# Android Application Design

Software System Design

Zhu Hongjun

# Introduction

- Teaching hours
  - Theory：50 hours
  - Practice：20 hours
- Grading
  - Theory（50%）+ Practice（50%）=100%
  - Theory=25’(mid)+25’(fin)=50’
  - Practice=E (40') +MOOC(10’) =40'
  - E=20’(mid)+20’(fin)=30’
  - MOOC=5’(video)+5’(quize)=10’

# Contents

- Theory
  - Contents：Introduction、Android Architecture、UML and Java Programming、Software Design Methods and Tools、SQLite Database、Android File/Network/Multimedia Programming etc.
  - Evaluation：written test
- Practice
  - Contents：design tools and techniques, android app programming

# References

- Slides and Internet Materials
  - Android: http://developer.android.com/index.html
  - UML: http://www.uml.org/
  - Java: http://docs.oracle.com/javase/specs/jls/se7/html/
  - 《Android应用UI设计模式》Greg Nudelman著，袁国忠译
  - 《Object-Oriented Software Engineering: An Agile Unified Methodology》David C. Kung
  - 《Database System: The Complete Book》Hector Garcia-Molina, Jeffrey D.Ullman, Jennifer Widom
  - Introduction to Mobile Application Development using Android
    - https://courses.edx.org

# Software

- JDK
  - Download from: www.oracle.com
- Android Studio
  - Download from: http://developer.android.com
- Android SDK
  - Download from: http://developer.android.com
- Tomcat
  - Download from: http://www.apache.org
- Axure
  - Download from: http://www.axure.com

# Session 1: Introduction

- Embedded System and Software
- Software Development Lifecycle
- Software Design
- Consumer Electronics Software Design
- Conclusions

# Embedded System and Software

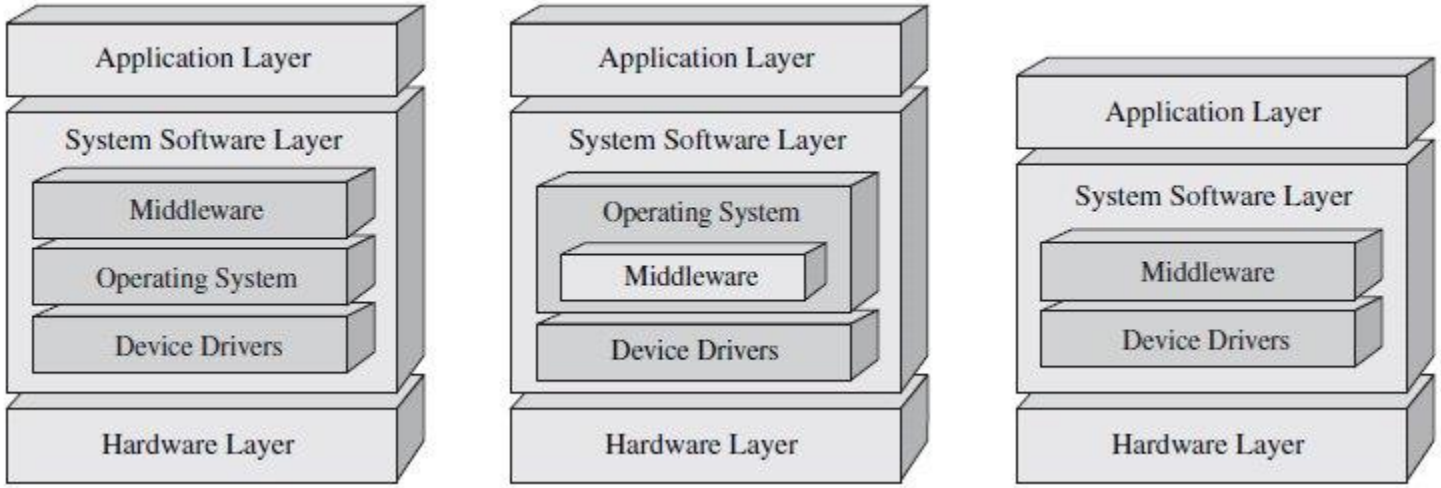
- Embedded system
  - An embedded system is nearly any computing system other than a desktop, laptop, or mainframe computer
  - Embedded systems can be defined as information processing systems embedded into enclosing products
  - Common characteristics
    - Single-functioned
      - usually execute only one program repeatedly
    - Tightly constrained
      - cost, size, performance, power
    - Reactive and real time continually react to
      - changes in the system’s environment, and must compute certain results in real time without delay

# Embedded System and Software

- Application areas where embedded system used
  - Transportation
    - Automotive, aircraft and trains electronics
      - anti-braking systems, GPS-systems, ...
  - Telecommunication
    - telephones, radio frequency equipments, ...
  - Consumer electronics
    - TV sets, refrigerators, smart phones, ...
  - Medical systems
  - Military sytems
  - Payment systems
  - .....

