DUY TAN University

Capstone Project Orientation

Nguyen Duc Man
Dean, International School
Duy Tân University

Contents

- Introduction
- IT Training Programs
- Software Development Capability
- Q&A

Introduction

- **The CMU Way**
- **Why Project Based Learning?**
- **State of Software Engineering Education and How its Changing.**
- **Professional Education Programs and Project Based Learning:**
  - Software Studio
  - Software Practicum
- **Goals of This Program**

Introduction

- There is a unique teaching philosophy at CMU which is embodied in our various software engineering programs.
- The key goal of this course is not to teach you what we do so you can copy our programs, but rather provide you with insight to help you as you design project based courses and program that best meet your needs.

Why Project Based Learning?

- **Remove the appendix.**
- **Knowing and Doing?**
- **There is a huge difference between knowing and doing!**
  - Knowing how to do something in theory is important in terms of creating first principles, defining vocabulary, common concepts, and so forth.
  - Doing is applying theory in practice.
- **Theory has limits:**
  - Without doing, there is no theory, there is no improvement in theory.
  - Theory may not work in less than perfect real world conditions

University Engineering Programs...

- **Most universities around the world focus on:**
  - software engineering and computer science theory
  - large numbers of individual assignments
  - limited team work
  - contrived problems and textbook examples
- **This is generally true for undergraduate as well as masters, and even PhD programs at many universities around the world**

But In The Real World… - 1

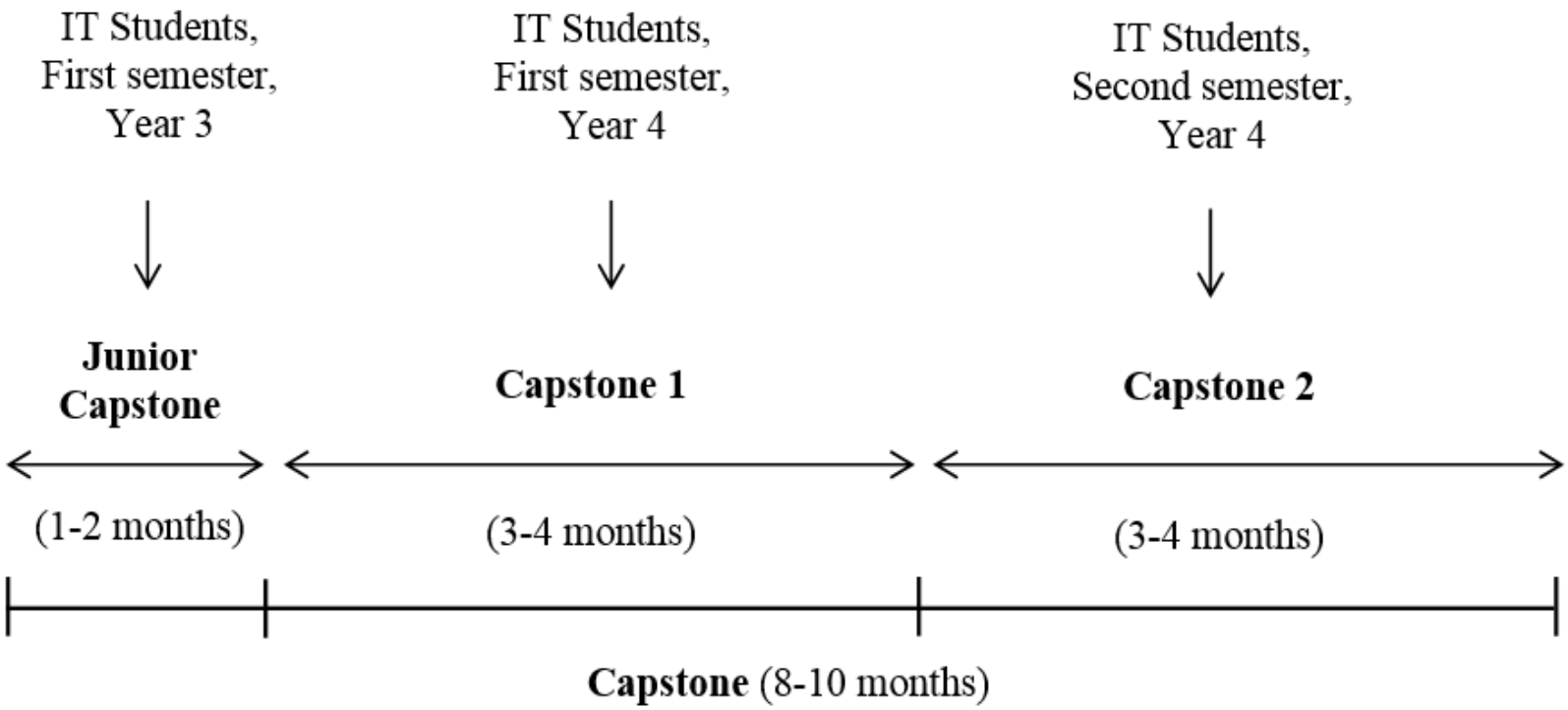
- **Engineers are faced with a very different situation in the real world:**
  - software engineering and computer science theory is often too academic for real problems
  - systems and applications are far larger than anything that they designed and built in university
  - products and systems are built in large teams
- **So in many cases the engineer is ill-equipped for the first day at work!**

