ARIZONA STATE UNIVERSITY

GENERAL STUDIES COURSE PROPOSAL COVER FORM

Course information:

*Copy and paste **current** course information from Class Search/Course Catalog.*

College/School: Ira A. Fulton Schools of Engineering | Department/School: **CIDSE**

Prefix: **SER** | Number: **416** | Title: Software Enterprise: Process and Project Management | Units: 3

Course description: **Project-centric course focusing on applying software process, project management, and technical leadership. Final course in the software enterprise sequence.**

Is this a cross-listed course? No | If yes, please identify course(s):

Is this a shared course? No | If so, list all academic units offering this course:

*Note- For courses that are crosslisted and/or shared, a letter of support from the chair/director of **each** department that offers the course is required for **each** designation requested. By submitting this letter of support, the chair/director agrees to ensure that all faculty teaching the course are aware of the General Studies designation(s) and will teach the course in a manner that meets the criteria for each approved designation.*

Is this a permanent-numbered course with topics? Yes

If yes, all topics under this permanent-numbered course must be taught in a manner that meets the criteria for the approved designation(s). It is the responsibility of the chair/director to ensure that all faculty teaching the course are aware of the General Studies designation(s) and adhere to the above guidelines.

Chair/Director Initials: **KG** (Required)

Requested designation: Literacy and Critical Inquiry–L **Mandatory Review:** Yes

*Note- a **separate** proposal is required for each designation.*

Eligibility: Permanent numbered courses **must** have completed the university's review and approval process. For the rules governing approval of omnibus courses, contact Phyllis.Lucie@asu.edu.

Submission deadlines dates are as follow:

For Fall 2018 Effective Date: October 1, 2017

For Spring 2019 Effective Date: March 10, 2018

Area(s) proposed course will serve:

A single course may be proposed for more than one core or awareness area. A course may satisfy a core area requirement and more than one awareness area requirements concurrently, but may not satisfy requirements in two core areas simultaneously, even if approved for those areas. With departmental consent, an approved General Studies course may be counted toward both the General Studies requirement and the major program of study.

Checklists for general studies designations:

Complete and attach the appropriate checklist

- Literacy and Critical Inquiry core courses (L)
- Mathematics core courses (MA)
- Computer/statistics/quantitative applications core courses (CS)
- Humanities, Arts and Design core courses (HU)
- Social-Behavioral Sciences core courses (SB)
- Natural Sciences core courses (SQ/SG)
- Cultural Diversity in the United States courses (C)
- Global Awareness courses (G)
- Historical Awareness courses (H)

A complete proposal should include:

- ☒ Signed course proposal cover form
- ☒ Criteria checklist for General Studies designation being requested
- ☒ Course catalog description
- ☒ Sample syllabus for the course
- ☒ Copy of table of contents from the textbook and list of required readings/books

It is respectfully requested that proposals are submitted electronically with all files compiled into one PDF.

Contact information:

Name: Kevin Gary | E-mail: kgary@asu.edu | Phone: 7-1373

Department Chair/Director approval: *(Required)*

Chair/Director name (Typed): Kevin Gary | Date: 9/29/17

Chair/Director (Signature):

Arizona State University Criteria Checklist for

LITERACY AND CRITICAL INQUIRY - [L]

Rationale and Objectives

Literacy is here defined broadly as communicative competence—that is, competence in written and oral discourse. **Critical inquiry** involves the gathering, interpretation, and evaluation of evidence. Any field of university study may require unique critical skills that have little to do with language in the usual sense (words), but the analysis of written and spoken evidence pervades university study and everyday life. Thus, the General Studies requirements assume that all undergraduates should develop the ability to reason critically and communicate using the medium of language.

The requirement in Literacy and Critical Inquiry presumes, first, that training in literacy and critical inquiry must be sustained beyond traditional First Year English in order to create a habitual skill in every student; and, second, that the skill levels become more advanced, as well as more secure, as the student learns challenging subject matter. Thus, two courses beyond First Year English are required in order for students to meet the Literacy and Critical Inquiry requirement.

