

COEN 6311 Fall 2023 Project Description

This course project follows the main aspect of a software process, namely 1) software specification, 2) software development, 3) software validation and 4) software evolution. The 4 deliverables are structured according to these four aspects.

1. Problem Definition

The main *problem* for this project to solve is to build a multiuser software system to share information between users and intelligent tools that aim to help user complete their work tasks more efficiently. Such a system is called Information Capture and Dissemination Environment (ICDE) shown in Figure 1.

2. The ICDE System

This ICDE case, associated context description and the figures are from Book Essential Software Architecture, 2nd Edition by Ian Gorton. ICDE is part of a suite of software systems for providing intelligent assistance to professionals such as financial analysts, scientific researchers, intelligence analysts etc. To this end, ICDE automatically captures and stores data that records a range of actions performed by a user when operating a workstation. For example, when a user performs a Google search, the ICDE system will transparently store in a database:

- The search query string
- Copies of the web pages returned by Google that the user displays in their browser

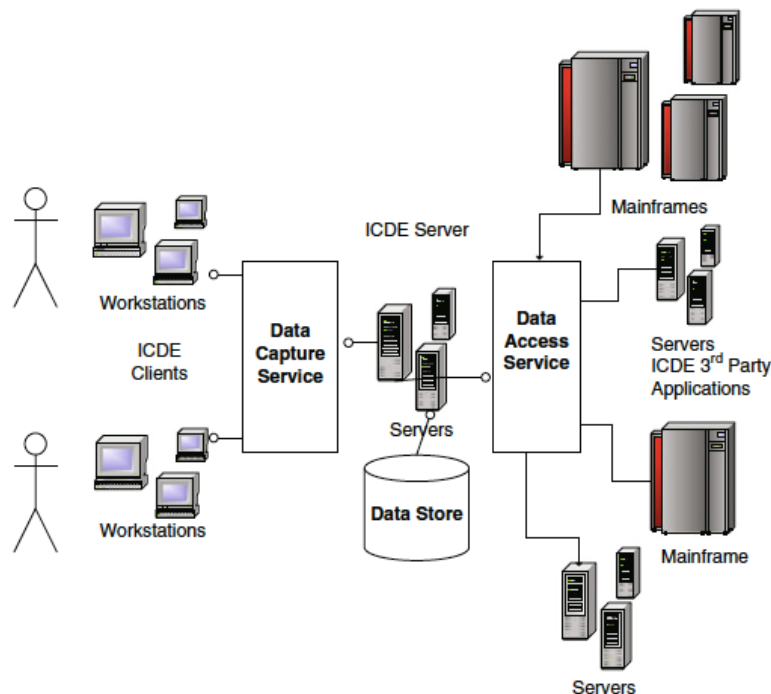


Figure 1 ICDE Reference Context

This data can be subsequently retrieved from the ICDE database and used by third-party software tools that attempt to offer intelligent help to the user. These tools might interpret a sequence of user inputs, and try to find additional information to help the user with their current task. Other tools may crawl the links in the returned search results that the user does not click on, attempting to find potentially useful details that the user overlooks.

A use case diagram for the ICDE system is shown in Figure 2. The three major use cases incorporate the capture of user actions, the querying of data from the data store, and the interaction of the third party tools with the user.