But In The Real World… - 2

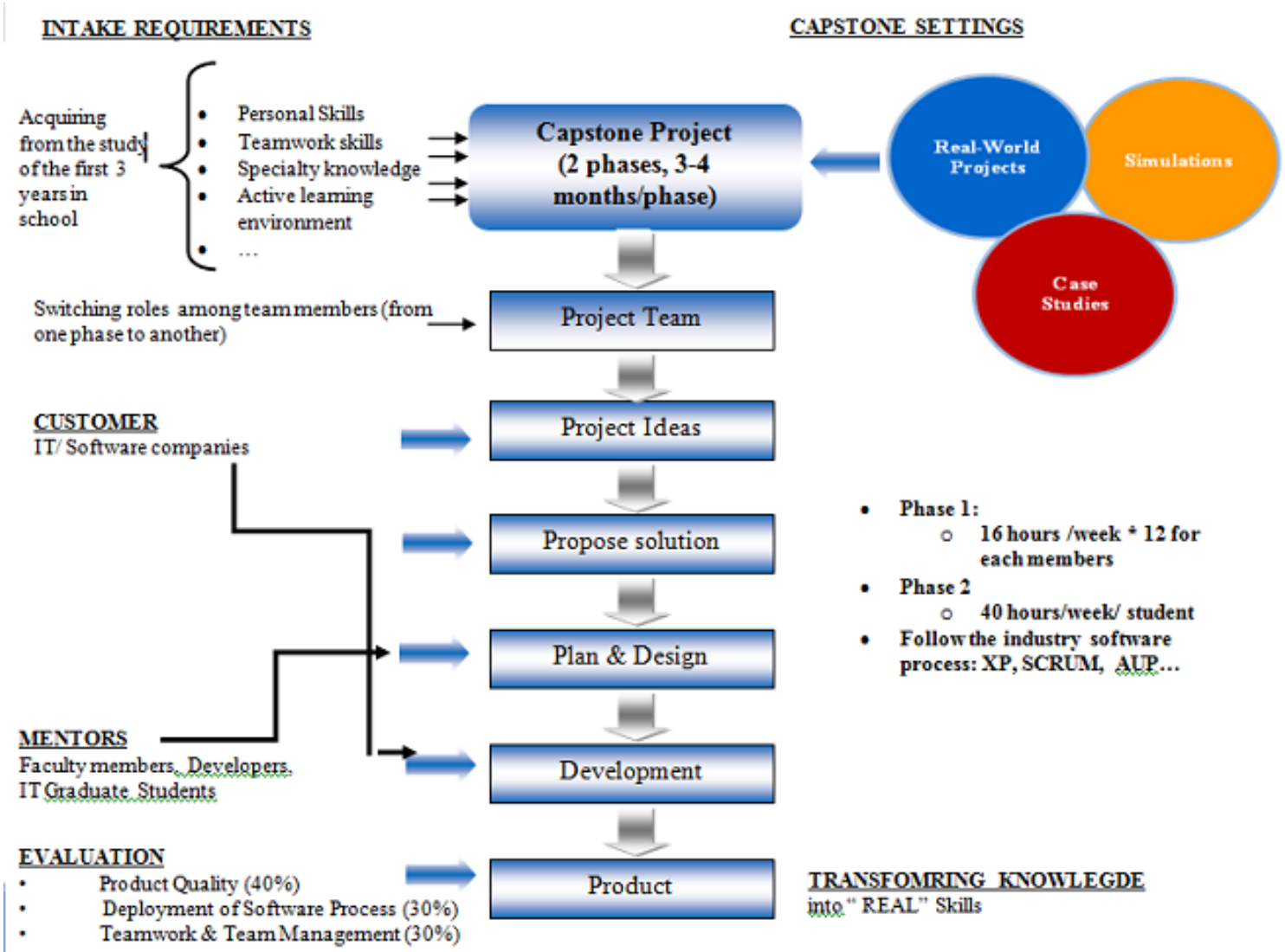
- **The software and systems engineering world is extremely diverse:**
  - practitioner specialization is required
  - domain experience is also important
  - the theoretical body of knowledge is large and
  - the theoretical body of knowledge is large and growing rapidly
  - systems must possess critical quality attribute properties – failure of which can cause loss of life, limb, and/or fortunes!
    - availability
    - security
    - safety, and many others