Most lower-level [L] courses are devoted primarily to the further development of critical skills in reading, writing, listening, speaking, or analysis of discourse. Upper-division [L] courses generally are courses in a particular discipline into which writing and critical thinking have been fully integrated as means of learning the content and, in most cases, demonstrating that it has been learned.
Notes:

1. ENG 101, 107 or ENG 105 must be prerequisites
2. Honors theses, XXX 493 meet [L] requirements
3. The list of criteria that must be satisfied for designation as a Literacy and Critical Inquiry [L] course is presented on the following page. This list will help you determine whether the current version of your course meets all of these requirements. If you decide to apply, please attach a current syllabus, or handouts, or other documentation that will provide sufficient information for the General Studies Council to make an informed decision regarding the status of your proposal.

Revised April 2014

Proposer: Please complete the following section and attach appropriate documentation.

ASU - [L] CRITERIA

TO QUALIFY FOR **[L]** DESIGNATION,THE COURSE DESIGN MUST PLACE A MAJOR EMPHASIS ON COMPLETING CRITICAL DISCOURSE--**AS EVIDENCED BY THE FOLLOWING CRITERIA:**

YES	NO		Identify Documentation Submitted
☒	☐	**CRITERION 1:** At least 50 percent of the grade in the course should depend upon writing assignments (see Criterion 3). Group projects are acceptable only if each student gathers, interprets, and evaluates evidence, and prepares a summary report. *In-class essay exams may not be used for [L] designation.*	Syllabus, example writing assignments and student work

1. Please describe the assignments that are considered in the computation of course grades--and indicate the proportion of the final grade that is determined by each assignment.

2. **Also:**

Please **circle, underline, or otherwise mark** the information presented in the most recent course syllabus (or other material you have submitted) that verifies **this description** of the grading process--and label this information **"C-1".**

C-1

☒	☐	**CRITERION 2:** The writing assignments should involve gathering, interpreting, and evaluating evidence. They should reflect critical inquiry, extending beyond opinion and/or reflection.	

1. Please describe the way(s) in which this criterion is addressed in the course design.

2. **Also:**

Please **circle, underline,** or **otherwise mark** the information presented in the most recent course syllabus (or other material you have submitted) that verifies **this description** of the grading process--and label this information **"C-2".**

C-2

☒	☐	**CRITERION 3:** The syllabus should include a minimum of two writing and/or speaking assignments that are substantial in depth, quality, and quantity. Substantial writing assignments entail sustained in-depth engagement with the material. Examples include research papers, reports, articles, essays, or speeches that reflect critical inquiry and evaluation. Assignments such as brief reaction papers, opinion pieces, reflections, discussion posts, and impromptu presentations are not considered substantial writing/speaking assignments.	

1. Please provide relatively detailed descriptions of two or more substantial writing or speaking tasks that are included in the course requirements

2. **Also:**

Please **circle, underline,** or **otherwise mark** the information presented in the most recent course syllabus (or other material you have submitted) that verifies **this description** of the grading process--and label this information **"C-3".**

C-3

ASU - [L] CRITERIA

YES	NO		Identify Documentation Submitted
☒	☐	**CRITERION 4:** These substantial writing or speaking assignments should be arranged so that the students will get timely feedback from the instructor on each assignment in time to help them do better on subsequent assignments. *Intervention at earlier stages in the writing process is especially welcomed.*	

1. Please describe the sequence of course assignments--and the nature of the feedback the current (or most recent) course instructor provides to help students do better on subsequent assignments

2. **Also:**

Please **circle, underline,** or **otherwise mark** the information presented in the most recent course syllabus (or other material you have submitted) that verifies **this description** of the grading process--and label this information **"C-4".**

C-4

Course Prefix	Number	Title	General Studies Designation
SER	416	Software Enterprise: Process and Project Management	L

Explain in detail which student activities correspond to the specific designation criteria.
Please use the following organizer to explain how the criteria are being met.