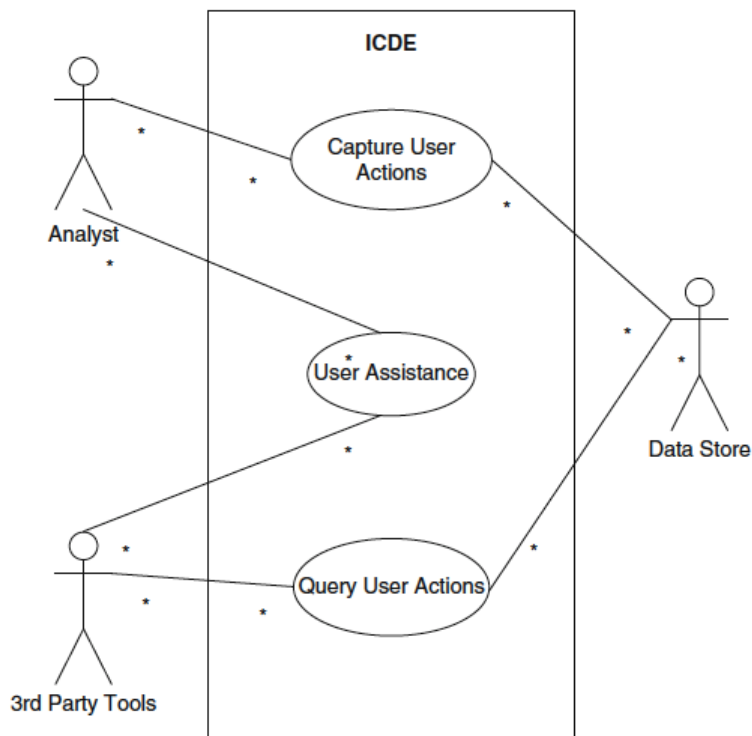


Figure 2 Use Cases of ICDE

The initial product of ICDE v1.0 targets a small user trial involving a few users, aiming for successful function development and deployment with the core features of the data collection, data storage and data query. The ICDE v2.0 further provides support for third party tools to leverage ICDE to build applications.

3. Project Demand

The project is to follow the software process to engineer the ICDE system for a specific domain application. Each team should build the core ICDE and define an application for ICDE to solve. Each application should leverage the ICDE's core function of data collection, data storage and data query.

3. Deliverables and Schedule.

The report or written documents required for deliverables should follow the template from <https://www.ieee.org/conferences/publishing/templates.html>. Mis-formatted documents will result in not being marked.

Deliverable 0: Team Registration and Engineering Setup

Due Date: Week 2 Mon (September 18) 23:55pm Moodle Site

Deliverable Format: Registration and Engineering Setup Report

Content: 1) Team member's details (name, sid, program and email, SE background).

Minimal 3 members, Maximum 4 members; 2) Tools decided on team development including public / private code repository for version control (for code, data and design document), IDE, programming language, group communication software, progress tracking tools.

Deliverable 0: is not marked but missing the submission or late submission will result in mark deduction portion to the total project marks.

Deliverable 1: Software Specification – **8 Marks**

Due Date: Week 5 Monday (October 16) 23:55pm Moodle Site

Select a process model to provide the following items:

1. The Problem Statement - Describe the general problem your ICDE-[AppName] aims to solve
2. Concrete context (architecture) of your ICDE-[AppName], referring to Figure 1;
3. Project Goal (What your ICDE-[AppName] – What's the key functionalities of your ICDE-[AppName]; Make concrete use cases of your [ICDE-AppName], referring to Figure 3;
4. Describe "user stories" for each use case;
5. Write requirement specification for each user stories
6. Further divide each user requirement into sub requirements and provide sensible ordering of requirements

Deliverable 2: System Model and Design – **7 Marks**

Due Date: Week 7 Monday (October 30) 23:55pm Moodle

Site Part I:

Following the Agile process:

- Setup the team – Set the role of [Scrum: product owner] and [Scrum: Master]
- Associate user stories to sub-requirements
- Define tasks under each requirement (can be in micro level). Order the task in priority (based on timeline urgency, time availability of each member weekly, skill sets and capability of the team)
- Following the Scrum sprint cycle to complete tasks (you make decision on the number of tasks/portion of tasks to complete according to the timeline)

