Software Process Modeling

Software Implementation

Session Outcomes

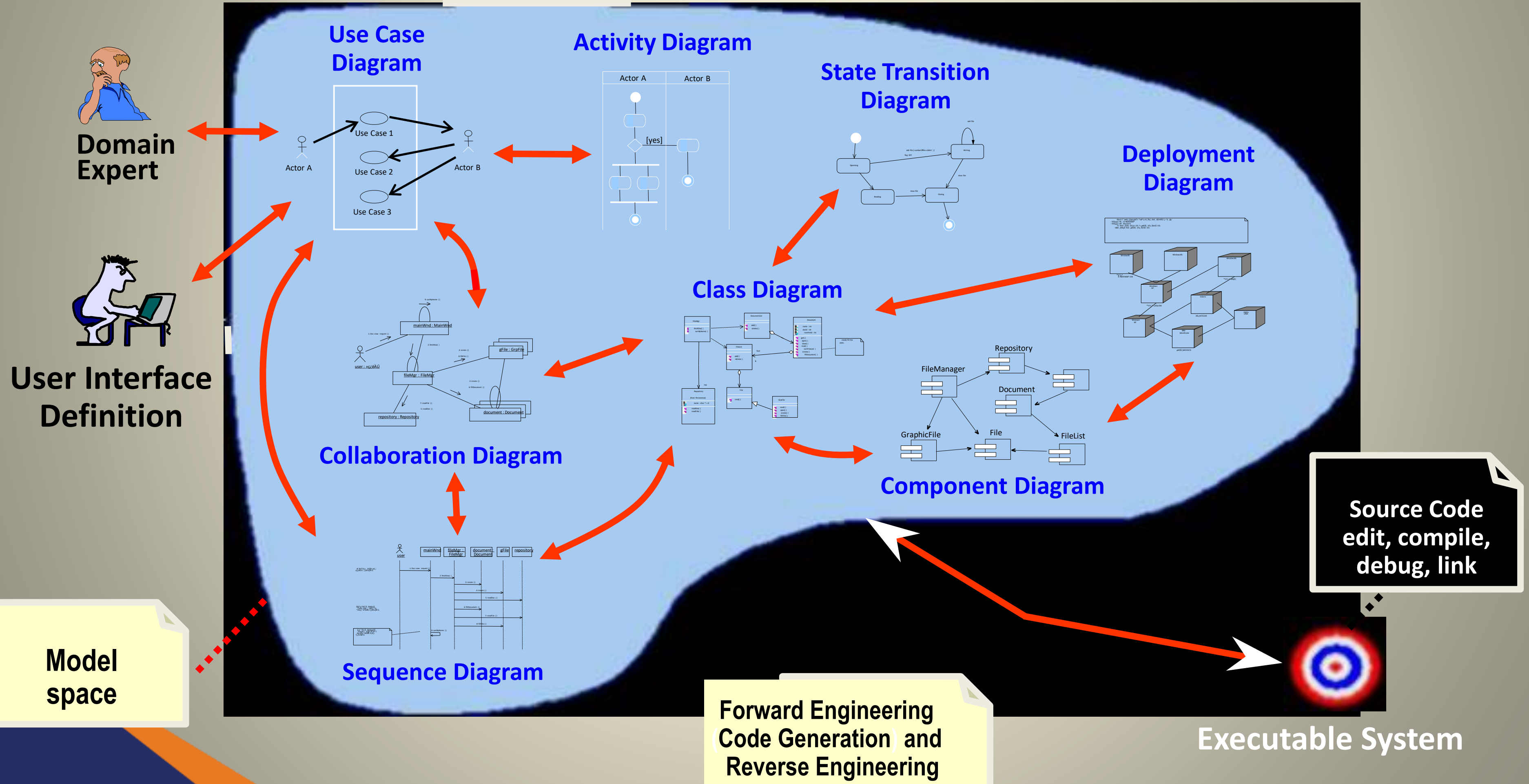
- Implementation
  - Design to Implementation
  - Round Trip Engineering
  - Implementation types
  - Coding standards

Design to Implementation

- Implementation is the process of realizing the design as a program.
- During the Design Phase, you learnt to build the design models which are independent of the programming language.
- Design Models aims to
  - Visualize
  - Specify
  - **Construct**
  - Document

