



**MACBANK**

---

# **ABM Machine**

**SE 3K04 : Software Development for Engineering**

**Instructor : Dr. Kamran Sartipi**

**By the ABM Group Inc.**

**Mechatronics Engineering  
McMaster University**

**Amer El-Kalache**

**Bahy Mehany**

**Mohamed Ismail**

Thursday, November 29, 2007



# Outline

---

- Overview (Mohamed)
- Requirements (Amer)
- Design (Amer & Mohamed)
- Implementation (Bahy)
- Verification (Mohammed)
- Product Demo (Bahy)
- Maintenance (Bahy)
- Conclusion (Amer)
- Lessons Learned (Mohamed)

# Overview

- **Software engineering:** the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software systems.
- **Importance:**
  - Tackles the software crisis
  - Creates systematic approach to programming
- **Applications:**
  - Healthcare
  - Automotive Industry
  - Automation

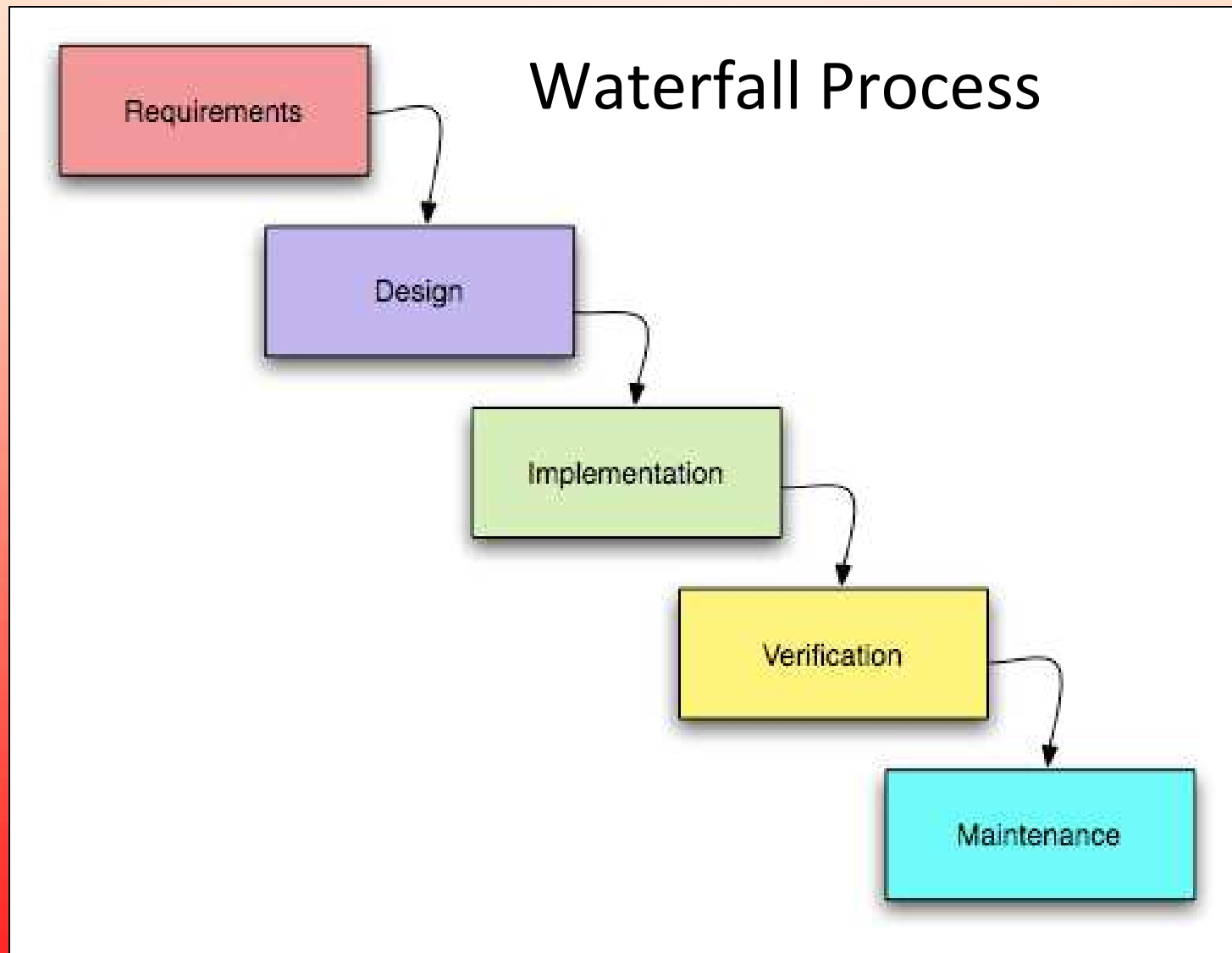


## Professional Responsibilities in Software Engineering

- Accept individual responsibility
- Focus on the real problem
- Produce reviewable designs
- Consider maintainability of software systems
- Respect IP rights for patents and copyright
- No computer misuse
- P.Eng applicants write ethics and law exams