Criteria (from checksheet)	How course meets spirit (contextualize specific examples in next column)	Please provide detailed evidence of how course meets criteria (i.e., where in syllabus)
C-1	The Exercises and Project categories of assessments on the syllabus consist primarily for outcomes directly connected to the L requirements, such as writing PM plans according to software engineering standards, and using evidence-based techniques to evaluate progress	In the attached syllabus the areas related are highlighted in underscored italic red font.
C-2	The project and exercises include activities to critically evaluate evidence gathered on a project and apply decision-making procedures across alternatives justified by that analysis	In the attached syllabus the areas related are highlighted in underscored italic red font.
C-3	The writing and oral activities in the project context are examples of this. Example project activity descriptions and student work are included.	In the attached syllabus the areas related are highlighted in underscored italic red font. Further, the team semester project deliverables outline and project activity overview are attached.
C-4	The course is part of the Software Enterprise project-based learning sequence. In Enterprise courses students apply Kolb's active learning approach to get rapid feedback on project activities, and include metacognition (individual and team reflections) at frequent intervals. The Software Enterprise forms the core curricular design structure of the BS in SE, has received NSF and other funding, and has been published (sample references provided) in several venues.	An Enterprise project description is included, and publications in the software engineering education research literature are available, including: 1. Gary, K. "The Software Enterprise: Practicing Best Practices in Software Engineering Education", The International Journal of Engineering Education Special Issue on Trends in Software Engineering Education, Volume 24, Number 4, July 2008, pp. 705-716. 2. Gary, K., Lindquist, T., Bansal, S., and Ghazarian, A. "A Project Spine for Software Engineering Curricular Design", Proceedings of the 26th Conference on Software Engineering Education & Training (CSEET 2013), Co-located with ICSE 2013, San Francisco, CA, May 2013. 3. Gary, K., "The Software Enterprise: Preparing Industry-ready Software Engineers" Software Engineering: Effective Teaching and Learning Approaches, (book chapter) Ellis, H., Demurjian, S., and Naveda, J.F., (eds.), Idea Group Publishing. October

		2008. 4. Gary, K., Koehnemann, H., and Gannod, B. "The Software Enterprise: Facilitating the Industry Preparedness of Software Engineers" National Conference of the American Society for Engineering Education (ASEE 2006), Chicago, IL, June 2006.

REQUIRED SYLLABUS ITEMS PER ACD 304-10

Course Number: *SER416*
Course Title: *Software Enterprise: Project and Process Management*
Credits: *3*
Prerequisites: *SER316 with C or better*
Catalog Description: Project-centric course focusing on applying software process, project management, and technical leadership. Final course in the software enterprise sequence.

1. Contact Information

   Course Coordinator: Ashraf Gaffar, PhD, PMP

2. Office Hours

   <Determined each session/semester>

3. Course Learning Outcomes (Mapping to Program Outcomes in parentheses):
   1. **(Technical Competence)** Explain the role of a project manager in the software industry, and how this role evolved over time.

      Assessable Items (AIs): exams, exercises

   2. **(Technical Competence)** Describe the *knowledge areas* of project management and their *associated processes*, and how they are applied in software projects.

      AIs: exams, exercises

   3. **(Software Engineering Practice)** Apply the integration of relevant project management concepts, techniques, and tools in managing various aspects of a software-intensive project (charter, WBS, scope, schedule, budget, and risk management)

      AIs: semester project

   4. **(Critical Thinking and Decision Making)** Collect and analyze process-related data and use it as a basis for comparing alternative strategies for project success.

      AIs: semester project, exercises

   5. **(Communication)** Express project status and planning in written form according to relevant software engineering standards at a professionally-accepted level.

