Design Software that is easily modifiable

Software Modifiability

- Documentation: make other team members to understand your code

- Comment lines: " "

- Modular design: least amount of coupling between modules.
  Avoid globar variables.

- UML-level Simulation: perform UML-level simulation before Conducting code modification.

Class, Responsibility and Collaboration (CRC) cards.

Credit: Nils BrummondNotes.

A CRC cards is an index card that is used to represent the resposibility of classes and the interaction between the classes. CRC cards are an informal approach to object oriented modeling. The cards are created through scenarios, based on the system requirements, which model the behavior of the system.

CRC cards were introduced by Kent Beck and Ward Cunningham in 1989.

Why uses CRC cards?

- They are portable.
- They allows the participants to experience first hand how the system will work. without relying upon computer tool.

1. The CRC card session

The Group

The ideal group size for a CRC card session is five or six people. This size generally allows everyone to productively participate. The core group should be composed of developers, domain experts, and an object-oriented technology facilitator.

The Cards.

The cards should look something like this:

Class Name	
Superclasses:	
Sub classes:	
Responsibilities	Collaborators

The exact format of the card can be customized to the preferences of the group, but the minimal required information is the name of the class, its responsibilities and the collaborators. The back of the cards can be used for a description of the class.

The sessions

- Creating class

The first step in modeling a system in the object-oriented paradigm is to identify the class in the problem domain.

problem statement or requirement → classes.
(nouns)

- Responsibilities.

problem statement or requirement → responsonsibilities.
(verbs)

Once a reasonable set of classes have been assigned to the group, responsibilities can be added.

- Superclasses and Subclasses.

Superclasses and subclasses can be defined any time they becomes obvious.

- Attributes

Attributes of class don't really need to be defined any time soon. They are implementation details. (on the back of the card).

- Scenario Execution

These are the heart of the CRC card session. Scenarios are walk-throughs of the functions of the system in detail. Take required functionality from the requirements in document, and use this as a scenario. Basically, we need to decide which class is responsible for this function. If no appropriate class exists, you need to make one.

2. CRC cards for analysis

Analysis is the process of modeling what a system does, not how it does it.

Activity 1.

problem statement

This application will support the operations of a technical library for an R & D organization. This includes the searching for and lending of technical library materials, including books, videos, and technical journals. Users will enter their company ids in order to use the system; and they will enter material ID numbers when checking out and returning items.

Each borrowers can be lent up to five items. Each type of library item can be lent for a different period of time (book 4 weeks, journal 2 weeks, video 1 week). If items are returned after their due date, the user will be charged a fine, based on the type of item (book $1/day, journal $3/day, video $5/day).

Materials will be lent to employees with no overdue lendables, fewer than five items, and total fines are less than $100.00.

with this problem statement, identify classes and assign them to the group members:

* library, material, book, video, journal, fine, date, employee.

Now let's try the first scenario: "what happens when John checks out a book if he has no fine and one outstanding book?"

Group members can analyze different scenarios to find out holes in requirements.

2. CRC Cards for Design

Design is the process of determining how a system does.

Things to consider in Design

- * Target environment
- * Language
- * Choice of supporting software components
- * performance requirements
- * memory, security.

Attributes.

Now it is the time to record the attributes of the classes on the back of the cards.

Example I: A simple model-view-controller interface framework

View	
Render the model Transform Coordinates	Controller Model

Controller	
Interpret user input Distribute Control	View Model

Model	
Maintain problem related info. Broadcast change notification	