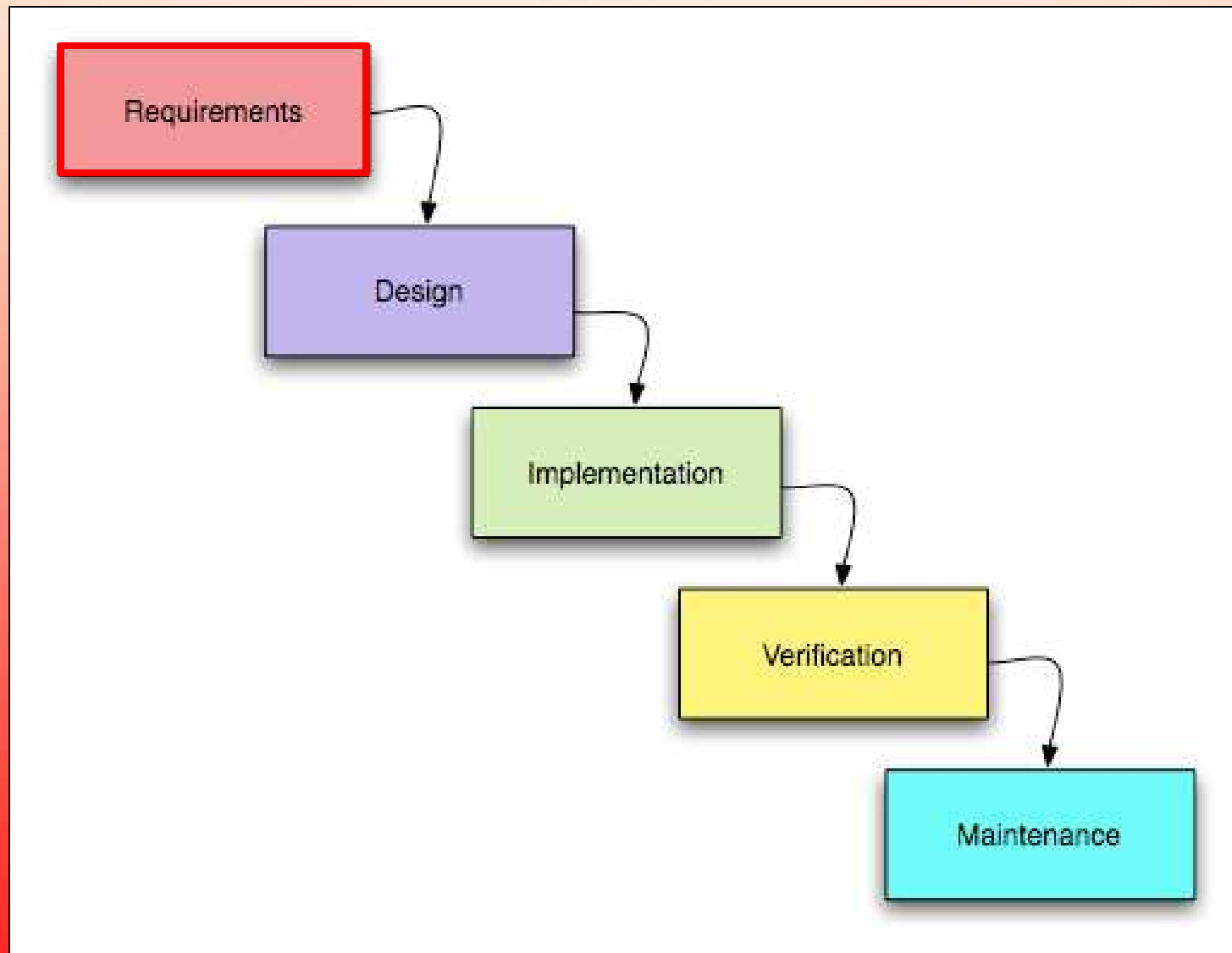


# MACBANK Phases of the Software Life Cycle Process





# Requirements



# Requirements

- Request for Proposal
- Software Requirement Specification
- Benefits:
  - Provides a good guideline for producing a high level design
  - Can be used as test cases after implementation



- MacBank CEO
  - Dr. Kamran Sartipi
- Employees
  - Naser Faramarzpour
  - Priya Jayaratna



## Process:

- Interview
- Produced an expanded list of features (RFP)
- Limited the list of features and produced the SRS



# Request for Proposal

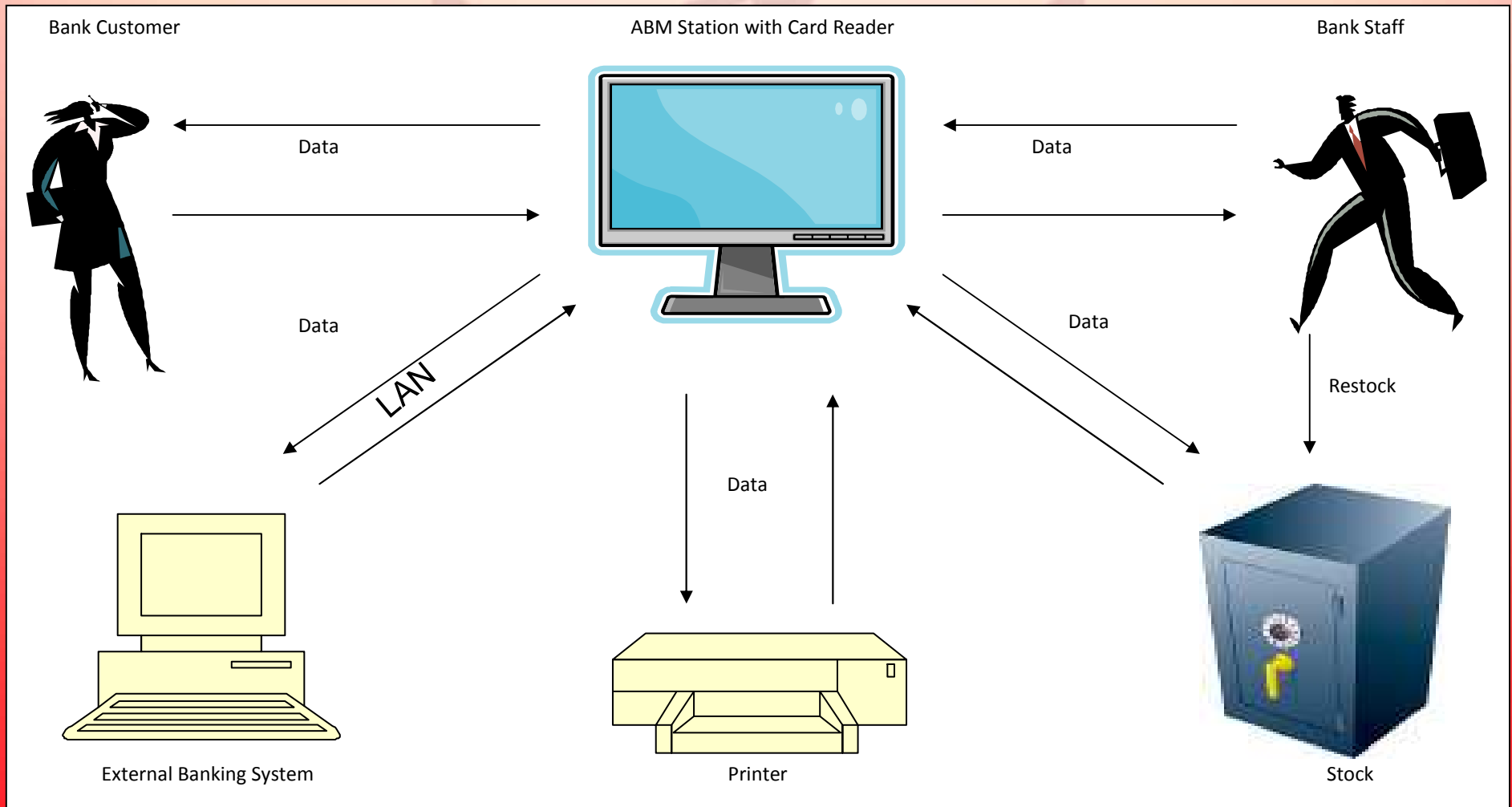
- Client (MacBank) was interviewed to gather required high-level requirements
  - ABM must have usable and friendly interface
  - Maximum cash withdrawals of \$500 at a time
  - 4-digit PIN number
  - Customer must be able to withdraw, deposit, and transfer money
  - Stock status must be evident and staff must be notified when low on stock
- Purpose of RFP – First step to binding a contract