Time frame for Capstone at DTU

Figure 1. Students at Different Stages of IT CDIO/Capstone projects



IT Capstone Projects Implementation



In This Lecture We Will Discuss…

- **Project Selection**
  - Marketing
  - Identifying Projects
  - Final Project Selection
- **Student Orientation**
  - Program overview
  - Program overview
  - Project Based Education
  - Client Project Presentations
- **Student Team Formation**
  - Student's Prioritized "Wish List"
  - Assigning Students to Teams
  - Mentor Assignments

More On These Later…

- Key Team Milestones
  - Initial team meeting(s)
  - Weekly team status meeting(s)
  - Individual meetings
- Presentations
  - Mid-Semester Review
  - End of Semester Presentation
- Grading and Evaluation
  - Project meeting
  - Black Friday

Key Milestones and Activities

- Project Selection
  - Marketing
  - Identifying Projects
  - Final Project Selection

Happens Before Students Arrive

- Student Orientation
  - Program overview
  - Project Based Education
  - Client Project Presentations
- Student Team Formation
  - Student's Prioritized "Wish List"
  - Assigning Students to Teams
  - Mentor Assignments

Setting Expectations

- It is important to establish realistic expectations with respect to project scope.
  - An educational program may be 1 or more years in duration, but not all of that can be dedicated to project time – *obvious to faculty, not to customers*
  - Also, not all student project time is "coding time" - these projects have all the same aspects of management, meetings, design, and so forth that real projects do.
  - Students must study – this is the focus of the program – *the project must support learning, not replace it.*
  - Base scope estimates on actual person hours NOT calendar time.

Example

- Approximate Practicum Man Hours:
  - 1st semester: 12 hours per student per week
  - 2rd semester: 24 hours per student per week
  - Given 16 weeks/semester and 2 semesters:
    - 16 weeks @ 12 hours per person per week = 192
    - 16 weeks @ 24 hours per person per week = 384
  - Total hours per student spent on practicum = 576
  - Assume a team size of 2 persons – each team has 576 X 2 = 1152 person hours of project time during the program
  - This is one full-time engineer for 29 weeks, or about ½ of a person year of effort (40 hr work week)