      AIs: semester project reports, exercises

   6. **(Communication)** Communicate and negotiate with project team members and leadership to justify project decisions

      AIs: semester project meetings and presentations

4. Grading Policies

Assessable Item	% of Grade	Description
Exams	30%	Midterm worth 10%, Final worth 20%
Exercises	*20%*	*(L) in-class and outside of class activities (5)*
Project	*50%*	*(L)*

Exams: 2 exams will be given, one at roughly the midterm point of the class and one during the regularly scheduled final exam period. The final exam is cumulative. Exams focus on the foundational body of knowledge in project and process management for software engineering.

Exercises: (Approximately, at the instructor's discretion) 5 homework exercises will be given. In keeping with the Enterprise pedagogy, these exercises will typically be started on in-class and completed outside of class. These exercises focus on skills building in particular project and process management areas, preparing you for application in the semester project. *One-half of these exercises will focus specifically on Critical Thinking and Communication outcomes (outcomes 4 and 5 above) and contribute to the Literacy (L) requirement for the course*.

Project: A team-based semester project will be assigned to provide an opportunity to "learn in context". In this project you will apply the skills and concepts from the lecture and exercise components of the course in the context of a software project. Your team will be asked to *prepare a project charter, a development plan, a quality assurance plan, a change management plan, and a risk management plan*. *Your team will monitor project progress according to these plans, and make strategic and tactical alterations to these plans to ensure project effectiveness. In accordance with the Enterprise pedagogy, teams and individuals will conduct meta-cognitive activities (reflection) on project and process management concepts*.

Grading assignments are made according to the standard grade scale expectations published by ASU at https://students.asu.edu/grades. These definitions are mapped into this numeric scale:

COURSE GRADING	
Based on Points (**absolute, fixed, no curve**)	
>= 98.0 <= 100.0	**A+**
>= 92.0 < 98.0	**A**
>= 90.0 < 92.0	**A-**
>= 87.5 < 90.0	**B+**
>= 82.5 < 87.5	**B**
>= 80.0 < 82.5	**B-**
>= 77.5 < 80.0	**C+**
>= 72.5 < 77.5	**C**
>= 70.0 < 72.5	**C-**
>= 60.0 < 70.0	**D**
< 60.0	**E**

5. Absence & Make-Up Policies

As a team-based project-centric course, students are expected to attend class to keep up with the material and not adversely impact their teammates's success. Accommodations will be made for religious observances provided that students notify the instructor at the beginning of the semester concerning those dates. Students who expect to miss class due to officially university-sanctioned activities should inform the instructor early in the semester. Alternative arrangements will generally be made for any examinations and other graded in-class work affected by such absences.

6. Readings, Assignments, Examinations, Special Materials, Required Activities

Required Text Book:

Information Technology Project Management, Revised 7th Edition, by Kathy Schwalbe

Publisher: Course Technology; 7th edition (September 9, 2013)
Language: English
ISBN-10: 1285847091
ISBN-13: 978-1285847092

Additional Readings:

User Story Mapping: Discover the Whole Story, Build the Right Product, 1st Edition by Jeff Patton

Publisher: O'Reilly Media; 1 edition (September 25, 2014)
Language: English
ISBN-10: 1491904909
ISBN-13: 978-1491904909

7. Classroom Behavior

Cell phones and pagers must be turned off during class to avoid causing distractions. The use of recording devices is not permitted during class. Any violent or threatening conduct by an ASU student in this class will be reported to the ASU Police Department and the Office of the Dean of Students.