**MACBANK**

# SRS: Software Requirement Specification

## Product Perspective

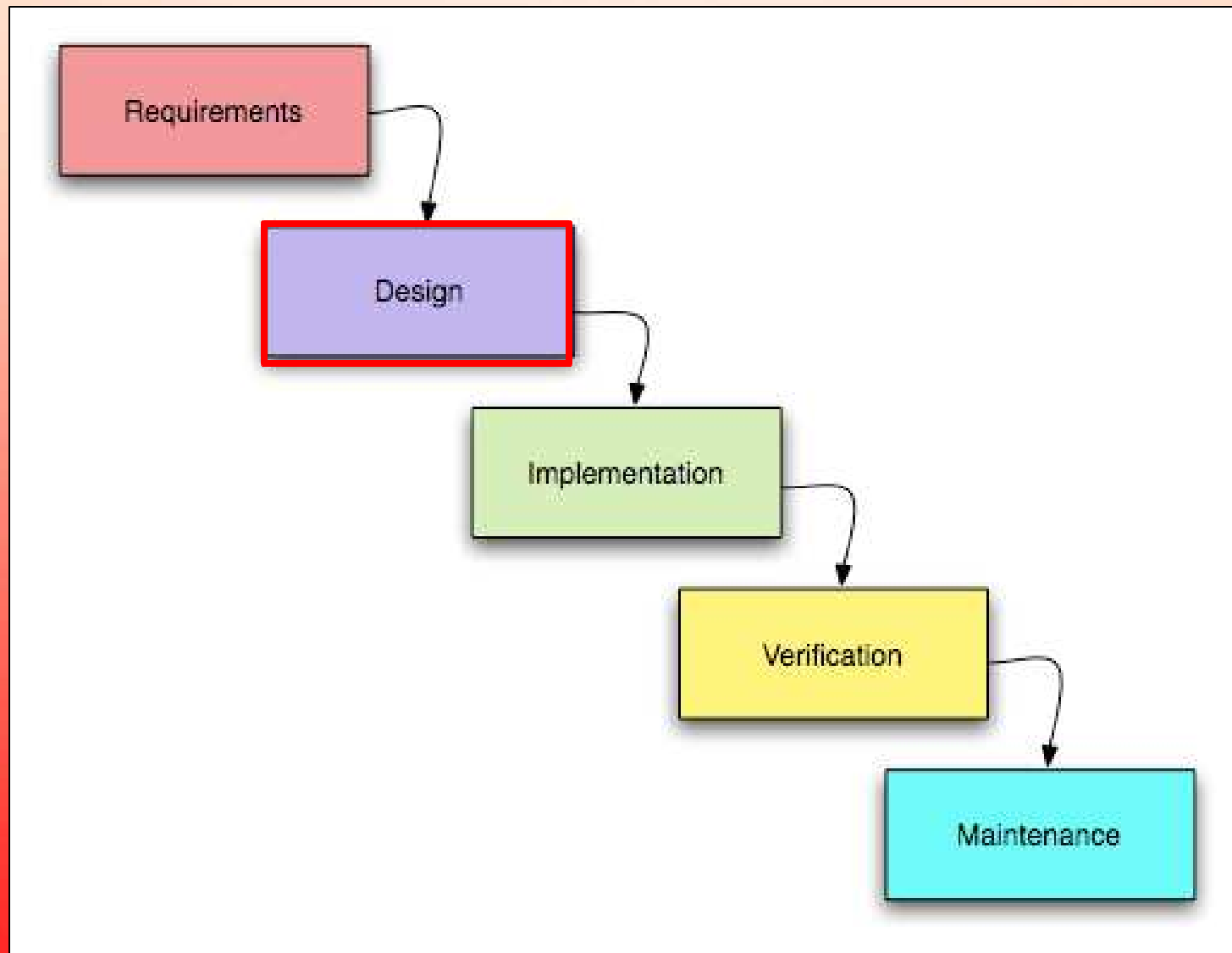




- **Standards and Goals**
  - IEEE Templates
  - Software Quality Attributes:
    - **Reliability** – no crashes should happen during customer use
    - **Flexibility** – Should operate on different machines
    - **Usability** – User-friendly, GUI, large font, clear explanations
    - **Maintainability** – Modular code that is well commented for future alterations