Project Course Description

- It is important that students understand that project courses are *self directed* and we expect students to *behave like professionals*:
  - be proactive in all ways
  - honor your commitments to the team
  - show up on time for team meetings
  - do your fair share of the work
  - leave your ego behind
  - be courteous

Nominal Project Activities

- Key Team Milestones
  - Weekly team status meeting(s)
  - Individual meetings
  - Team meeting guidance
- Presentations
  - Nominal Project Artifacts
  - Mid-Semester/Mid-Project Reviews
  - End of Semester/Project Presentations
- Grading and Evaluation
  - Project Grading Meeting
  - Black Friday Review
  - General Timeline of Events

Initial Team Meetings

- Each week there are two key meetings that involve faculty – *team and individual mentor meetings:*
  - Teams are responsible for arranging their own meeting and status meetings with the mentor.
  - Individual team members are responsible for arrange individual mentor meetings with students.
- Students are responsible for setting up the initial meetings with the mentors to start the project.

Team Status Meeting – 1

- This meeting is held each week with the entire team and the mentor - *nominally about 1 hour or so.*
- The purpose of the meeting is:
  - review general progress of the team
  - review specific artifacts and/or activities such as plans, estimates, designs, and so forth…
  - provide an opportunity for the students to ask the advice of the mentor
  - give the mentor an opportunity to offer feedback, advice, and course corrections to the team as a whole

Team Status Meeting – 2

- During these meetings, the mentor should observe the following:
  - How often and how well is the team interacting with the client?
  - Does the team have established processes and are they following them?
  - Does the team understand the problem and the business context?
  - Do they have or are they working toward a requirements specification?
  - Are they estimating the work using some disciplined technique?

Team Status Meeting – 3

- In addition to observing the team's performance on the project task, mentors should be alert for any signs of conflict or dysfunction on the team.
  - Are there any members that do not say anything during any team meetings? Do any team members dominate the conversation?
  - Does there appear to be signs of conflict between team members that is evolving into more than intellectual disagreement?
  - Is the team fractioning into sides?
  - Has there been open hostility?
  - Is the team meeting together?
- Any of these could be signs of conflict – *we will discuss how to deal with team conflict later in the course.*

Individual Mentor Meetings – 1

- In addition to team meetings, students will also meet with mentors individually.
- Individual meetings provide opportunity for students to…
  - ask specific questions about, and get help with technical and programmatic problems
  - express and discuss their private issues and concerns
  - ask mentors about real world software engineering and get career advice

Individual Mentor Meetings – 2

- Individual meetings provide opportunity for mentors to…
  - see how individual students respond to probing questions
    - Consider an example where you ask team members individually about planning, and you get different answers, this is an indication that there is probably a problem with planning in the team.
  - probe students to see how they are holding up under pressure
  - offer course correction advice in private regarding less than desirable performance or behavior

Conducting Meetings

- Meetings are the largest time drain in industry *and on our projects.*
- One thing that our students learn is how to plan and execute efficient meetings.
- To be effective, meetings:
  - need to be planned and managed
  - must have clear objectives
  - should help the team accomplish its goals, not hinder it
- Students never seem to know how to conduct effective meetings – *and must be taught how.*

eeting Protocol

- All of our time (student and faculty) is extremely valuable. For this reason we coach teams on a rigid meeting protocol.
  - Agenda and Meeting Notification
  - Meeting Preparation
  - Meeting Roles
  - Managing Time
  - Action Items
  - Explicit Meeting Strategy
- Teaching students how to conduct efficient meetings helps them use their (and yours) time more efficiently .