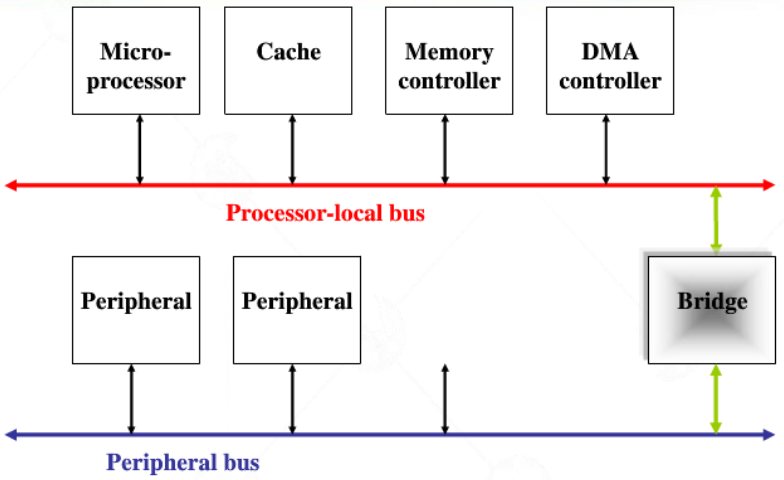
# Embedded System and Software

- Embedded system model



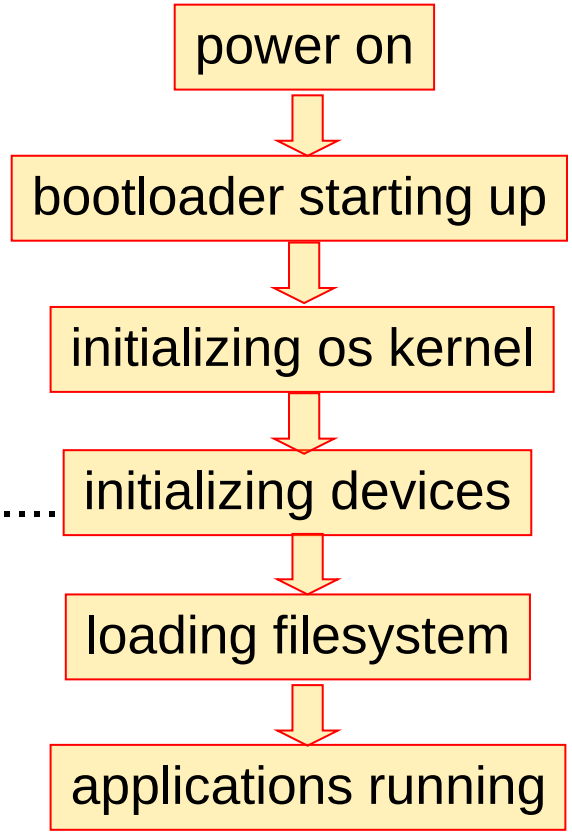
# Embedded System and Software

- Hardware layer
  - Processors
    - microprocessor, μP
    - microcontroller, μC
    - digital signal processor, DSP
    - graphics processing unit, GPU
  - Memories
    - Cache, RAM, ROM, Flash memory
  - I/O devices
    - LED, keyboard, camera, touch screen, .....
  - Buses
    - data bus, control bus, address bus

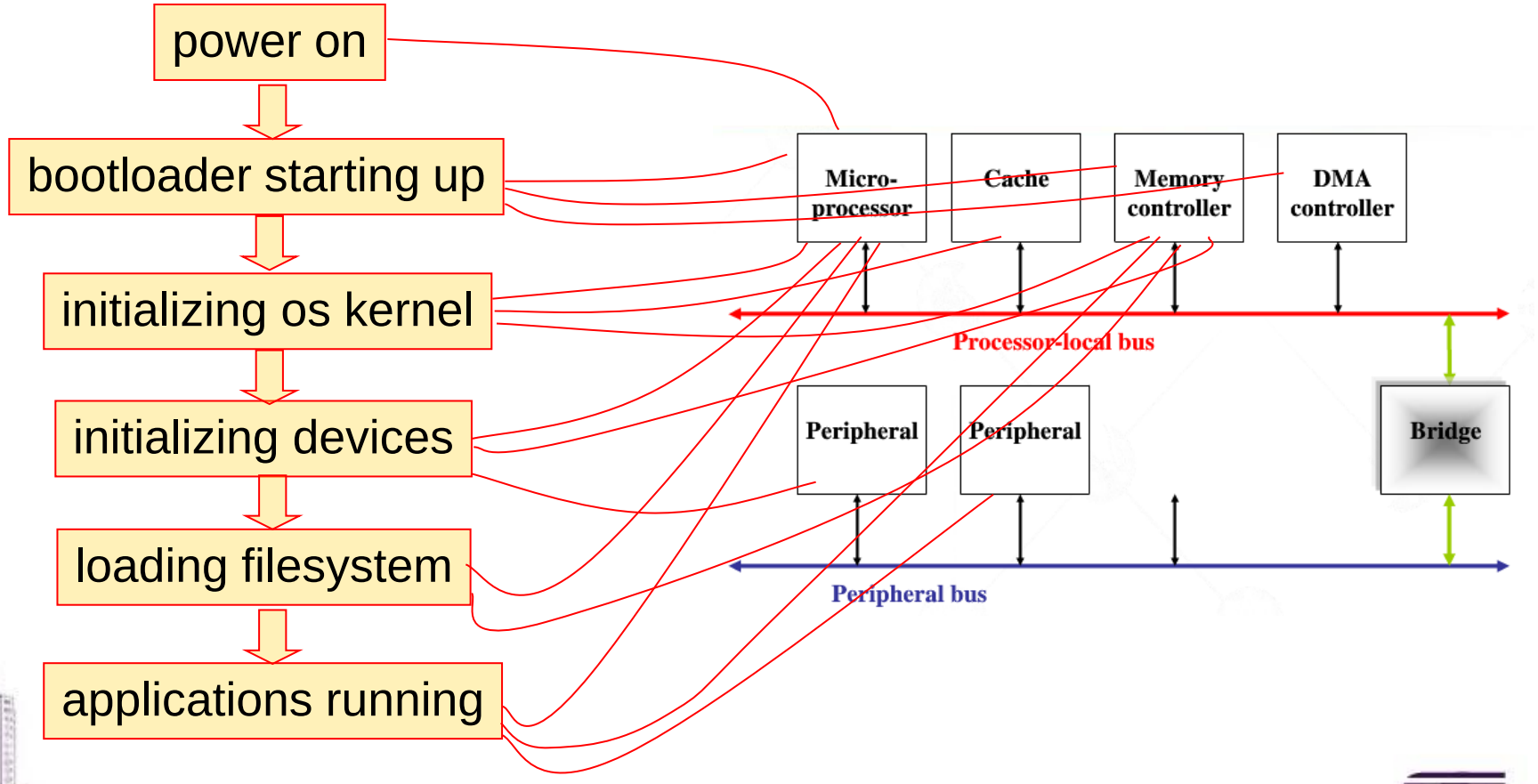


# Embedded System and Software

- Software layer
  - Applications
    - browser, email, gallery, ......
  - System softwares
    - middleware: CORBA, Java ME, ......
    - operating system: Android, μC/OS, Linux, ......
    - drivers: network driver, camera driver, ......
    - bootloader: U-boot, vivi, Redboot, ......
  - Support softwares
    - database: SQLite, PostgreSQL, ......
    - programming tools: IDE, Debuggers, compliler, ......