# Design



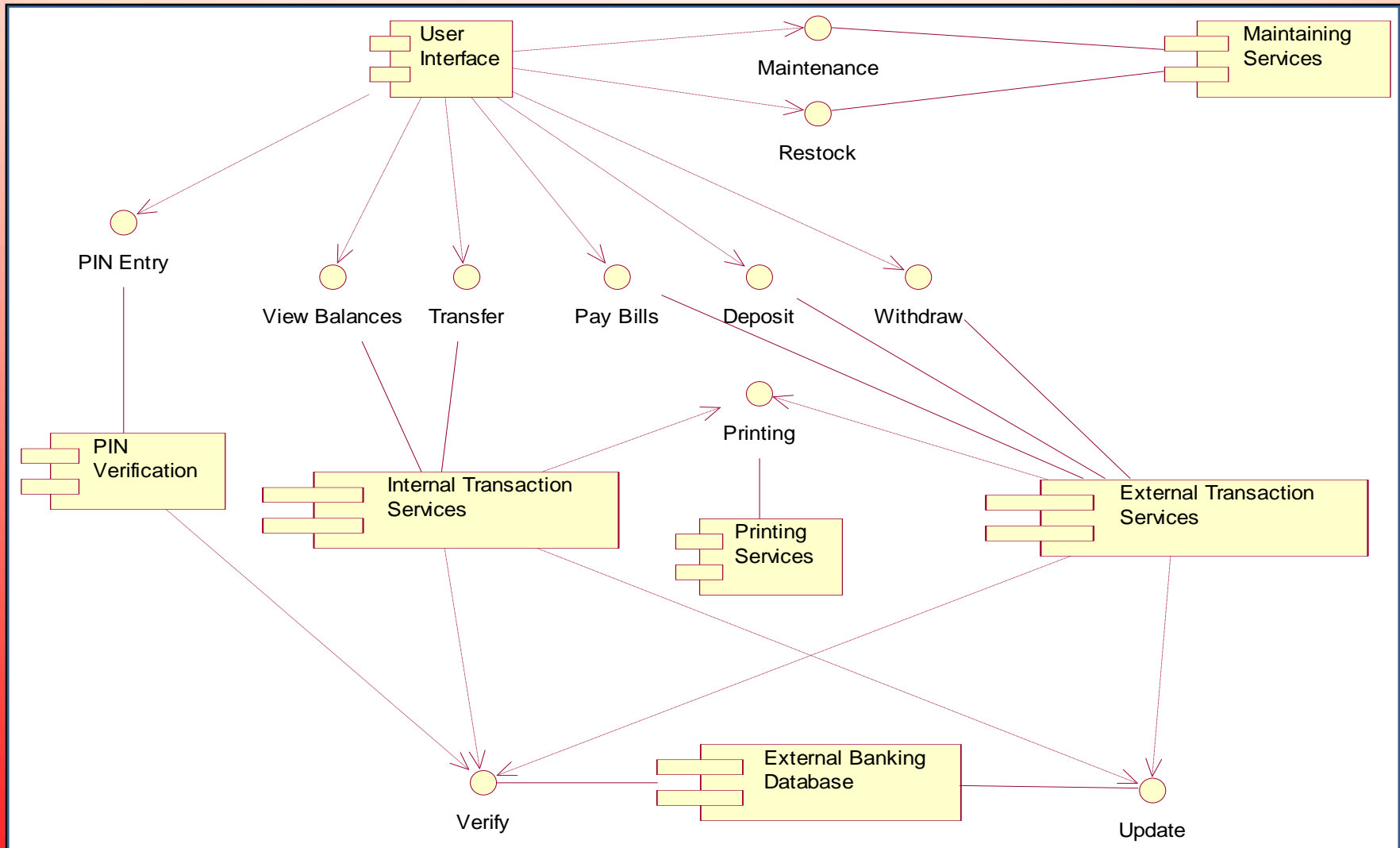


# Design

- SDS: Software Design Specification
  - High Level Design
  - Low Level Design
- Benefits:
  - Helped implement the ABM system
  - Helped design required software modules

