Lengfan Yan

1. We have studied four architectural styles: data-centered, data-flow, call-return, and layered. Now give an example software project for which you think each of these styles would fit the best. Briefly justify the choice of architectural style for the example project you chose. [12 points]

data-centered: a news media website, all client uses tools to get access to the same data of news website given, but none of client able to change the data.

data-flow: restaurant app example showed in class. The information of order flows through app, order, product then to client

call-return: windows, main program is the windows it self, and able to call other software installed in system, like MS word, IE, files

layered: also operating systems like windows? Client usually unable to reach in inside layer of system, but only base on UI layer

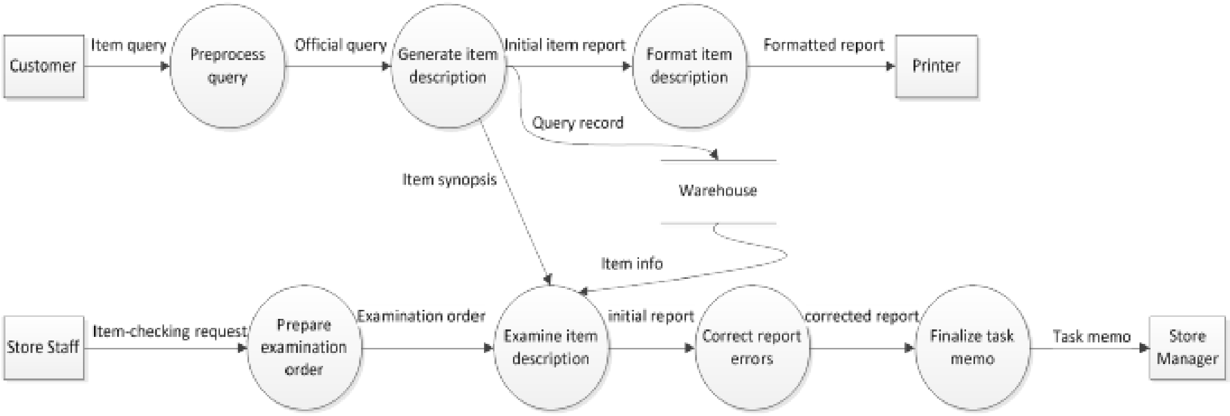
2. There are two measures of functional independence in software design: cohesion and coupling. Explain in your words what each of these measures means. Is it better to have higher/lower cohesion? Why? Is it better to have higher/lower coupling? Why? [18 points]

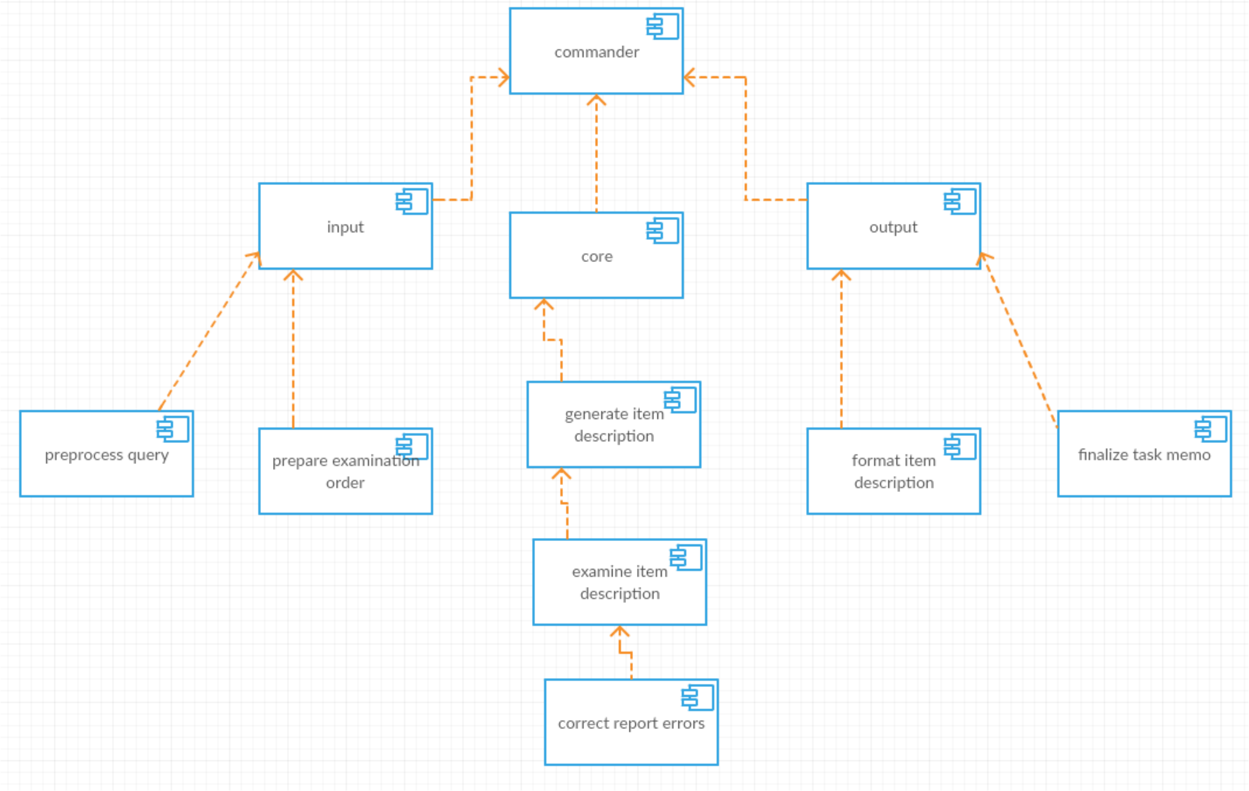
Cohesion is how well things inside a component work together. Coupling is how much different component work together.

We want high cohesion and low coupling. Since high cohesion makes a component do single job and makes it do it well. But high coupling will create more dependency between components, that change one thing will lead to need of change other, which we don’t want that.

Lengfan Yan

3. The following data flow diagram (DFD) is part of the flow models for the requirements of the ACE store management software. It describes the process of responding to a customer request for querying about an item. Derive the architecture design from this DFD, and draw the component diagrams that represent your architectural design. [20 points]


Customer
Item query
Preprocess query
Official query
Generate item description
Initial item report
Format item description
Formatted report
Printer
Query record
Warehouse
Item synopsis
Item info
Store Staff
Item-checking request
Prepare examination order
Examination order
Examine item description
initial report
Correct report errors
corrected report
Finalize task memo
Task memo
Store Manager


commander
input
core
output
preprocess query
prepare examination order
generate item description
format item description
finalize task memo
examine item description
correct report errors