Agenda

- The agenda includes:
  - time, date, location, start and stop time for the meeting
  - the objective(s) and/or desired outcomes of the meeting required preparation – include any necessary materials
  - topics for discussion and the amount of time allocated for each topic
  - presenter or discussion leader for each topic or "all" for items involving all participants

Example Agenda

MEETING NOTIFICATION		
Meeting: Bi-Weekly Team Status **Date:** 10/20/2009 **Time:** 9AM – 10AM **Purpose:** To review team status **Requestor:** Dr. Silva		
Agenda Item	**Duration**	**Who**
Results of Client Meeting	15	Jeff
Final SRS Status	15	Tom
Design Team Readiness Status	20	Robert
Review Action Items and Wrap-up	10	All

- It can be helpful to create email templates to create meeting agendas.
- We enforce a 48 hour advance notification protocol unless it is a reoccurring meeting

Example Agenda

SECTION -1 (to be completed by the STUDENT prior to meeting)	
Students' Names:	Supervisor Name:
Date:	Date of previous meeting:
Work undertaken since last meeting:	
Issues you would like to discuss in this meeting:	
SECTION -2 (to be completed by the SUPERVISOR at the meeting)	
Work student should undertake between now and next meeting:	
SECTION -3	
Date of next meeting:	
Signatures:	Student (Team Leader):
	Supervisor:

Nominal Project Artifacts

- Normally project artifacts include:
  - Statement of Work (SOW)
  - Project Plans and Process Description
  - Requirements
  - Designs
  - Test Plans
  - Source Code
- Other artifacts may include
  - manuals (user, installation, maintenance),
  - schedule tracking, process data, web pages, and others…

SOW

- The purpose of the statement of work is to describe the relationship between the client and the team. Includes
  - description of client and domain
  - description of identifies and describes team
  - general project description and outline
  - general responsibilities of team AND client
  - any contractual issues (required resources, payment responsibilities, and so forth)

Project Plans and Process

- Project planning should describe how the team will
  - elicit, analyze, and manage requirements
  - long term, strategic plans as well as day-to-day tactical plans (usually 2-4 week scope)
  - project tracking and oversight to include
    - project performance measurement
    - work breakdown structures
    - task estimation
  - quality assurance strategy
  - configuration management
  - project process and development lifecycle
- May be in one or several documents; paper or electronic, but readily accessible by all (including the mentor).

Requirements

- There are many ways to document requirements, but should include:
  - functional requirements – what the system or product must do
    - "shall" statements, use cases,…
  - quality attribute requirements – how the system or product does what it does
    - performance, scale, modifiability, security,…
  - business constraints – premade business decisions
    - time, cost, organization, business model,…
  - technical constraints – premade design decisions
    - OS, language, hardware, commercial or open source elements,…

Designs

- For our projects, we require that our students create architectural design documents.
  - describes coarse-grained design decisions
  - sets the stage for detailed design and/or implementation
- Detailed designs are typically created as part of the implementation phase.
  - students may use a variety of representation techniques as well as structured or object-oriented design paradigm

Test Plans

- Students should describe how they plan to test the product or system to ensure that it meets the architectural drivers.
- Test plans should address:
  - unit, integration, and deployment testing
  - user interface quality testing if applicable
    - students should how they will validate qualitative measures such as such as "easy to use"
  - estimate for how much time will be devoted to test
  - test schedule
  - resources required (users, environments, equipment, and so forth)
  - product or system acceptance criteria
  - test coverage

Source Code

- Our students will produce source code as part of the project, and our embedded software engineering student may also develop some hardware as well.
  - we typically evaluate code vis-à-vis a demonstration
  - while we want our students to produce quality code, we do not routinely inspect code – *there is usually too much for complete faculty review*
  - We are interested in code quality (structured and commented) as well as functionality.

Project Reviews

- For the 2 semester practicum projects:
  - these teams are required to provide a graded mid-project presentation and a final-project presentation
  - essentially this is one presentation at the end of each semester they are in the project course

General Content of These Presenations

- Every team provides a presentation that ALL students and ALL mentors see that covers the following topics:
  - project context: project overview, client description, business goals and needs
  - description of project plans and processes
  - tracking data and quantitative measure of progress
  - presentation of technical work and challenges
  - next steps
  - lessons learned

Project Grading And Evaluation

- Evaluating student performance and assigning a grade can be challenging because there is a subjective element:
  - evaluate the quality of the artifacts the team produced
  - determine if the progress made is reasonable and is somewhat correlates with what the team planned to do
  - judge whether the team is learning: *are they trying to apply good software engineering practices or are they just hacking out a solution*