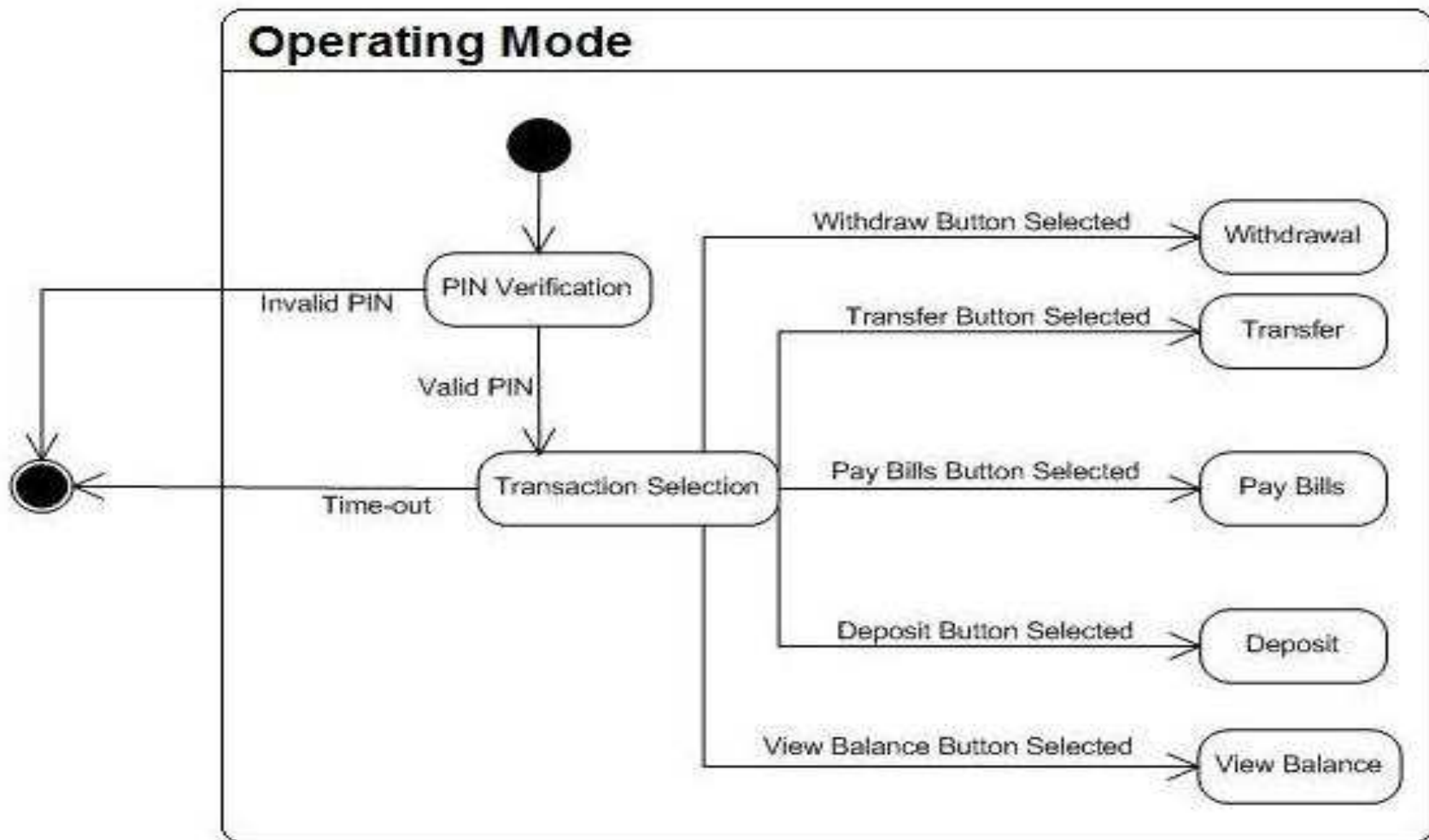


# SDS: Component Diagram





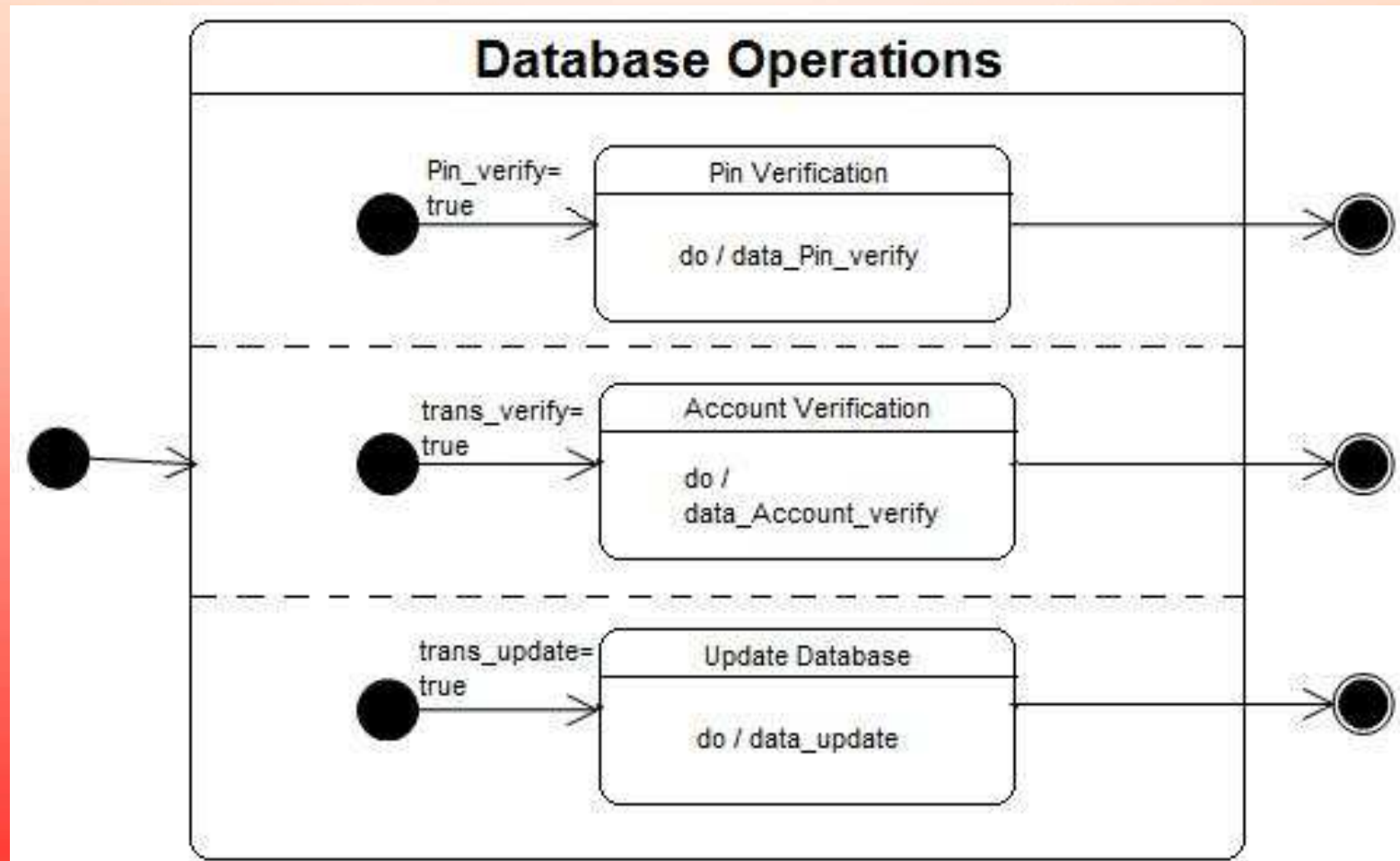
# Example of High Level Design







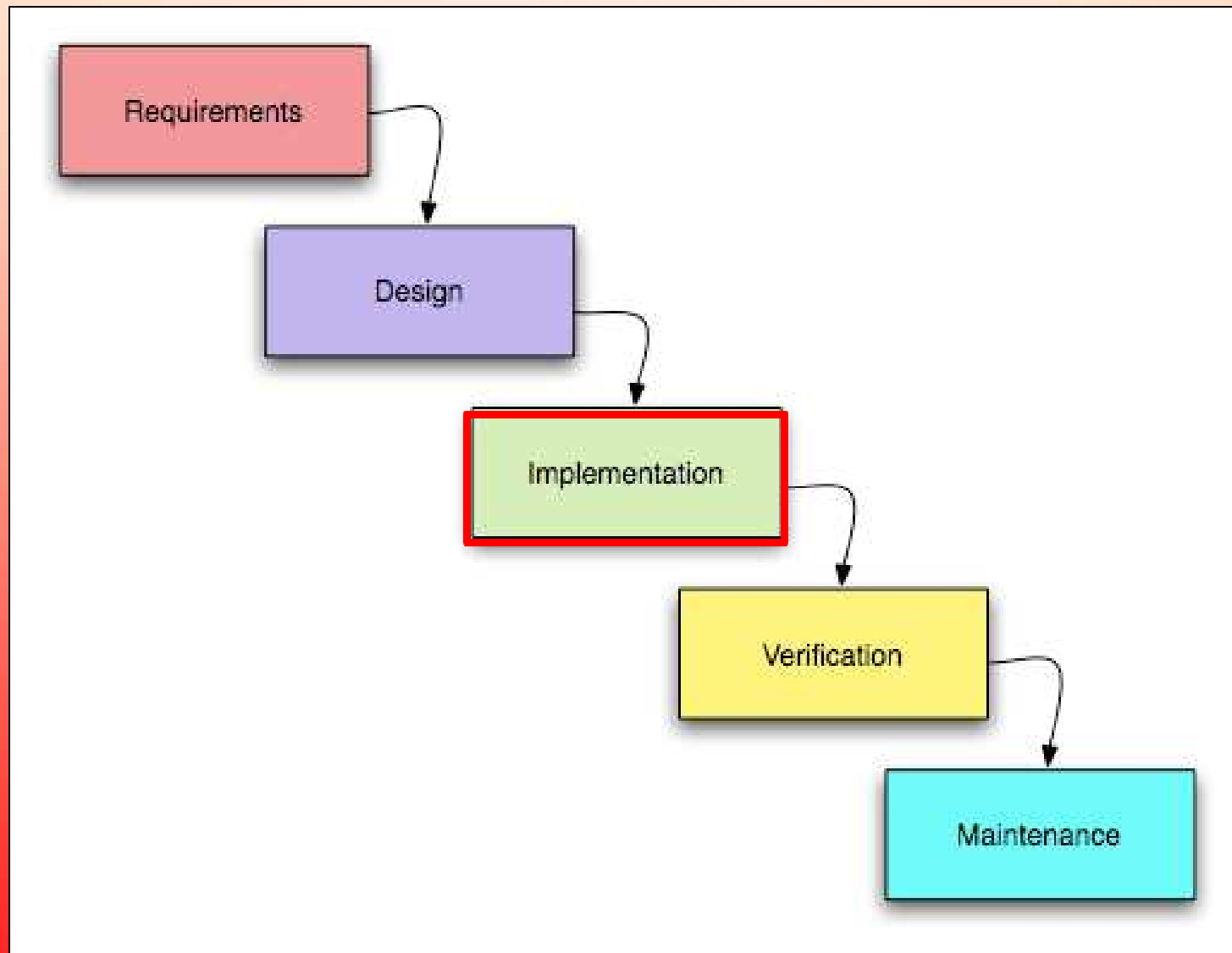
# Example of Low Level Design



**Questions?**



# Implementation



# Implementation: Database

ACCOUNT	CARD	PIN	CHEQ	CHEQ HOLD	SAV	SAV HOLD	CRED
000000000000,	00000000,	0000,	-----,	-----,	00000000.00,	00000000.00,	00000000.00
111111111111,	10000000,	1111,	00100000.00,	00000000.00,	00010000.00,	00000000.00,	00001000.00
222222222222,	20000000,	2222,	00200000.00,	00000000.00,	00020000.00,	00000000.00,	00002000.00
333333333333,	30000000,	3333,	00300000.00,	00000000.00,	-----,	-----,	00003000.00
444444444444,	40000000,	4444,	00400000.00,	00000000.00,	00040000.00,	00000000.00,	00004000.00
555555555555,	50000000,	5555,	00500000.00,	00000000.00,	00050000.00,	00000000.00,	00005000.00
666666666666,	60000000,	6666,	00600000.00,	00000000.00,	00060000.00,	00000000.00,	-----
777777777777,	70000000,	7777,	-----,	-----,	00070000.00,	00000000.00,	00007000.00
888888888888,	80000000,	8888,	00800000.00,	00000000.00,	00080000.00,	00000000.00,	00008000.00
999999999999,	90000000,	9999,	00900000.00,	00000000.00,	00090000.00,	00000000.00,	00009000.00