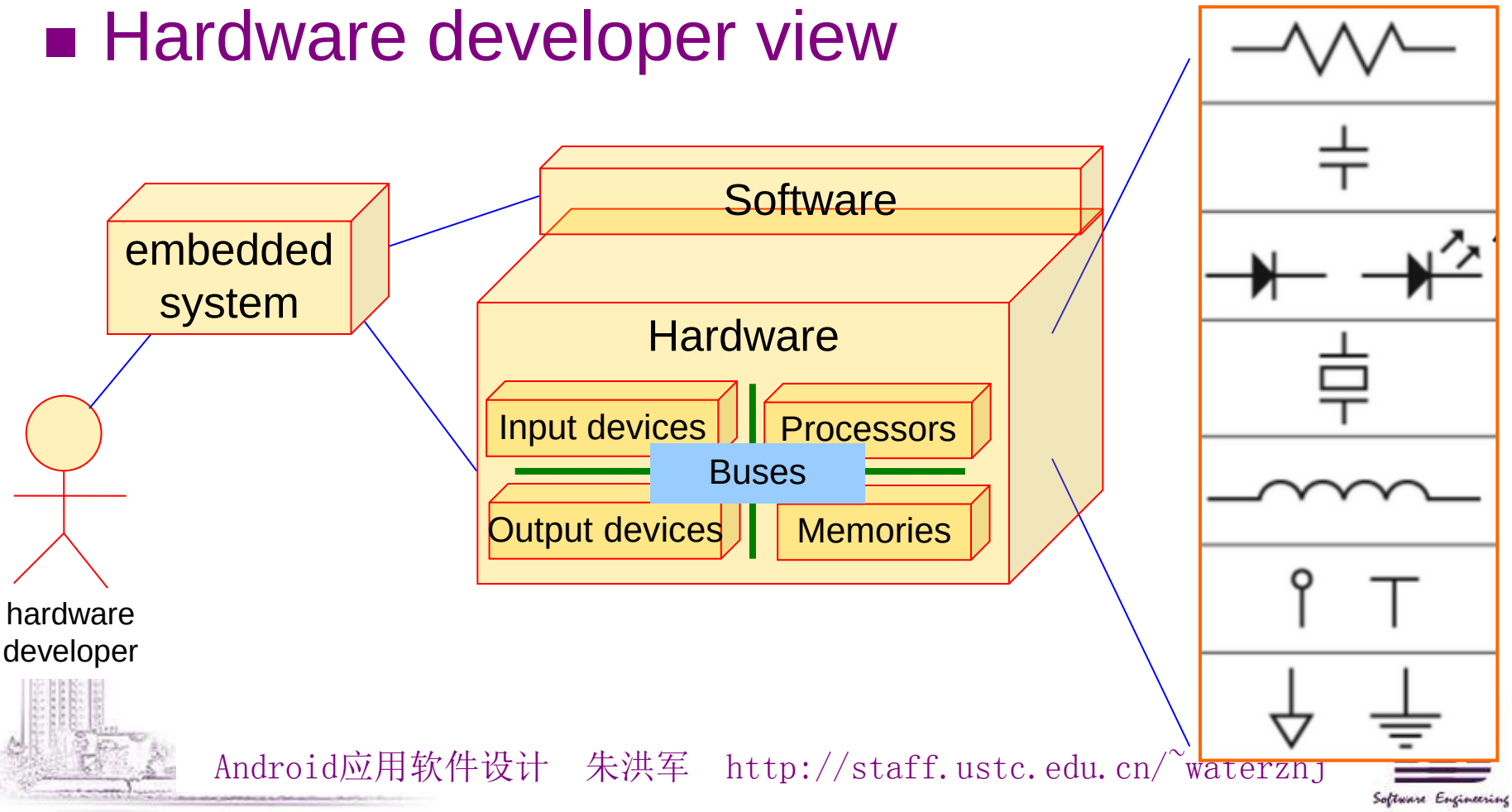


# Embedded System and Software



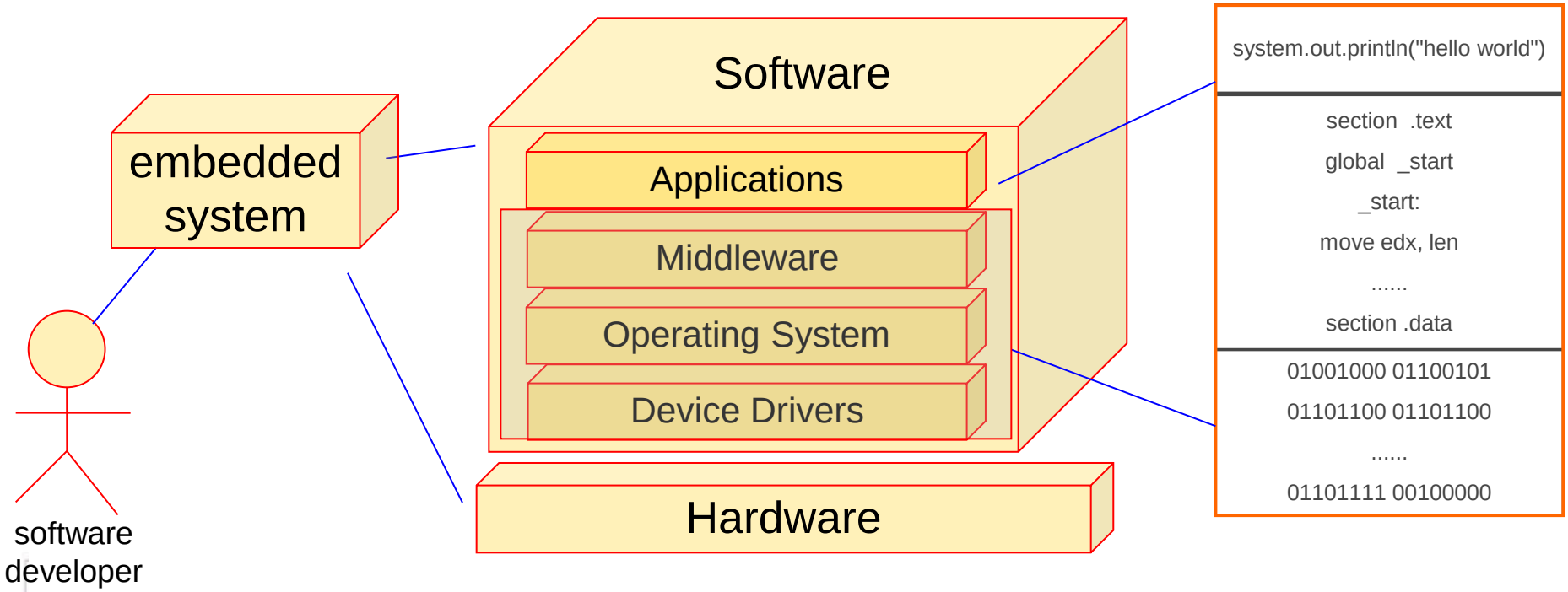
# Embedded System and Software

- Hardware developer view



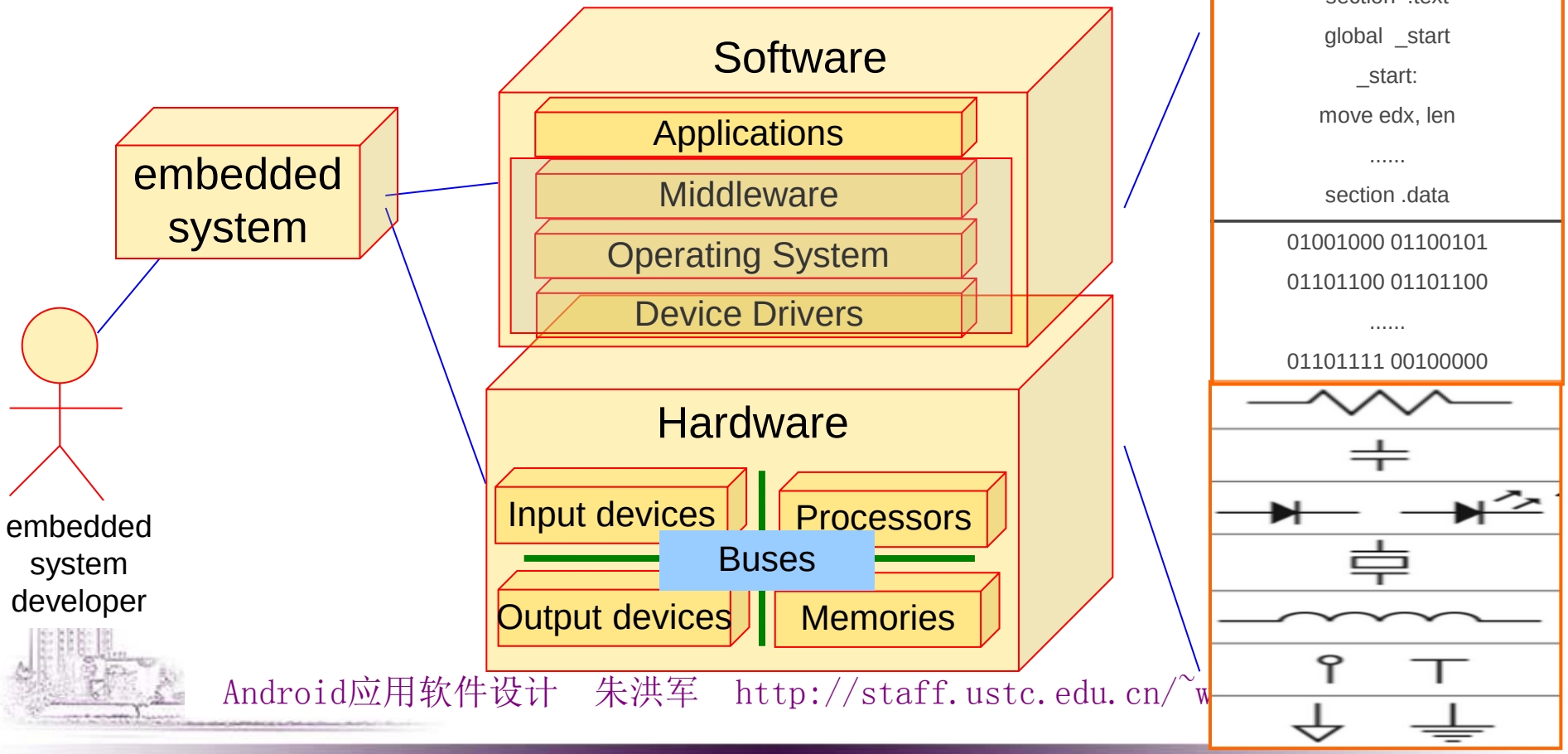
# Embedded System and Software

- Software developer view



# Embedded System and Software

- Embedded system developer view



# Embedded System and Software

- So, embedded system developer
  - in the field of software, knows more hardware skills and knowledges than software developer
  - in the field of hardware, knows more software skills and knowledges than hardware developer

# Embedded System and Software

- Embedded system trends
  - Low power consumption, low cost, small size
  - Multicore processors
  - Internet of things
  - Multi-function
  - More powerful development tools
  - UI becomes more friendly
  - Embedded systems development has become mainly software-driven