Peer Reviews

- The team projects are graded largely as teams, but we do include individual evaluation in terms of 360° peer reviews
  - each team member will complete a survey that rates their performance and the performance of their team mates
  - each student on a team will complete the survey and send it to their mentor
  - the data is keep private from the students, but the mentor uses the data to look for trends:
    - may show a student that is not fairly contributing
    - indicate signs of conflict

Peer Review Elements

Software Engineering

1. Applies knowledge gained from core MSE courses (where appropriate)
2. Applies SE "best practices" to his/her own working environment
3. Recommends new methods, alternatives or solutions to improve team productivity or quality
4. Avoids an ad hoc, unprofessional approach to software development

Software Process

1. Follows established group processes
2. Solves basic problems using existing procedures and guidelines
3. Looks for opportunities to improve defined group processes (where appropriate)

Team Work

1. Interacts with others in an effective, tactful and pleasant manner
2. Helps other team members who need assistance
3. Identifies barriers to effective team work
4. Facilitates informal discussions between peers to meet team objectives
5. Knows when to involve team members to solve problems or complete tasks

Effective Communication

1. Expresses ideas in a clear, concise manner
2. Effectively uses data and/or diagrams to communicate with team members
3. Regularly listens to and understands others' viewpoints before speaking
4. Encourages team members to share new or risky ideas

Leadership

1. Provides a positive example that others choose to follow
2. Demonstrates initiative within his/her assigned area of responsibility
3. Proactively identifies and addresses team issues and problems
4. Is open and flexible to new ideas that may alter team direction

Project Grading Meetings

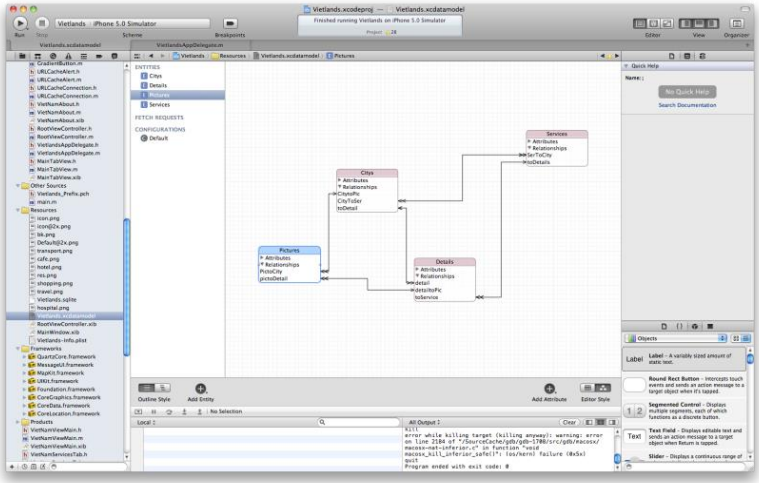
- Project grading meetings are held at the end of each semester to grade individual and team progress.
  - All mentors attend and contribute to the grading process.
  - Mentors complete a grading form for their teams (the 360° peer review form data is included in the mentor judgment).
  - Each team's grades are discussed by the group and final *individual grades* are assigned
    - there is usually a close correlation between team members' grades
    - not all team members will always get the same grade

Project Grading Criteria

Criteria	Description
Attitude	Student should show interest and participate actively in the project. Student should show strong commitment and sense of responsibility.
Initiative	Student should look for means to solve problems diligently
Knowledge	Student should show that they understand the project well. Student should be able to apply relevant knowledge acquired in school and show competency in solving the problem.
Product	Student should design and implement a product/an application that meets specifications, is functional, reliable and practical
Documentation	Student's project report should express ideas and concepts orderly, comprehensively and logically. Student should explain technical specifications and achievements clearly and comprehensively.
Presentation	Student's presentation should have good flow, is relevant, comprehensive and clear. Student should use aids that are relevant and clear. Student should show a good command of language and manage his presentation time well.