**MACBANK**

# Implementation: Database

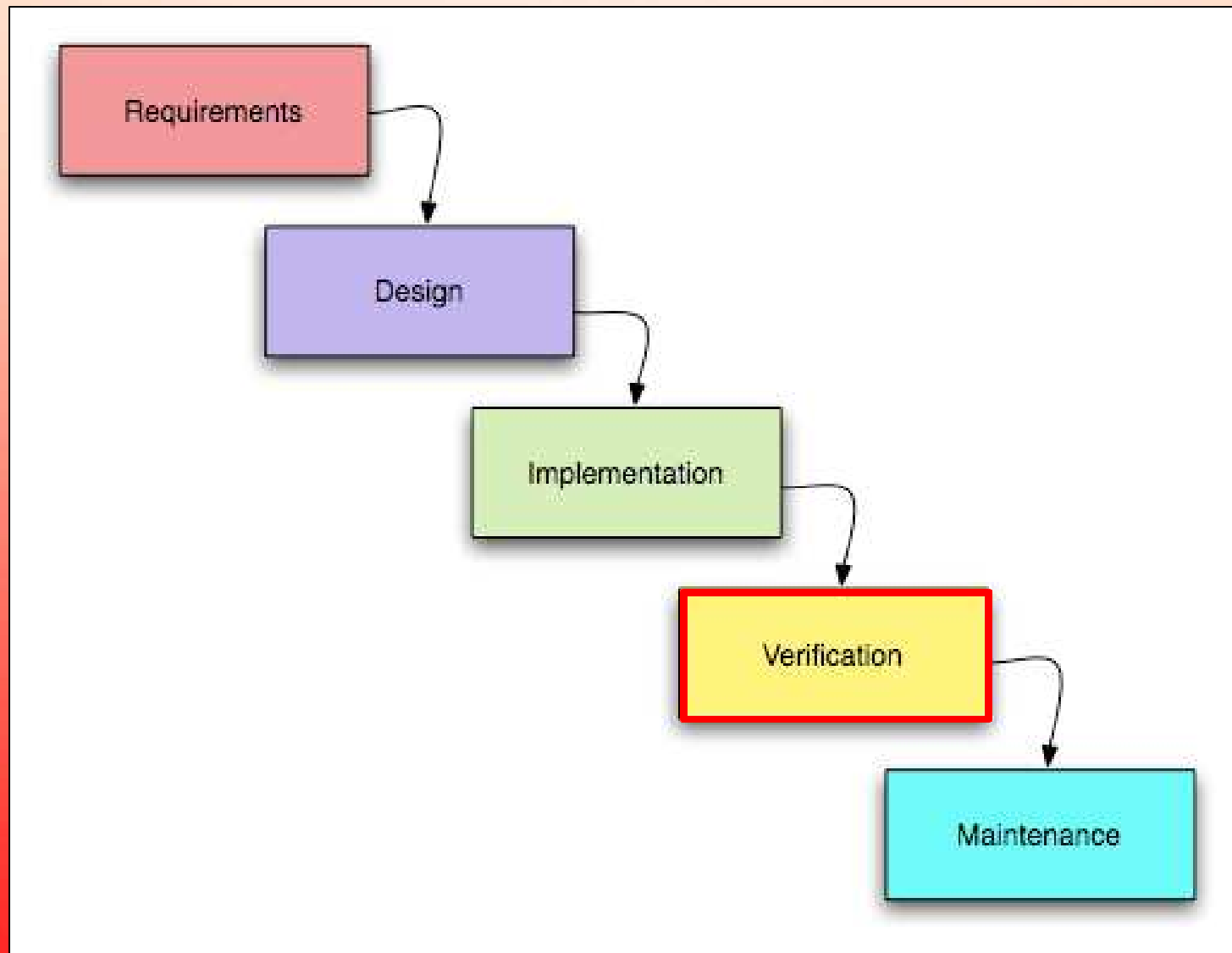
ACCOUNT	CARD	PIN	CHEQ	CHEQ HOLD	SAV	SAV HOLD	CRED
000000000000,	00000000,	0000,	-----	-----	00000000.00,	00000000.00,	00000000.00
111111111111,	10000000,	1111,	00100000.00,	00000000.00,	00010000.00,	00000000.00,	00001000.00
222222222222,	20000000,	2222,	00200000.00,	00000000.00,	00020000.00,	00000000.00,	00002000.00
333333333333,	30000000,	3333,	00300000.00,	00000000.00,	-----,	-----,	00003000.00
444444444444,	40000000,	4444,	00400000.00,	00000000.00,	00040000.00,	00000000.00,	00004000.00
555555555555,	50000000,	5555,	00500000.00,	00000000.00,	00050000.00,	00000000.00,	00005000.00
666666666666,	60000000,	6666,	00600000.00,	00000000.00,	00060000.00,	00000000.00,	-----
777777777777,	70000000,	7777,	-----	-----	00070000.00,	00000000.00,	00007000.00
888888888888,	80000000,	8888,	00800000.00,	00000000.00,	00080000.00,	00000000.00,	00008000.00
999999999999,	90000000,	9999,	00900000.00,	00000000.00,	00090000.00,	00000000.00,	00009000.00

- Database Demo

- **Language: Visual C#**
  - A simple, modern, general-purpose, object oriented programming language
  - Very similar in syntax to C
  - A hybrid between C++ and Java
- **IDE: Microsoft Visual Studio 2005**
  - Demo