# Embedded System and Software

- How to develop an embedded system
  - Hardware development
    - Analysis, design, producing, test, deploy
  - Software development
    - Analysis, design, programming, test, deploy
  - Integration
    - Integrating, test, deploy

# Embedded System and Software

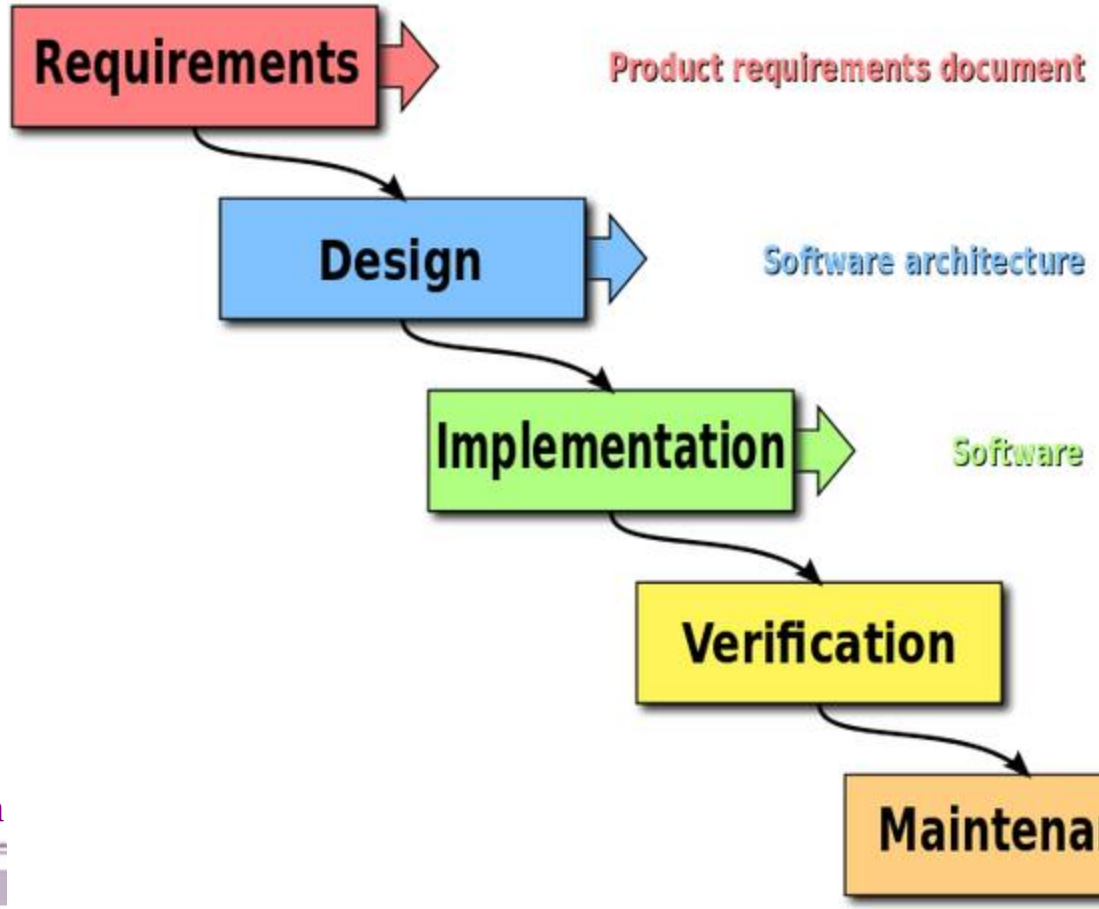
- This course cares more about
  - software
    - application layer software
  - programming
    - consumer eclectronic applications design and programming

# Software Development Lifecycle

- Software process
  - is a series of phases of activities performed to construct a software system
  - each phase produces some artifacts which are the input to other phases
  - each phase has a set of entrance criteria and a set of exit criteria
  - process models
    - waterfall model, prototype model, spiral model, unified process model, agile method model, etc.