Ref: Software Engineering, I. Sommerville, 10th Edition

Design as a template for Implementation



Domain Expert

User Interface Definition

Model space

Use Case Diagram

Activity Diagram

State Transition Diagram

Deployment Diagram

Class Diagram

Collaboration Diagram

Component Diagram

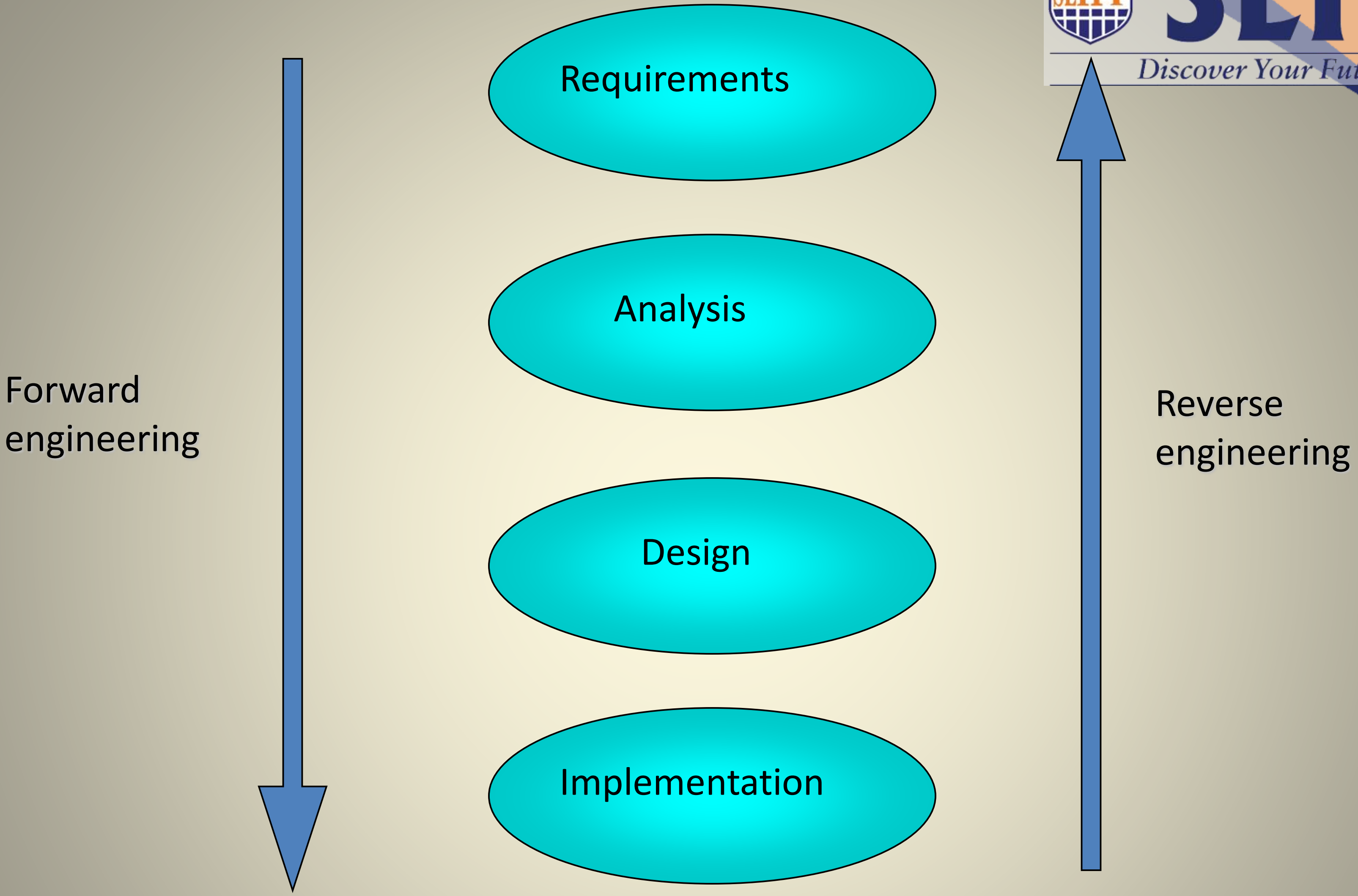
Sequence Diagram

Source Code edit, compile, debug, link

Executable System

Forward Engineering (Code Generation) and Reverse Engineering