# Verification



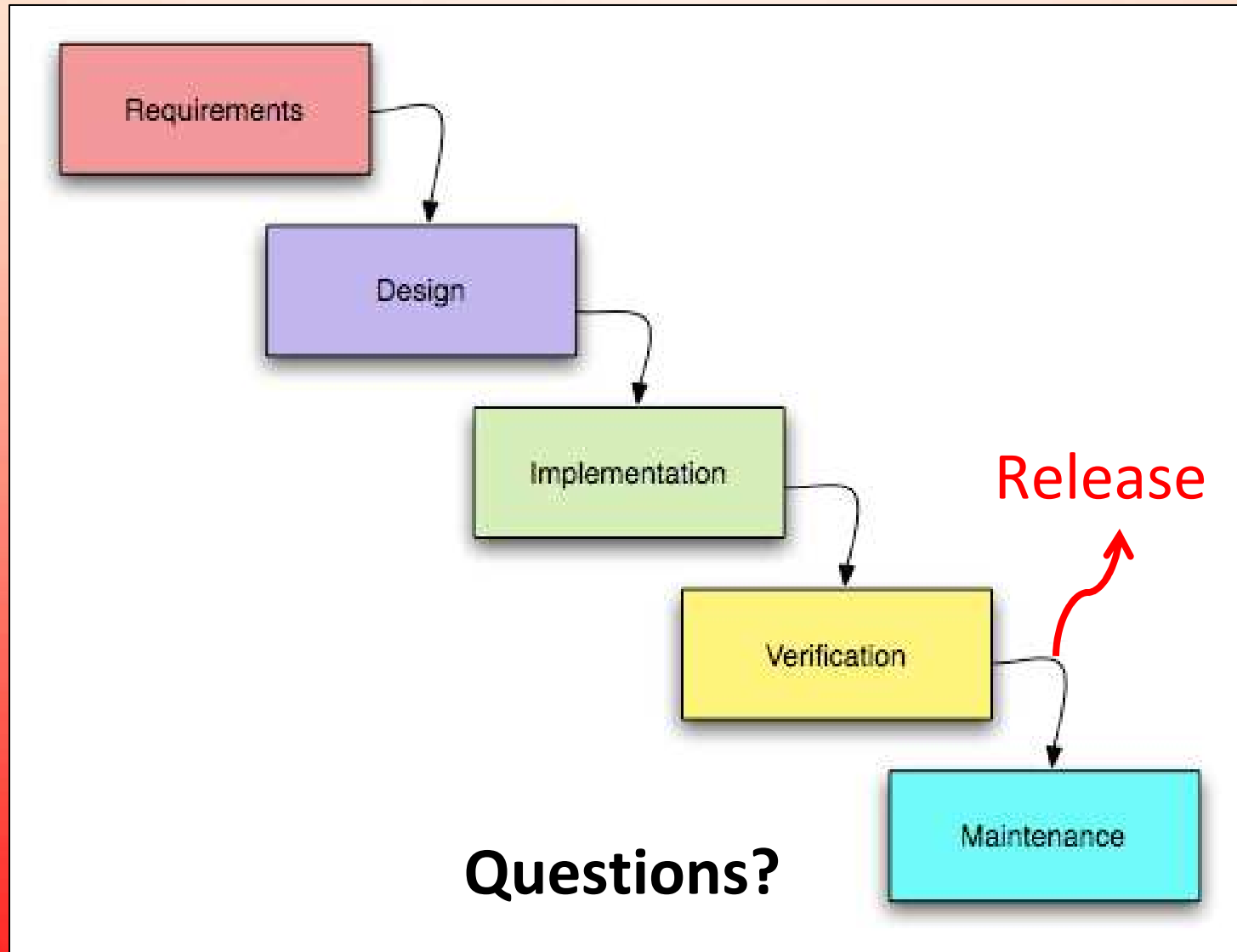
# Verification

- Used test cases to verify implementation
- These test case were acquired from the SRS
  - Primitive - Clicking inactive choices/buttons
    - Nothing
  - Medium - Withdrawing an amount greater than non-hold funds
    - Does no accept
  - Advanced: Restock so that amount becomes greater than \$10,000
    - Red light goes off