8. Academic Integrity

All students in this class are subject to ASU's Academic Integrity Policy (available at http://provost.asu.edu/academicintegrity) and should acquaint themselves with its content and requirements, including a strict prohibition against plagiarism. All violations will be reported to the Dean's office, who maintain records of all offenses. Students are expected to abide by the FSE Honor Code (http://engineering.asu.edu/integrity/). Specific rules for this class are:

A. practicing any form of academic deceit;
B. referring to materials or sources or employing devices (e.g., audio recorders, crib sheets, calculators, solution manuals, or commercial research services) not specifically authorized by the instructor for use during tests, quizzes, homework, and class activities;
C. acting as a substitute for another person in any academic evaluation or using a substitute in any academic evaluation;

D. possessing, buying, selling, or otherwise obtaining or using, without appropriate authorization, a copy of any materials intended to be used for academic evaluation in advance of its administration;
E. depending on the aid of others to the extent that the work is not representative of the student's abilities, knowing or having good reason to believe that this aid is not authorized by the instructor;
F. providing inappropriate aid to another person, knowing or having good reason to believe the aid is not authorized by the instructor;
G. submitting the ideas or work of another person or persons without customary and proper acknowledgment of sources (i.e., engaging in plagiarism);
H. permitting one's own ideas or work to be submitted by another person without the instructor's authorization; or attempting to influence or change any academic evaluation or record for reasons having no relevance to class achievement.
I. turning in work/code done by someone else or another pair/group
J. copying work/code done by someone else or another pair/group
K. writing code together with someone else or with another pair/group (unless expressly allowed by the instructor)

Further, as a team-based project class, you have a professional responsibility to your team to give your best effort on each project activity. Failure to do so will be considered an ethics violation.

The penalty for an Academic Integrity Violation on an exercise will be a reduction of a course letter grade for the first offense, and failure of the course for a second offense. The penalty for an Academic Integrity Violation on an exam or project activity is immediate failure of the course. The penalty for an ethics violation will be a zero for the project activity. All violations will be referred to the Dean's Office of the Ira A. Fulton Schools of Engineering.

13. Disability Accommodations.

Suitable accommodations will be made for students having disabilities and students should notify the instructor as early as possible if they will require same. Such students must be registered with the Disability Resource Center and provide documentation to that effect.

Notice: Any information in this syllabus (other than grading and absence policies) may be subject to change with reasonable advance notice.

Notice: All contents of these lectures, including written materials distributed to the class, are under copyright protection. Notes based on these materials may not be sold or commercialized without the express permission of the instructor.

The Software Enterprise in ASU's BS in Software Engineering

Kevin Gary, Associate Professor
School of Computing, Informatics, and Decisions Systems Engineering, Arizona State University
kgary@asu.edu, (480)727-1373

The Software Enterprise is a multi-semester sequence designed to accelerate student learning of practical, "real world" considerations in applied computer science. The Enterprise has several defining characteristics that separate it from other problem-based courses:

Continuous – The Enterprise is currently designed as a contiguous four-semester sequence. This sequence exposes students, in a specified order, to all phases and roles of the software process lifecycle, and provides a continuity to the experience.

Scalable – Students work through two project lifecycles instead of one while in the Enterprise. The continuous, yearlong project cycle allows for larger teams and larger projects, so students get exposed to a myriad of issues that occur when scaling up.

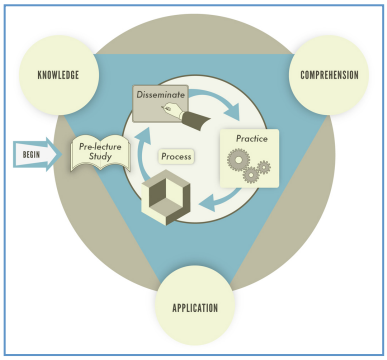
"Real-world" – Students are exposed to the full spectrum of forces affecting software development projects. Teams are asked to cope not only with technical issues but also with social or soft-skill issues with real-world industry sponsors.

Collaborative – Students work in teams across academic year boundaries. Students role-play, with participants responsible for different process-oriented roles. Teams deal with outside roles, such as customers and CXX-level management. Teams work with other Enterprise teams. For example, 4th semester students of the Enterprise mentor and manage students in the 2nd semester.

