
Algorithms



"What is the shortest route from A to B?"



Domain Model



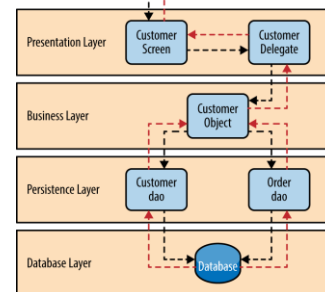
Database Model

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

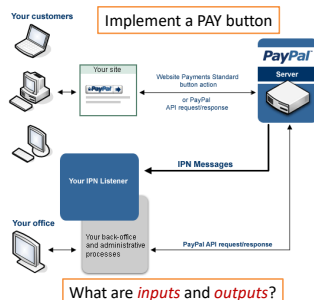
student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Architecture



Testbed

- An external component is **tested apart** from the program/system it will later be added to.
- A skeleton framework is implemented around the module so that the module behaves as if already part of the larger program.



Small Exercises

Testing a **new** library, platform or technology.



- Just **simple (but useful)** applications:
 - Use C/C++ (OpenCV) with Java code (Android) to speed up loading and saving an image.
 - Write a login feature with Xamarin.
 - Write a login feature with React Native.



Pilot

- A *pilot* (or trial) uses the full production system and tests it against a subset of the general intended audience.
- The *reason* for doing a pilot is to get a better understanding of how the product will be used in the field and to refine the product.
- This clearly defines *how the system must operate* within its environment.
- *Example*: Migrating data from SQL 2005 to SQL 2016.



Why Proof of Concept?

- Helps in *testing partial solution*.
- Helps in analyzing *requirements* completely.
- Helps in identifying and planning for *risks*.
- Helps in making *cost-benefit* analysis.
- Helps in making plan for *training* developers for implementing the system.
- Can be *reused* for actual system.



Thank You & See You Again