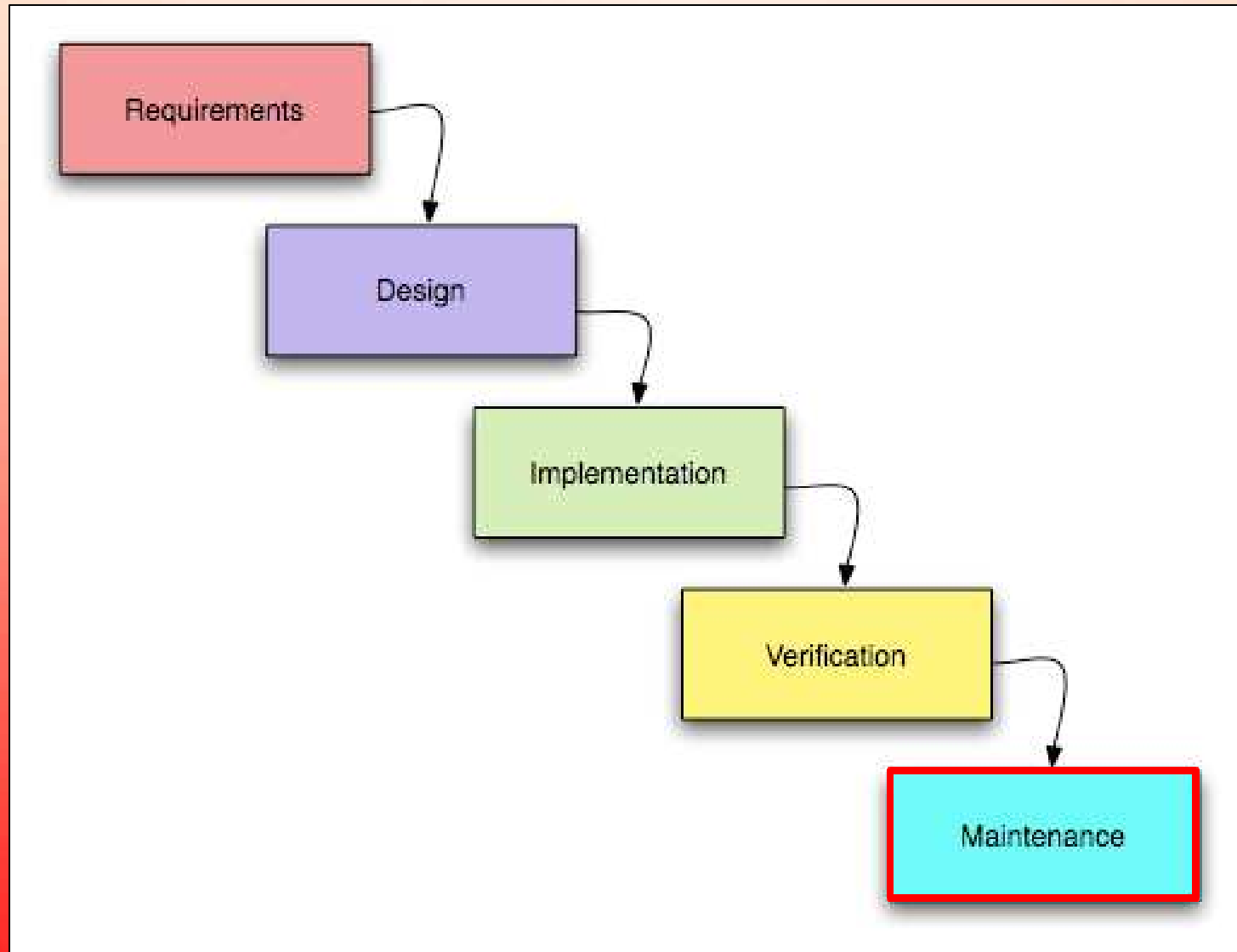


# ABM Demonstration (Finally)





# Maintenance



# Maintenance

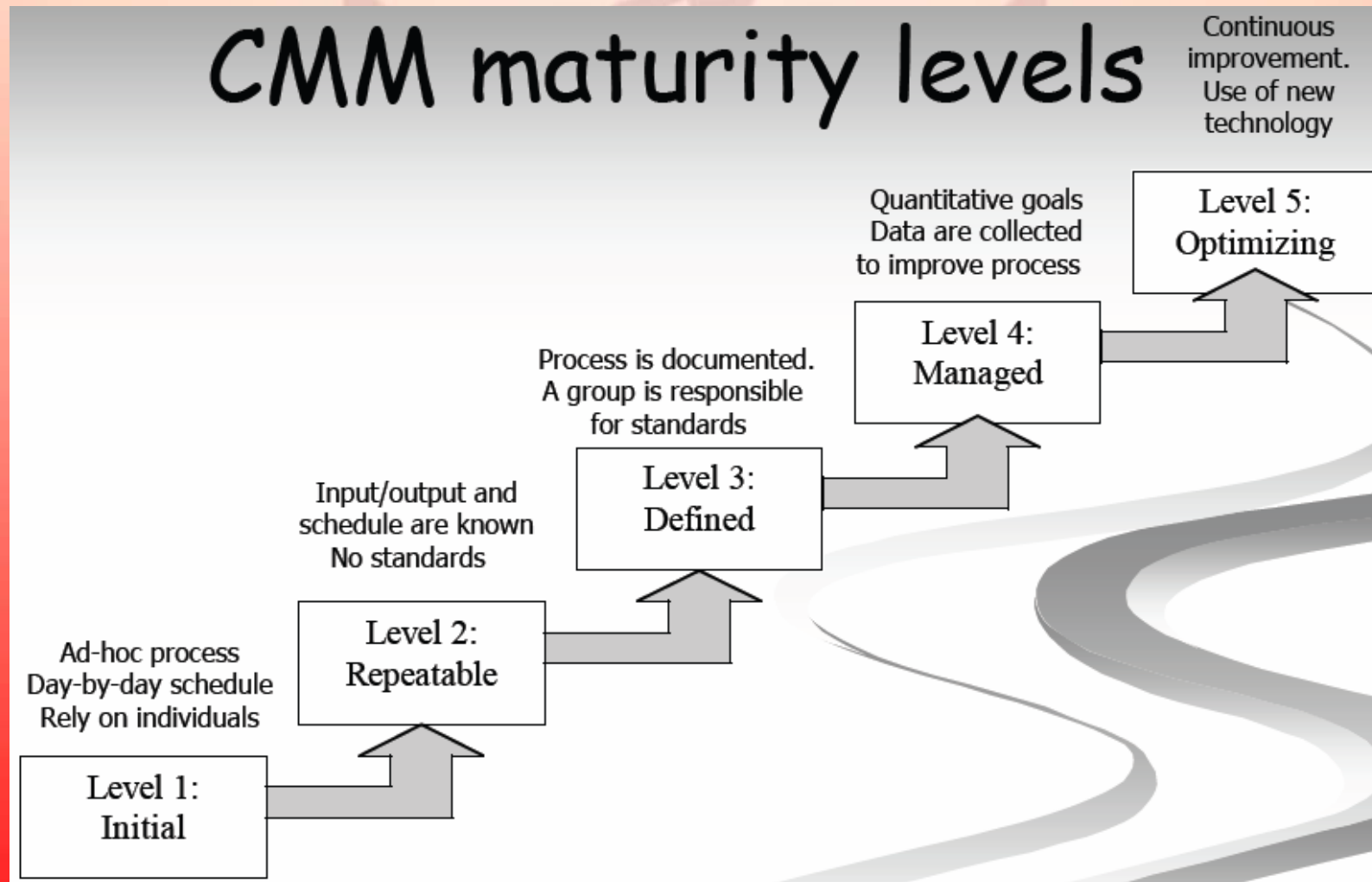
---

- Easy to locate and fix bugs
- System does not require a lot of maintenance, since thorough testing was carried out
- New developers can easily follow code and fix problems



# Conclusion

- Compare to Compatibility Maturity Model



# Conclusion

---

- Our Product was between level 2 (Repeatable) and level 3 (Defined)
- Level 2 involves the ability to do the same thing with another product
- Level 3 involves fixed standards

# Conclusion

Based on our Gantt Chart and since we have IEEE standards for documentation, we are somewhere between level 2 and level 3

ID	Task Name	Start	Finish	Duration	Oct 2007							Nov 2007		
					16/9	23/9	30/9	7/10	14/10	21/10	28/10	4/11	11/11	
1	Software Requirements Specification	9/18/2007	10/9/2007	16d										
2	Software Design Specification	10/9/2007	11/2/2007	19d										
3	Coding	11/2/2007	11/12/2007	7d										
4	Testing	11/12/2007	11/22/2007	9d										
5	Debuggin	11/12/2007	11/22/2007	9d										

# Lessons Learned

---

- The tiny design component of Eng 1D04 can be spread out into a 4 unit course
- Software Engineering is not only coding
- The importance and benefits of design in the software development process
- Team management and delegation
- Incorporation of diverse team member skills
- Importance of pair programming and modularization



# Feedback on Labs

---

- Good milestones to keep us working throughout the term
- TA's should have a more clear understanding of lab layout
- More feedback required on documentation
- Good synchronization points (RFP, SRS, Design)

**Questions?**