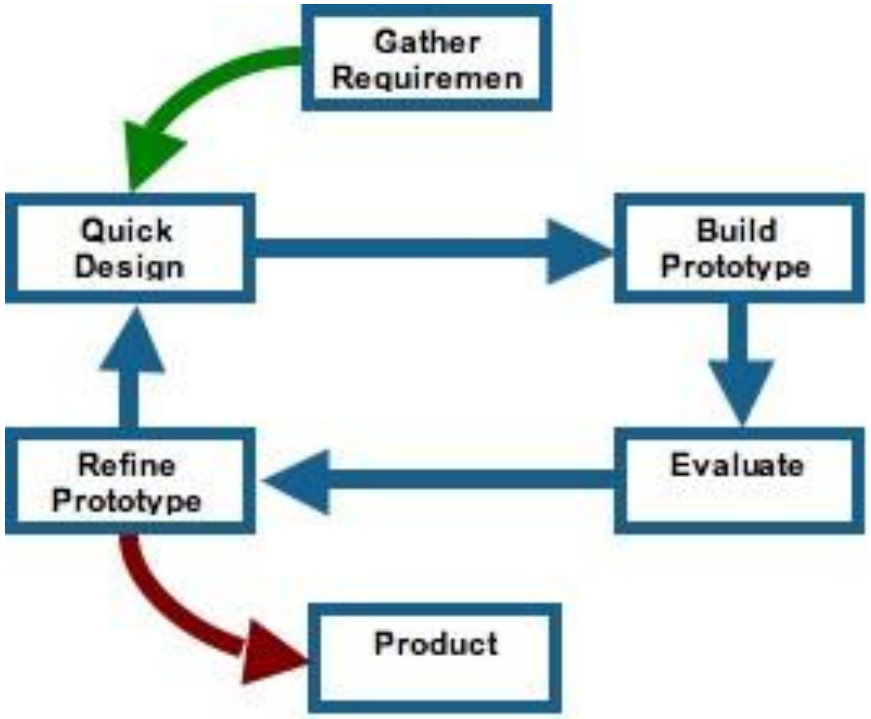
# Software Development Lifecycle

- Waterfall process model



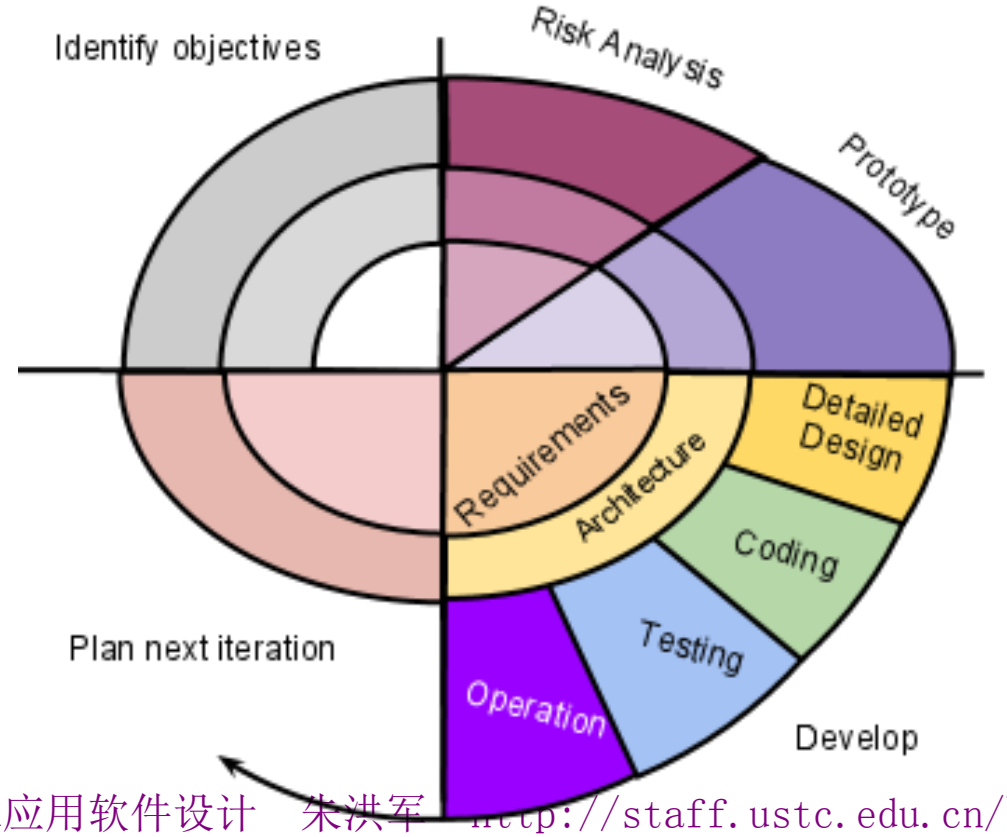
# Software Development Lifecycle

- Prototype model



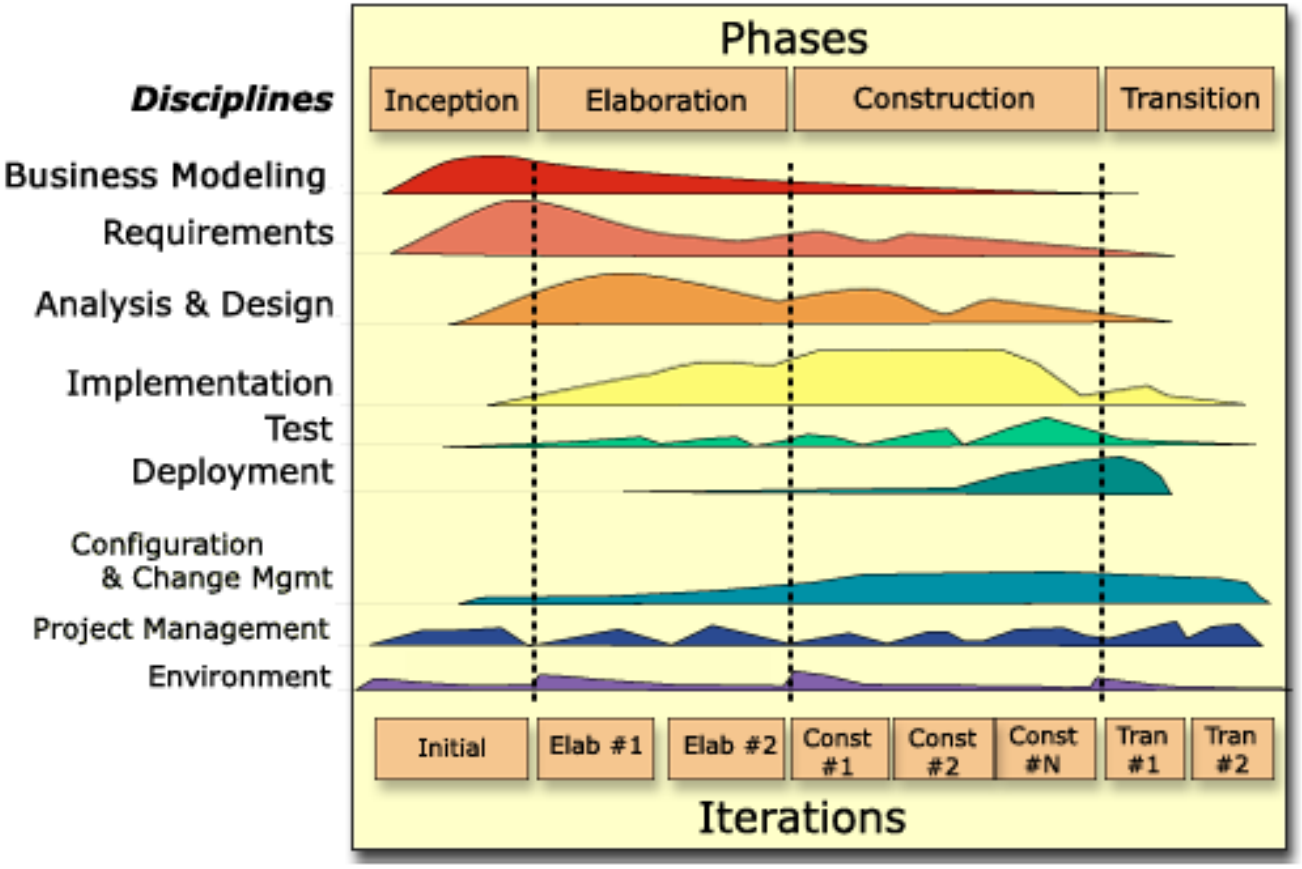
# Software Development Lifecycle

- Spiral model



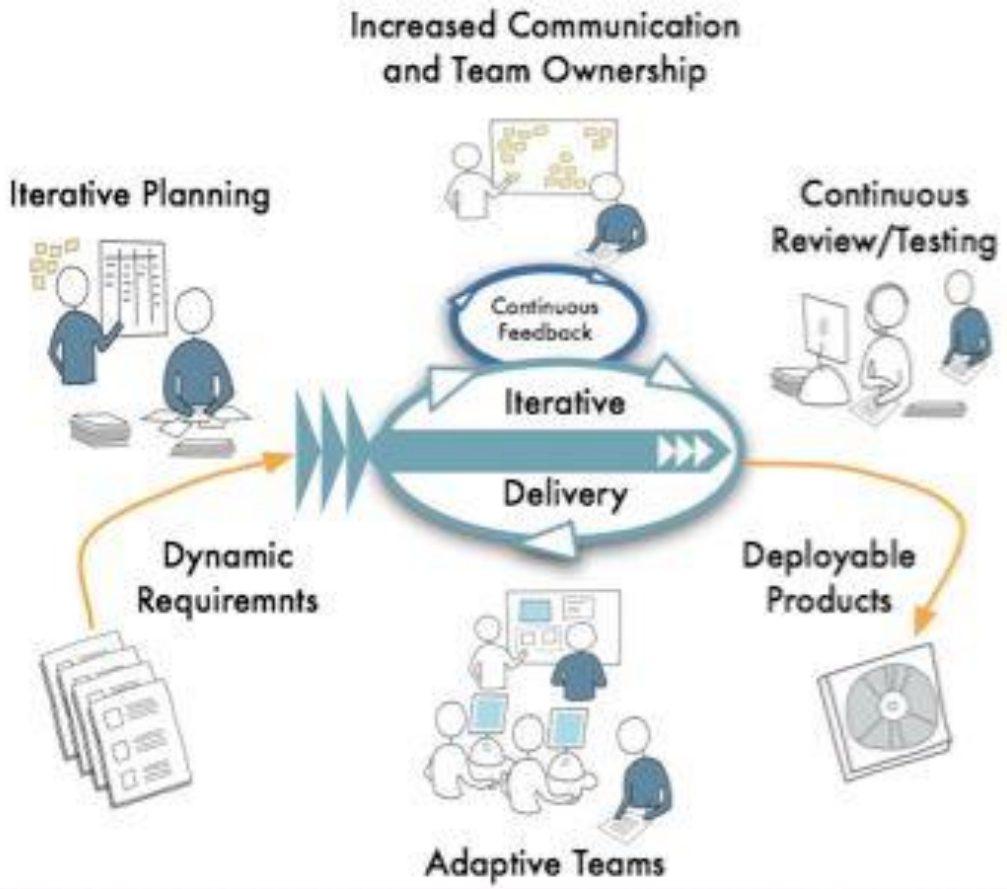
# Software Development Lifecycle

- Unified process model