The five semester sequence leads students through "Personal Software Process", "Tools and Process", "Construction and Transition", "Inception and Elaboration", and "Project and Process Management". By the conclusion of the Enterprise sequence, students have an appreciation for the role of software process, the challenges of software maintenance, the impact of open source, the pros and cons of off-the-shelf software integration, business considerations in building software, and other practical aspects of software development.

Our vision for the Software Enterprise is to educate students on the following principles:

- Provide a situated educational experience for students that expose them to impacts of organizational, social, and cultural influences on the success of software projects.
- Help students realize software decisions *are* business decisions; such decisions are not made based on technical attributes alone.
- Help students understand stakeholder expectations; Project Managers, Customers, End users, SMEs, Business Development, etc.
- Provide extended hands-on enterprises applying the techniques learned in traditional classroom settings so new graduates not only comprehend the subject content but know how and when to apply it.



- Have student learning evolution follow the learning track of new professionals so as to expedite the assimilation of our graduates as new hires in software organizations in the greater Phoenix metro area and beyond.

The Software Enterprise employs a teaching and learning model (shown at right) combining traditional lecture, group engagement, problem-centered learning, reflection, and hands-on practice on scalable applications. In this model, students attend a reduced lecture schedule to create awareness and understanding, put the lecture concepts into practice through group lab sessions each week, integrate these skills into the scalable yearlong project, reflect on this rapid learning experience, and engage in group discussion regarding lessons learned. It is our premise that this model facilitates comprehension through to applied knowledge, making Enterprise graduates better prepared for the workforce upon graduation.

The Enterprise is currently in its 12th year. With support from the National Science Foundation, Kaufmann Foundation, and ABOR, and State Farm, the Enterprise has engaged in multi-university projects with a variety of industry sponsors on over 75 projects.

References:

1. Gary, K., "The Software Enterprise: Practicing Best Practices in Software Engineering Education", *International Engineering Education Special Issue on Trends in Software Engineering Education*, 2008.
2. Gary, K., "The Software Enterprise: Preparing Industry-ready Software Engineers", *chapter accepted to Software Engineering: Effective Teaching and Learning Approaches and Practices*, Ellis, H. (ed.).
3. Gary, K., Koehnemann, H., and Gannod, B., "The Software Enterprise: Facilitating the Industry Preparedness of Software Engineers", Proceedings of the National Conference of the American Society of Engineering Education, 2006 (ASEE '06).
4. Gary, K., Gannod, G., Koehnemann, H., and Blake, M.B., "Educating Future Software Professionals on Outsourced Software Development", Proceedings of the National Conference of the American Society of Engineering Education, 2005 (ASEE '05).
5. Gary, K. Gannod, G., Koehnemann, H., Lindquist, T., and Whitehouse, R., "WIP: The Software Enterprise", Frontiers in Education, 2005 (FIE '05).

Support:

1. Gary, K. "The Software Enterprise: A Reinforcing Pedagogical Teaching Model", National Science Foundation Course Curriculum and Laboratory Improvement Phase I Grant (NSF CCLI I), January 2009-December 2001, $148,938.
2. Gary, K., Koehnemann, H., Sannier, A., Kagan, A., and Acharya, R., "Agile Methods for Entrepreneurship", Enterprise@ASU *Pathways to Entrepreneurship Grant (PEG)* Award ($39,731) through support from the Kaufmann Foundation, November 2007 (implementation 7/1/08-6/30/09). *This award supports a variety of activities, one of which is to introduce a cohort to the Software Enterprise, joining DCS students with B-school entrepreneurs.*
3. Gary, K. and Gannod, B., "The Software Enterprise: Preparing Industry-ready Software Engineers" Arizona Board of Regents Award ($24,041), Learner-Centered Education (LCE) program, April 2005. *This grant supported initial work creating LCE labs and post-graduations surveys of the Enterprise.*
4. Gary K., Ghazarian, A., and Mazzola, D. "The Software Enterprise", State Farm Foundation Good Neighbor Grant, $40K ($18K share). 2015

