Quantico CAD Project Summary

Charles Carter

July 20, 2020

CAD project summary The CAD project consists of five parts: project inception, database development, the software development process (*how* you develop an application), an application development phase, and your final project presentation. There are 18 steps. Each step consists of academic topics and deliverables. The steps, academic topics and deliverables, are summarized below.

Step	Description	Academic Topic	Deliverables
Project Inception			
The *software lifecycle* consists of these six stages: inception, elaboration, construction, transition, production , and retirement. As the first of the six phases, inception is about understanding the project scope and objectives and getting enough information to confirm that the project should proceed — or to convince you that it should not.			
1	Project Exploration	• Markdown • Version control • git	• Projects review paper • PowerShell Lab • Command Line Lab
2	Project Selection	• git • Github • course file structure	• Project selection paper • Github account README file • Github account `.gitignore` file • course file structure

Continued on next page

Table 1 – *Continued from previous page*

Step	Description	Academic Topic	Deliverables
3	Project Presentation I	• Revise project selection paper • Review Github accounts • UML - Activity diagrams	• Oral presentation, written paper, and (optional) PPT slide deck • Activity diagram
Database Development			
The database development cycle consists of five steps: (1) requirements gathering, (2) conceptual design, (3) logical design, (4) physical design, and (5) implementation. In the next three weeks, we will take a look at requirements & conceptual design, logical & physical design, and implementation.			
4	Requirements & Conceptual Design	• Entities • Attributes • Relationships • Weak entities • Multiplicity	• Entity-Relationship Diagram (ERD) as a PDF
5	Logical & Physical Design	• Normalization • Integrity constraints (entity, domain, referential) • Unique and nullability constraints • Default constraints • Data types	• Database Diagram (DBD) as a PDF

Continued on next page

Table 1 – *Continued from previous page*

Step	Description	Academic Topic	Deliverables
6	Implementation & Presentation II	• Structured programming • Top down development • Stepwise refinement • Iterative development • Incremental development • The spiral development cycle • *Program Development by Stepwise Refinement* - Niklaus Wirth	• SQL script consisting of the following: 1. script creating project database 2. Script inserting test data 3. Script running queries • PPT slide deck (optional)
Software Development Process			
The software development cycle can be seen as consisting of the repetition of five steps: (1) requirements gatherig, (2) requirements analysis, (3) application design, (4) implementation, and (5) testing. In the next five weeks, we will take a look at each step in turn.			
7	Requirements	• Software lifecycle (review) • Software design cycle (review) • Software development processes (waterfall and agile) • Requirements gathering	• A written use case is required • optionally you can add a use case diagram
8	Analysis	• Business Requirements Document (BRD) • Functional and non-functional requirements • IEEE Std 803-1998 • DOD-STD-2167, Secs 5.1–5.4, pages 19–33	• A Software Requirements Specification (SRS), specifically — a functional requirements specification

Continued on next page

Table 1 – *Continued from previous page*

Step	Description	Academic Topic	Deliverables
9	Design	• The design process • Introduction to UML • Review of selected UML design diagrams • IEEE Std 1016-2009	• UML diagrams[1] • Class diagram • Activity diagram
10	Implementation & Testing	• Importance of software quality assurance • Writing unit tests • Integration tests • Regression tests • User acceptance tests	A source code listing implementing the 1st use case
11	Project Presentation III	• Waterfall process • Scrum • Kanban • eXtreme Programming (XP) • Rational Unified Process (RUP)	PPT slide deck
Application Development			
This part of the CAD project consists of three iterations during which you will construct your final project. Each iteration will consist of a two week iterative cycle, in which you will complete requirements gathering, an analysis activity, software design, implementation, and testing.			
First Iteration			

Continued on next page

[1]There are four UML diagrams listed: class diagrams, sequence diagrams, activity diagrams, and state machine diagrams. The purpose is not to teach UML to the students, but to familiarize students with the purpose of program design in general using UML. Instructors may select different diagram types at their discretion.

Table 1 – *Continued from previous page*

Step	Description	Academic Topic	Deliverables
12	Single Responsibility	• Single Responsibility	Complete a use case for *one* requirement, a functional requirements specification, and an application design
13	Open-Closed	• Open-Closed	Source code listing implementing the use case
		Second Iteration	
14	Liskov Substitution	• Liskov Substitution	Complete a use case for *one* requirement, a functional requirements specification, and an application design
15	Interface Segregation	• Interface Segregation	Source code listing implementing the use case
		Third Iteration	
16	Dependency Inversion	• Dependency Inversion	Complete a use case for *one* requirement, a functional requirements specification, and an application design

Continued on next page

Table 1 – *Continued from previous page*

Step	Description	Academic Topic	Deliverables
17	Code Security	• Don't trust user input (particularly when working with SQL) • Use EntityFramework (or another ORM) rather than "raw" SQL • Minimize possible states (e.g., convert ints to enums) • Don't DIY encryption/password handling/etc. • Consider the confidentiality of your data (encryption at rest, encryption on the wire) • Use HTTPS via TLS (not SSL) • Keep software/operating systems up to date • Pay attention to security vulnerability publications (CVE and etc) • Always build systems assuming the attacker already has access • Use tested, validated libraries for things like getting input from users • Resist temptation to allow inputs of queries, commands, etc., which you pass to a parser and runtime system to do for you • Write intentional code that is not copy and pasted from an untrustworthy resource. • Select 3rd party libraries that have a large user base who have tested the software.	Source code listing implementing the use case

Continued on next page

Table 1 – *Continued from previous page*

Step	**Description**	**Academic Topic**	**Deliverables**
Project Completion and Evaluation			
18	Final Project Presentation	none	• All source code listings • Written paper • PPT slide deck (optional) • Oral presentation