# Software Development Lifecycle

- Agile method model




Andro

/~waterzhj

Software Engineering

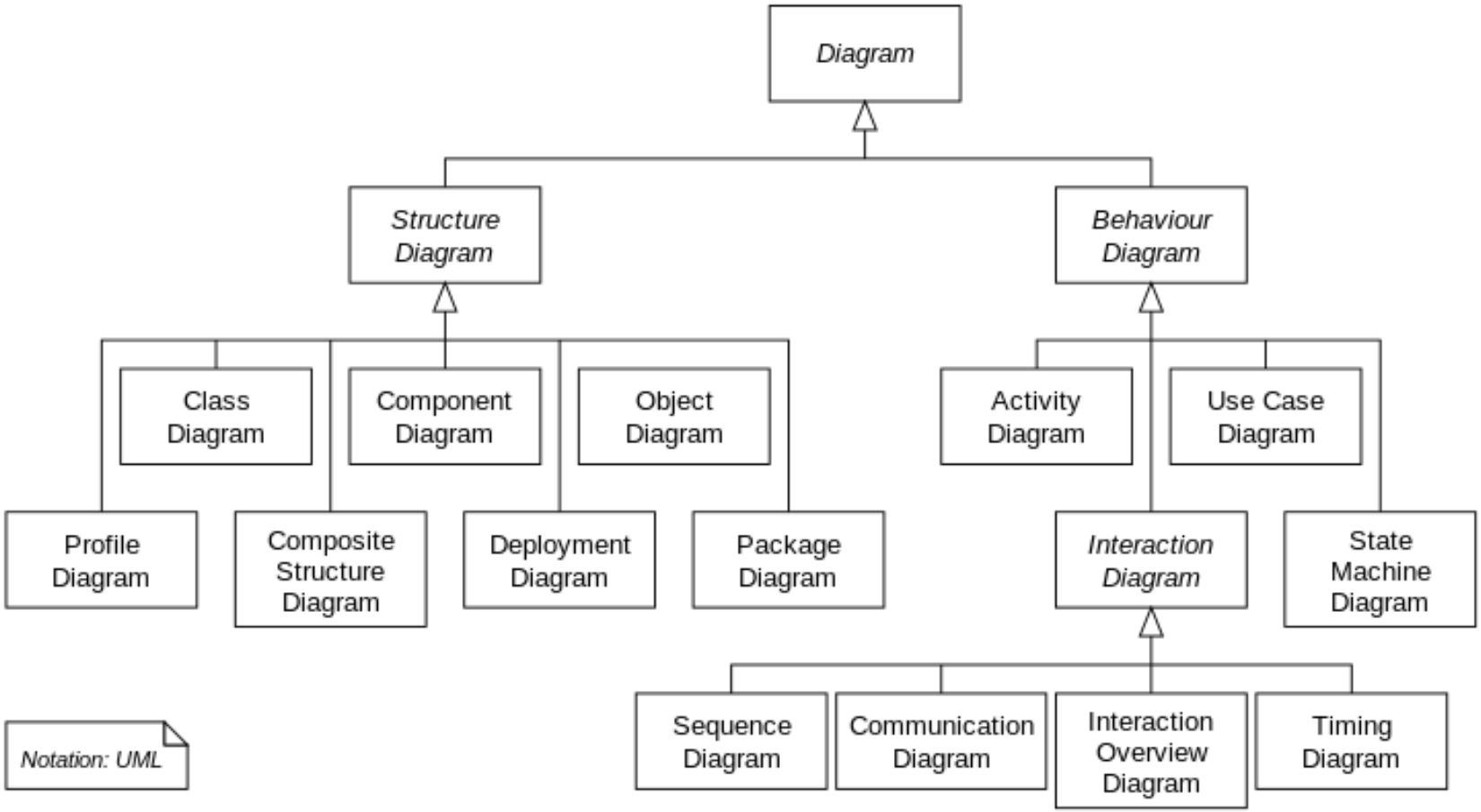
# Software Design

- Design phases
  - Architecture design
    - system type, subsystems, modules, public data structures, security frameworks, etc.
  - Database design
    - database mode, data storage, triggers, stored processes, etc.