SER 416 Online
Class Project
Best Community Service

The objective of this project is to design and build a **prototype** for an interactive Web application for a community service named “Best Community Service”, provided by a non-profit organization. This organization aims at benefiting a small community by providing various kinds of services to the local community. The non-profit will charge small fees for some services to cover its costs. Other services will be offered free of charge. Additionally, the non-profit organization will rely on volunteers to provide some of the services offered to the community. It will also rely on a number of paid employees (full time and part time employees) to take care of managing the business and running the services, as well as the operational costs and other activities. The main sources of income for running the business will be donations from known and anonymous donors as well as from small fees charged for some the offered services. The organization will run from a local building, which offers large spaces for organizing social events like weddings, community meetings, and funerals. It will also offer rental service for equipment needed for the events, as well as catering services. Other in-house services include Spanish, arts and crafts, and painting classes given by volunteers. External services are also offered in terms of home care for the elderly and handicapped as well as shuttle services.

You are an expert software consultant. While the customer approached you with their “initial project description”, known as the customer “**Wants**”, it is your responsibility as an experienced software consultant to extract detailed requirements and design, and to add/modify the initial project description as needed then discuss them with the customer. This is typically known as the customer “**Needs**”. You can see the Wants as the customer’s wish list, while the Needs are the informed and validated details identified by software engineers. This will determine the software application that will be developed.

Initiation Tips:

- This is a typical high-level project description, not a final one
- Identify a clear set of high-level requirements extracted from the given description (the customer “Wants”)
- Make any necessary but reasonable assumptions needed
- Identify different user roles (personas) to help you better understand how the software will be used

- Identify and organize all detailed requirements and activities as elaborated from high-level requirements
- Propose any additional activities you deem important for the customer (customer “Needs”)

The Best Community Service contacted you as a well-know software consultant with the following mandate:

1- To help solidify the requirements and provide all necessary details (software requirements specification, SRS)
2- To provide a detailed software design for the application. The application will run online (design document)
3- To provide a detailed project management plan including the estimated scope, budget and schedule required to complete the project
4- To provide an interactive prototype of the software. The prototype will show all necessary screens of the software including home page design, and a list of services as well as a login page. Additional pages will show the details of services offered, and how users will request a service.
5- The prototype will only show the user-facing part of the Web application (all screens the user will interact with online). **No server-side screens are needed** (for example, no need to show how managers will run the business, or how HR person will process the payroll)
6- The prototype will run as a mockup, so backend data is NOT required in your prototype (for example, no files or database to store actual data is needed)
7- However, your project estimate **must** include the estimates of building a **complete online system** including both user-side and server-side application development estimates.
8- This means that you will use your experience in software engineering and software development as well as the detailed analysis and design of the prototype to gain sufficient insight into **accurately estimating** the actual costs of developing a **complete working system**.
9- Your deliverables will be distributed over the 7 modules and will include :
    a. An **SRS** including a list of services and a brief description of each of them
    b. A **design document** that include:
        i. A high-level wireframe showing the main screens and the relationship between them
        ii. A detailed list of screens showing their contents. No detailed visual design is needed for the screens

iii. A use-case diagram showing user roles and the main use cases
iv. A list of use cases (see Alistair Cockburn). Each use case will show a sequence of one successful scenario and one alternate scenario

c. Project management plan, including:
   i. Project scope statement
   ii. Project Work Breakdown Structure and Activities
   iii. Project Schedule and Budget
   iv. Project risk breakdown structure and matrix

d. A working prototype showing the main screens of the user side. You will need to submit both the source code (HTML) as well as screenshots of all developed screens.

10- The breakdown of these deliverables will be announced in each weekly module

Important Tips: While the working prototype is due at the end of class (week 7) as your final project deliverable, you will need to start working on it from week 1.