Ref: Fundamentals of Visual Modeling with UML



Requirements

Analysis

Design

Implementation

Forward engineering

Reverse engineering

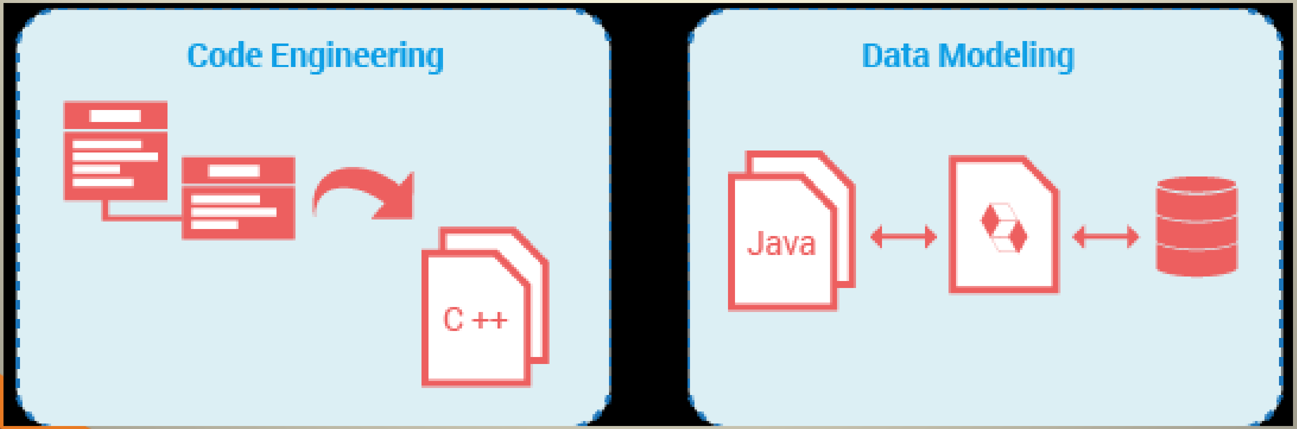
Reverse and Forward Software engineering

Round-trip engineering (RTE)

- Forward and Reverse engineering combined
- Examples:
  - producing source code from class diagrams and class diagrams from source code.
  - translation from ER-model to relational model and back.
- You can use RTE to build a system in a given programming language.