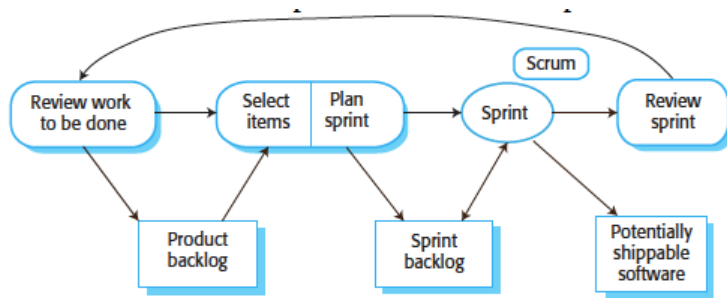


Figure 3 Scrum sprint cycle (picture from the textbook)

After you have already gone through a full scrum cycle. Now review your task by further completing its modeling and design.

1. Revise the context/external description of your application – with use case diagrams and architecture diagrams. The use case diagrams should follow two levels – the overall level and the level of user requirements/system requirements. The architecture diagram is differentiated into high level architectures and system level architectures.
2. Revise the documentation of the task already implemented with modeling and design.
 - Present and document the context of the task using *class diagram*
 - Present and document the structural context and data flow of this task using *activity diagram*
 - Present and document the interaction context of this task with other components and systems using *sequence diagram*
 - Present and document the behavior context of this task with response to events using *statement diagram*

Part 2

3. Select a new task that has not been started or completed, repeat requirement 2.
4. Further evaluate your progress and milestones to adjust your project plan, schedule or even budget (including HR hours).

Deliverable 3: Implementation, Deployment – 10 Marks

Due Date: Week 10 Monday (November 20) 23:55pm Moodle Site

Part 1: Reflection on the architectural designs that you have on Deliverable 1 and 2.

Please A) Identify and articulate what are your architectural designs and the associated software engineering process. B) Indicate if any revision to your architectural design is necessary. If the answer is Yes, please explain what revision, and the reason. C) Further adopt MVC and Layered Architecture Pattern to your ICDE-App. Describe your design decisions, and discuss the pros and cons of the design.

Part 2: Software metrics and granularity of components. The components' granularity is at the level of classes/objects, packages, libraries, frameworks and platforms. Please

make a statistical count of software metrics from all your tasks implemented so far and form a table given the template below. An example is given below.

Task Name	LOC	Component Granularity Level	Numbers of Unit
Save an user activity record	102	Java Class	4
	NA	Framework (MySQL)	1

Deliverable 4: Week 12 Monday (December 4), Final Class Time Demo/ Report – 5/5 Marks

Due Date:

Part 1. Group **Live Demo** at the last lecture time, online zoom session (Details are on Moodle). The application should be accessible through the Internet, NOT through localhost.

Part 2. **Final Project Report in Slides**, due Week 12 Monday (December 4) 23:59pm.

Part 1: The live demo of 5 minutes sharp for each group to demonstrate:

1. The problem the product [ICDE-App] to solve
2. Demonstrate top three functionalities of [ICDE-App]
3. Explain the software process that the [ICDE-App] has been produced.
4. Lessons Leant

Part 2: Final Project Report in Slides

Using the slides to document your final project to cover the following structure and slide template

Slide1 : The Product Problem

The problem the product [ICDE-App] to solve

Slide 2: Requirements

The left panel lists the top 3 user requirements; the right panel lists the top 3 system requirements

Slide 3: Architecture Design

Present the architecture design in a UML diagram and indicate which requirements are addressed at which parts (just indicating the requirement numbers from Slide 2)

Slide 4: One Major System Design

Present the most important system design in a UML diagram and indicate which requirements are at which parts (just indicating the requirement numbers from Slide 2)

Slide 5: Software Metrics

The complete software metrics as in the format of your D3 requirements

Slide 6: The Summary of Software Process

Present the software process and timeline of your whole product's stages

Slide 7: Tools used for Software Process

List the tools and the their purpose for facilitating your software process (NOT

tools that used for implementation, such as database or web server)

Slide 8: Lessons Leant

Slide 9: Group Member Role and Contribution Portion