- **You will need to decide on and select suitable software tools to use for developing your prototype. Select tools that you are already familiar with.**
- **You will need to set up your development environment and make sure you're comfortable with it from week 1.**
- **As your design evolve, you should start working on your prototype progressively to gradually reflect your deign**

Outlines for SER 416 Online Project Deliverables

***Disclaimer**: These outlines are to be used as guidelines only. The information provided is a summary of deliverables to be covered in the class. While it is generally accurate, actual deliverables might vary, and will be announced in each class module on the blackboard.*

Preparation: Textbook and additional readings. See course syllabus

Important tips for your design and deliverables:
You can use any supporting project management tools / software, but this is optional, and not required. You can alternatively use Microsoft office (Word, PowerPoint, Excel) which are also commonly used in project management

Some deliverables have recommended size limit, which should normally be sufficient. If you need to, you can **slightly** exceed the recommended document size.

Consistency is important. In your screen designs, you need to give each screen a specific number or a representative name that should be consistent throughout your design.

Module 1: Introduction to Software Engineering Project Management

- Deliverables: **Assignment #1 (5%):**
    - A comparison between 1) Waterfall, 2) Spiral, 3) RUB, and 4) one additional process model of your choice. For each model, provide
        - One paragraph summary
        - One paragraph of advantages (bullet points)
        - One paragraph of disadvantages (bullet points)

    Size: 2-page max.
    Format: MSWord or PDF

Module 2: Detailed Case Study and Class Project

- Deliverables: **Assignment #2 (5%):**
    - A 5x10 Matrix showing the “process groups” mapping to the “knowledge areas”.

    Size: 1-page max.
    - Format: MSWord or PDF

Module 3: Project Scope Management

- Deliverables: **Assignment #3 (10%):**
    - A project Scope statement for the “Best Community Services” (1-2 pages, Word document)
    - A detailed WBS for the same project (1 -2 pages, word document)
    - The project SRS (see project description, optional size limit)

Module 4: Project Schedule Management

- Deliverables: **Assignment #4 (10%):**
    - A mapping from WBS to detailed activities (1 -2 pages, word document)
    - A project schedule with suitable milestones (2-3 pages, Word document with diagrams)
    - A high-level design draft of the online software:
        - A high-level wireframe (empty screens) showing the main functionality of each screen and the relationship between them (i-2 pages). **Each screen will be given a number or a name for reference and traceability**

        - A use-case diagram showing user roles and the main use cases (1-2 pages)

Module 5: Project Budget Management

- Deliverables: **Assignment #5 (10%):**
    - A project budget for the Best Community Services (1 -2 pages, word document)

Module 6: Project Risk Management

- Deliverables: **Assignment #6 (10%):**
    - A project risk breakdown structure (1 -2 pages, word/Power point/Pdf document)
    - A Risk Probability/Impact Matrix (1-2 page Word/Pdf document)

Module 7: Project Prototype Development

- Deliverables: **Assignment #7 (20%):**
    - A detailed (final) design document of the online software (no size limit):
        - A high-level **wireframe** showing the main screens and the relationship between them (copied from module 4)
        - A list of the main screens showing their contents. Screen contents in this list can be sketches, screen layout and text description of each screen; no detailed visual design is needed for the screens
        - A use-case diagram showing user roles and the main use cases (copied from module 4)
        - A list of all main use cases (see Alistair Cockburn: Writing Effective Use Cases 1st Edition or any similar book or online material). Each use case will show a sequence of one successful (main) scenario and one alternate scenario

Final Module: Project Delivery

- Deliverables: **Software Project (30%):**
    - A working prototype showing the main screens of the project and how to navigate between them according to your detailed design and requirements. The deliverable will include:
        - Source code
        - Screenshots of all screens, explaining the contents of each one.
        - Any supporting documents

BRIEF CONTENTS