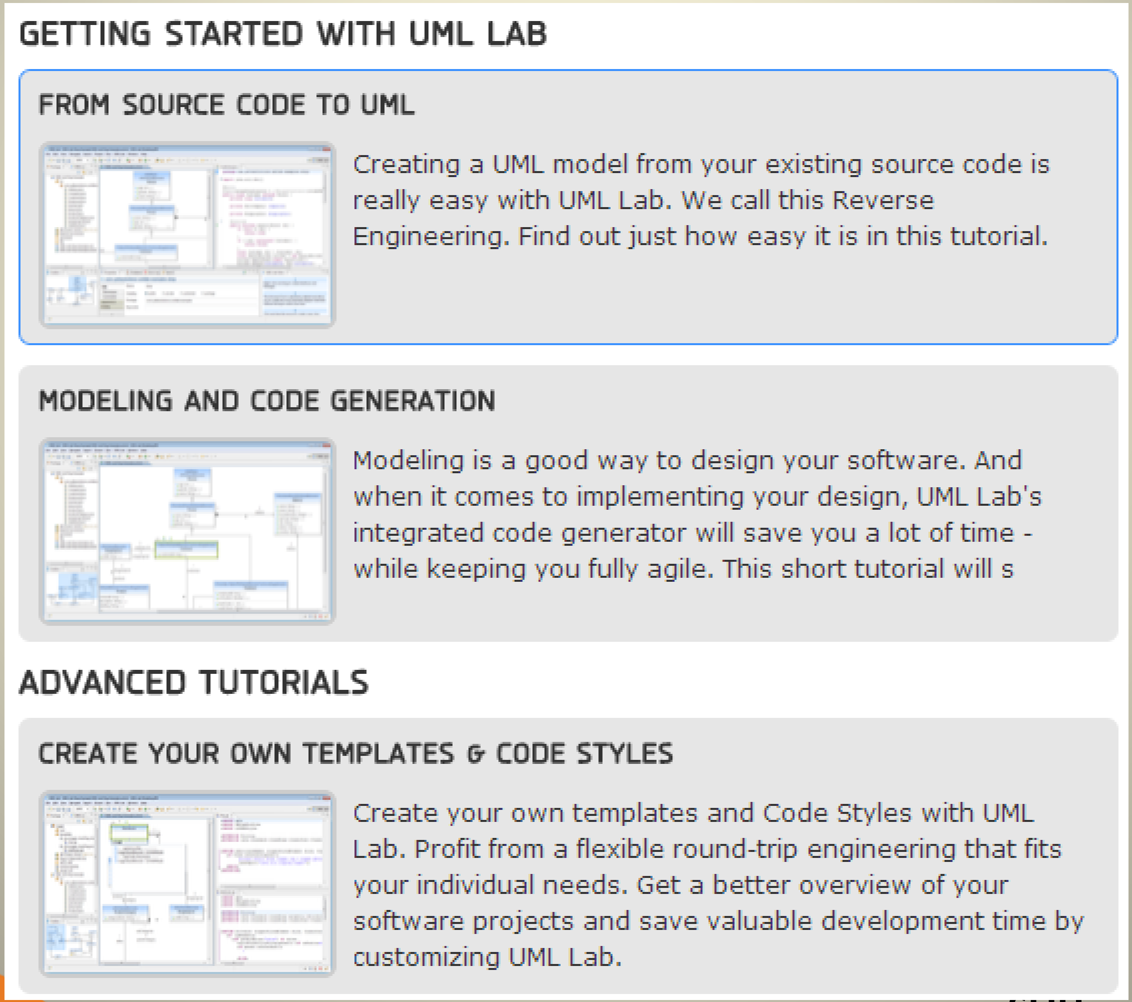
UML to Code, Code to UML

- https://www.visual-paradigm.com/features/code-engineering-tools/
  - – **Java Round-Trip Engineering**
    - Generate Java source code from UML class model
  - – **C++ Round-Trip Engineering**
    - Generate ANSI C++ source code from your UML class model



Model to Code

- http://www.uml-lab.com/en/uml-lab/features/roundtrip/



GETTING STARTED WITH UML LAB

FROM SOURCE CODE TO UML

Creating a UML model from your existing source code is really easy with UML Lab. We call this Reverse Engineering. Find out just how easy it is in this tutorial.

MODELING AND CODE GENERATION

Modeling is a good way to design your software. And when it comes to implementing your design, UML Lab's integrated code generator will save you a lot of time - while keeping you fully agile. This short tutorial will s

ADVANCED TUTORIALS

CREATE YOUR OWN TEMPLATES & CODE STYLES

Create your own templates and Code Styles with UML Lab. Profit from a flexible round-trip engineering that fits your individual needs. Get a better overview of your software projects and save valuable development time by customizing UML Lab.

Implementation types

1. **Build**
   - The previous slides explained how you could build your own system
   - You could even get a third party to develop the system / part of system for you.
2. **Buy**
   - In a wide range of domains, it is now possible to buy COMMERCIAL OFF-THE-SHELF systems (COTS) that can be adapted and tailored to the users' requirements.
   - For example, if you want to implement a medical records system, you can buy a package that is already used in hospitals. It can be cheaper and faster to use this approach rather than developing a system in a conventional programming language

COTS

- Advantages
  - Cheaper
  - Shortens design-to-production cycles
  - General Purpose (more flexible for different applications)
- Disadvantages
  - May not be suitable for all applications
  - May not meet reliability requirements of mission critical systems (flight control, weapons direction, medical equipment)

Implementations types

3. Open source development

- Project/Community open source
  - developed and managed by a distributed community of developers
  - volunteers are invited to participate in the development process.
- Commercial Open Source Software/ COSS
  - controlled by a single entity
  - The owner only accepts code contributions if the contributor transfers copyright of the code to this entity.
  - They may distribute their software for free or a fee.

Activity

- Give few examples of open source products you know and use.