  - Interface design
    - input, process, output, gui, communication, etc.
  - Detailed design
    - data structures, algorithm, procedure, etc.

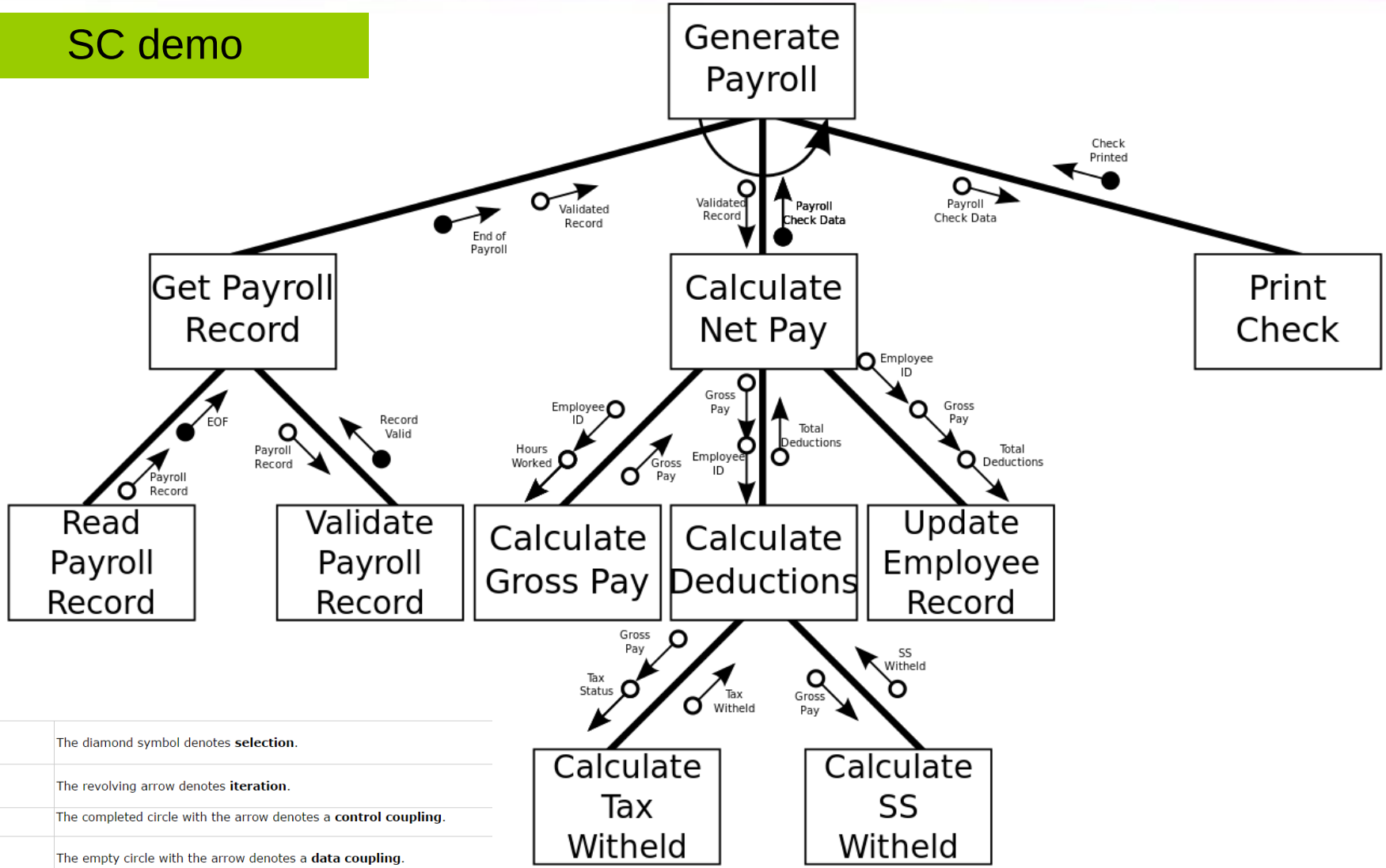
# Software Design

- Design methods
  - Structured design method
  - Object-oriented design method
- Modeling languages
  - Unified modeling language, UML
  - Structure chart, SC
  - ……