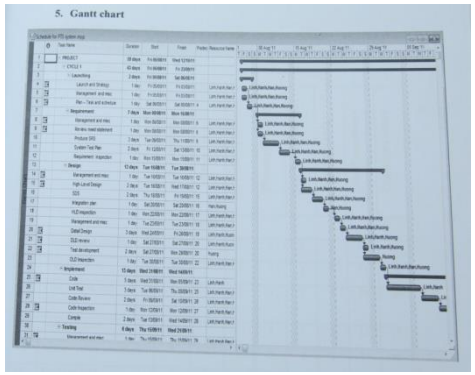
Capstone 1

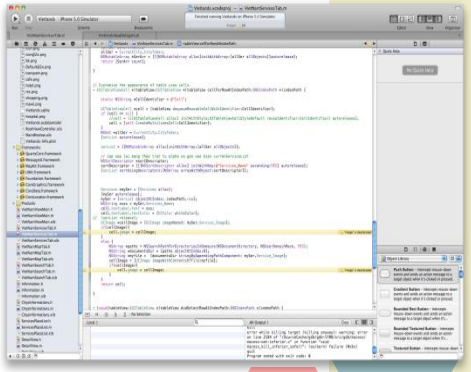
(Analysis & Design, Group Work, Presentation)



Capstone 2

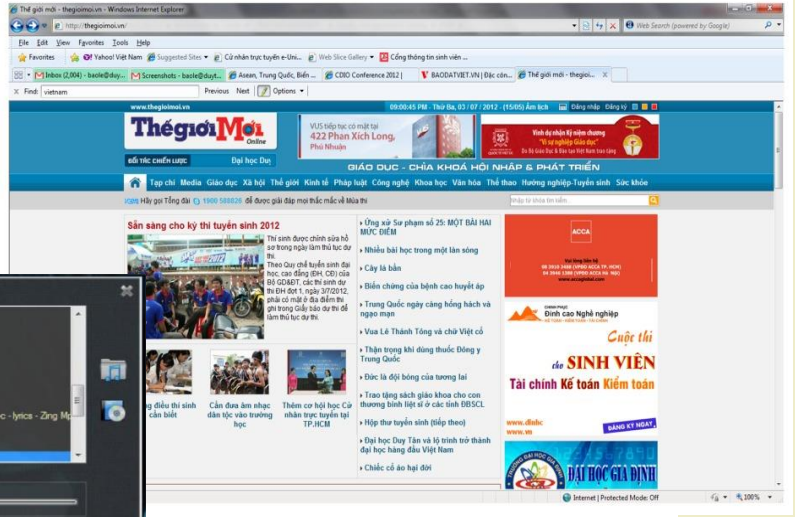
(A&D, Process, Coding, Prototyping)





Capstone 3

(graded by faculty members from Carnegie Mellon, IT companies)



- **Development methodology: Scrum, Agile**
- **ALM tools: TFS, Redmine, Jenkins, GIT**
- **Application types:**
  - Web application (mostly Microsoft Technologies)
  - Desktop application
  - Mobile application (iOS, Android, Windows Phone…)
  - SaaS application deployed on cloud environment

Q&A

Thank you very much!

<Project Title>

Background	*One paragraph introducing and motivating the problem.* *Should answer: Which area of computer science/computer engineering/information systems is this about? What particular part of that area? Why is this important?*
	2-3 paragraphs giving more detailed background. *Should answer: What has been done by others in this area? What is the current state of the art?*
Project Scope	*Project scope statement*
Project Description	*1-2 paragraphs detailing the gap in our current knowledge.* *Should answer: What is missing in our current knowledge? What is the main purpose of doing this project? What are the main features of this project?*
Expected Outcome	*State the concrete results that will be the deliverables/output from the project.*
Method/Approach	*In what way and process that you can reach your goal/result?*
Relevant references	*Books, journals, conference papers, and (not many) some internet links*