Activity Answer

- Project/Community
  - The best-known open source product is, of course, the Linux operating system which is widely used as a server system and, increasingly, as a desktop environment.
  - Other important open source products are Java, the Apache web server and the mySQL database management system (mySQL has a commercial license as well).
- COSS
  - Red Hat
  - Facebook is the largest COSS code contributor

Open Source

- Advantages
  - Free to Try
  - Free Support
  - Fewer Bugs and Faster Fixes
- Disadvantages
  - Free support is not always the fastest support
  - Not a favorite for unskilled users

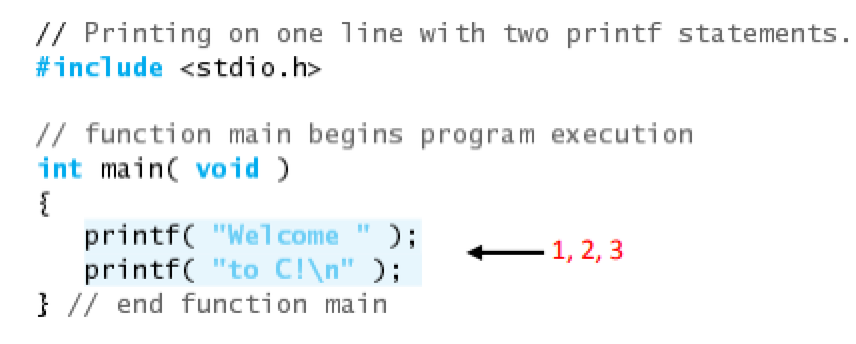
Build

- When building your own system
  - Confirm the detailed design
  - Understand required standards. For example, **Coding Standards**
  - Implement Code
    - Plan the structure based on the SDD (Software Detailed Design)
    - Code - > apply standards -> self inspect -> compile -> unit test
  - Try to reuse as much as possible

Coding Standards

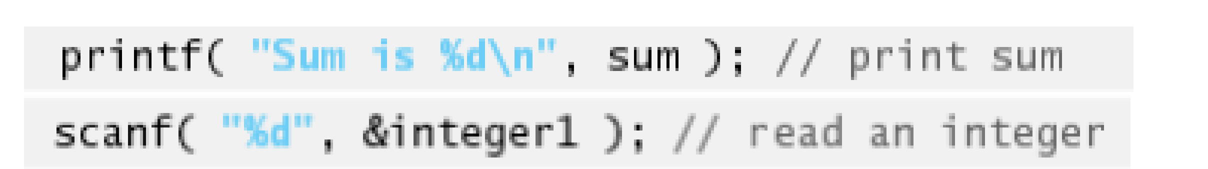
Coding Standards

- You have already used Coding Standards in IP.
  - Indentation
  - Commenting Code
  - Whitespace
  - Naming Variables



```
// Printing on one line with two printf statements.
#include <stdio.h>

// function main begins program execution
int main( void )
{
   printf( "Welcome " );
   printf( "to C!\n" );
} // end function main
```



```
printf( "Sum is %d\n", sum ); // print sum
scanf( "%d", &integer1 ); // read an integer
```

Activity

Code A

```
if (g < 17 && h < 22 || i < 60) {
return true; }
else {
System.out.println (“incorrect”) ;
 return false; }
```

Code B

```
if (g < 17 && h < 22 ||  i < 60)
{
                return true;
}
 else
{
                System.out.println(“incorrect”);

                return false;
}
```

What code is following indentation ?

Activity Answer

Code A

```
if (g < 17 && h < 22 || i < 60) {
return true; }
else {
System.out.println (“incorrect”) ;
 return false; }
```

Code B

```
if (g < 17 && h < 22 ||  i < 60)
{
            return true;
}
 else
{
            System.out.println(“incorrect”);

            return false;
}
```

What code is following indentation?

Code B

Activity

Code A

```
for(int i=0;i<40;i++)
            {system.out.println(i);}
```

Rewrite the above code according to the coding standards you learnt.

Activity Answer

Code A

```
for(int i=0;i<40;i++)
            {system.out.println(i);}
```

Code Answer

```
     for( int i = 0 ; i < 40 ; i++ ){
           system.out.println(i);
      }
```

References

- Software Engineering – 10th Edition by Ian Sommerville, Chapter 7
- https://www.visual-paradigm.com/features/code-engineering-tools/
- http://www.uml-lab.com/en/uml-lab/features/roundtrip/
- Courseweb Documents
  - Coding Standard Document
  - open-source-vs-proprietary-software-pros-and-cons