# UML Diagram Hiararchy



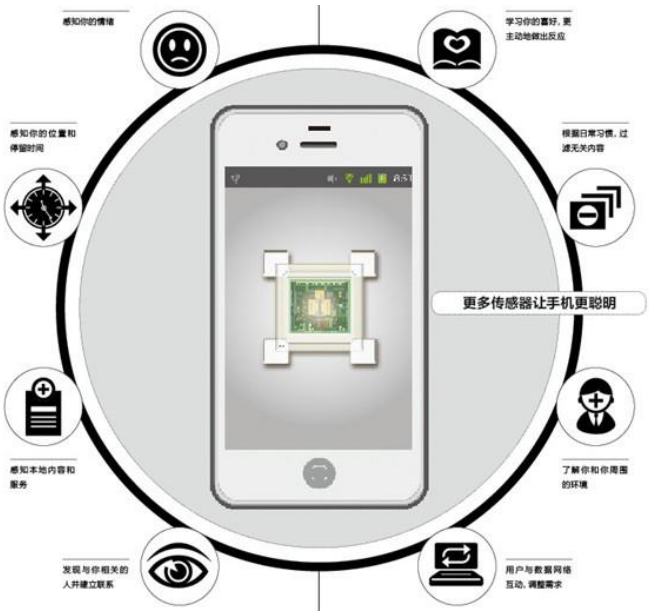
# SC demo




| | |
|---|---|
| ◇ | The diamond symbol denotes **selection**. |
| ↺ | The revolving arrow denotes **iteration**. |
| ●→ | The completed circle with the arrow denotes a **control coupling**. |
| ○→ | The empty circle with the arrow denotes a **data coupling**. |

# Consumer Electronics Software Design

- Software characters
  - Resources constrained
    - screen size, cpu, memory, ......
  - Security demanded highly
    - privacy protection, payments, ......
  - Mobility
    - location is not fixed
  - Interaction ways
    - touchscreen, virtual keyboard, ......
  - Diversity
    - sensors, domains applied, networks, ......



# Consumer Electronics Software Design

- Consumer electronics software design cares more about
  - Simplicity
  - User interface
  - Features
  - Security
  - Cost
  - Performance

# Consumer Electronics Software Design

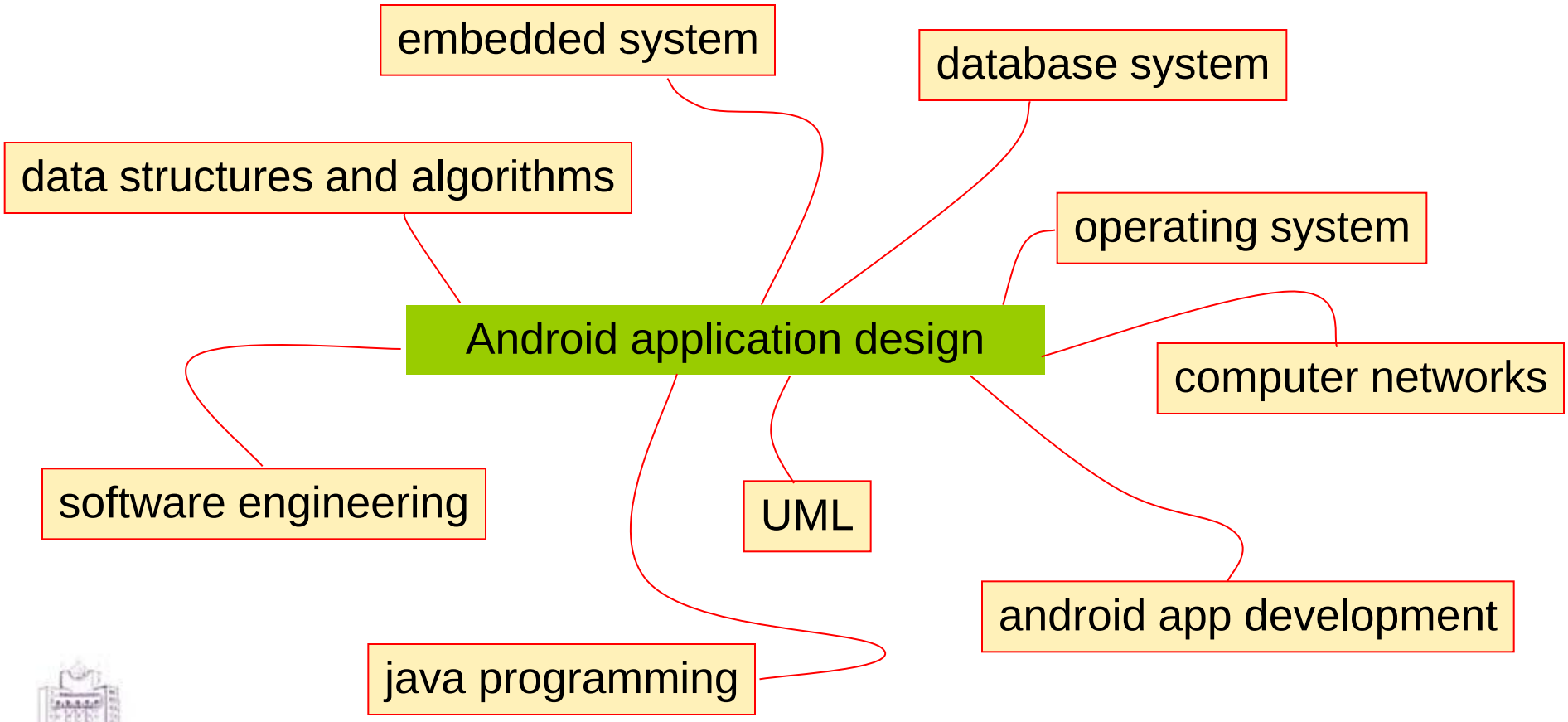
- Design objects
  - Software architecture
  - User Interface
  - Data structure and storage
  - Network communication
  - Security
  - Performance

# Consumer Electronics Software Design

- This course covers
  - Architecture design—session 2
  - Prototype design—session 3
  - Database design—session 4
  - Multi-thread programming—session 5
  - Network programming—session 5
  - File programming—session 5
  - Unit testing—session6

- Design objects
  - Software architecture
  - User Interface
  - Data structure and storage
  - Network communication
  - Security
  - Performance

# Conclusions



# Conclusions

- Embedded System and Software
- Software Development Lifecycle
- Software Design
- Consumer Electronics Software Design