

# Industrial Software Engineering

## CS522L

Instructor: Dr. Sridhar Chimalakonda

*with a pinch of fun,  
research, software, innovations!*

# Course Syllabus – What to Teach?

## **Course Objective:**

- To understand an industrial perspective of software engineering with a focus on modern software engineering principles and practices;
- To apply these principles and practices to develop tools and projects.

# Course Syllabus – What to Teach?

- Software engineering as a discipline; software process and product engineering; software development lifecycle models; agile software development;
- requirements engineering; software architecture; software design; Unified Modeling Language (UML); design patterns;
- software construction; testing; verification and validation; software metrics; software project management;
- advanced software engineering topics such as reuse, reengineering and evolution.

# Course Syllabus – What to Teach?

- Refresher of traditional software engineering;
- Current state of software engineering discipline;
- Modern software engineering process models;
- DevOps; Configuration management: Git, version control systems, continuous integration;
- Open source software development;
- Software architecture styles; Software testing techniques;
- Mining software repositories: software analytics and visualization of various artifacts such as architecture, code, bugs;
- Software quality and metrics;

# Software engineering as a discipline;

- Why Software Engineering? (*Ill* vs Well-formed, Precise Vs Imprecise)
- What is Software Engineering Anyway?
  - Past, Present and Future
    - AI-Based Software Engineering
    - Energy-aware Software Engineering
- Software Engineering as a Discipline, Nature
- Why is SE challenging?
- Programs Vs Software
- Who is a Software Professional?
- Software Quality Engineering
  - Kano Model, Deming, Garvin's, Qualities (Portability, Usability, Reliability...)

# software process and product engineering;

- How can we assess quality? – Verification and Validation
  - Formal Methods
  - Testing
  - Standards Compliance
- Software Engineering Principles – separation of concerns, abstraction, rigor and formality, anticipation of change, cohesion, coupling...

# lifecycle models; agile software development;

- The life cycle models
  - Feasibility study, Requirements analysis and specification, Design, Coding, Testing, Maintenance – Four types
- Traditional - Waterfall, Prototyping, Spiral, ...
- Agile – Chaos Report, Agile Principles, Practices, User Stories, Scrum...
- DevOps – Continuous integration, delivery, deployment
- CMMI Process models
- Influences – people, processes, tools, products

# requirements engineering;

- The worlds
- Requirements analysis and specification
- Domain analysis
- Types of requirements – functional and non-functional
- Software Requirements Specification
- Requirements Engineering process
- Good and Bad SRS - Incomplete, Inconsistent requirements
- Use Cases, User Stories
- Wireframing
- User Interface Design
  - Users, tasks, usability, Do's and Don'ts
  - UI Tools and User Experience



# software architecture; Unified Modeling Language (UML);

- Software Architecture
  - Context, Components, Connectors, Configurations, Constraints
  - Architecture Drift, Erosion, Prescriptive and Descriptive Architecture
- Modeling Software Architecture
  - The Zachman EA,
  - Models
- Unified Modeling Language
  - **12+ Diagrams**
  - 4+1 View Model
- Software Architecture Styles
  - Model View Controller
  - Layered
  - Client-server
  - Pipes and filters
  - Publish Subscribe
  - ...

# software design; design patterns;

- Design, Software Design – Feasibility and Desirability
- Design notations, tools, failures
- Software design is a wicked problem
- Object Oriented Design
- SOLID Principles
- Design Patterns – Creational, Structural, Behavioral

# software project management;

- Core Skills for a Software Engineer – Programming, Design, Process, Communication, Team work +Tools
- Software Planning, Estimation and Scheduling
  - Cone of uncertainty
  - Estimation methodologies
  - Work Breakdown Structure
  - Process and Product WBS
- Metrics
  - LOC, function points,

software construction; advanced software engineering topics such as reuse, reengineering and evolution

- Coding standards, platforms, languages, debugging, static analysis tools, what makes you a great software engineer?
- Anti-patterns
- Code smells
- Refactoring

# testing; verification and validation;

- Program dependency graphs
- *Testing, Static verification, Inspection, Formal proofs*
- Testing
  - Unit, Integration, System, Acceptance, Regression, Black-Box, White-box (glass-box), alpha, beta,
- Equivalence classes – split the input spaces uniquely!
- Coverage – statement, branch...

# Did we learn this?

- Software engineering as a discipline; software process and product engineering; software development lifecycle models; agile software development;
- requirements engineering; software architecture; software design; Unified Modeling Language (UML); design patterns;
- software construction; **testing; verification and validation;** software metrics; software project management;
- **advanced software engineering** topics such as reuse, reengineering and evolution.

# What should you do?

- Coding alone is not enough!
- Coding alone is not enough!
- Coding alone is not enough!
- Coding alone is not enough!
- Coding alone is not enough!
- Coding alone is not enough!
- Coding alone is not enough!
- Coding alone is not enough!
- Get deeper into SE, and
  - AI for Software Engineering
  - Software Engineering for AI
  - Energy efficient SE

# What are the key take-aways?



1  
Improve Software  
Engineering Tasks  
(Software Industry)

2  
Societal  
Challenges  
(Systems)

3  
Can AI  
replace  
people?

AI-Driven Software  
Development  
AI Maturity Models

AI-Driven  
Software for  
Society

1000x  
productivity?  
But still many  
open



Can we design  
Responsible  
Software?

Can we design  
Responsible AI  
Systems?

Can we design  
Responsible and Ethical  
AI+SE?

*AI that does not harm society*  
*Software that does not harm*  
*society*

## How Software Engineering chases you in your day to day life in Internship.

To: Sridhar Chimalakonda <ch@iittp.ac.in>

Mon, May 7, 2018 at 9:47 PM

"This is a first hand experience of mine during the initial days of internship. If I learnt anything, it would be this " Every Software Engineer needs to be a programmer, but not every programmer can be a Software Engineer ".

It started from Day 1 itself with the discussion of project that needs to be done over the 8 weeks of internship. The team mentor gave details of the product to be delivered in that time and asks for you to prepare a "Design Document" which encompasses all the requirements that are to be fulfilled by the product along with a detailed schedule of releases that are needed to be shown every week during the weekend sprints (Note:The schedule has to be meticulously prepared on your own).

It is followed by a detailed discussion with the mentor and manager to sort out the loop holes and refine it, it seems that one needs to refine it as many times as possible(from what I have heard).

The process that is being followed is something like an agile++ , each day starting with a stand up meeting with the mentor to review your's past day work and each week ending with a sprint assessing that week's release, followed by some not so good remarks(sometimes).The only thing which the mentor reiterates is software being developed by you is not your sole property to develop as you please, it needs to be extended or modified by anyone who's going to work for me in the future.

Its the overall condition in a startup, and Software Engineering class is not a joke, coding alone wouldn't get me what is being demanded here (For example I am asked to prepare an Ontology map for the User experiences of my project on the second day morning itself), the industry is cent times amplified than what is being seen in the lecture.

# The Internship Views

## 2018

# 2018

- “As part of our software engineering course, I learnt a lot about software development methodologies which I got to experience first hand such as VCS and scrum meetings. The knowledge and experience gained from doing projects in class was extremely useful.”
- “The work I could complete here was satisfactory. I have tried to develop as many addons as possible even got very encouraging results with some of them and my course on principles of Software Engineering has helped me quite a lot .”
- “Software Engineering lab: - The practices we followed in the SE lab are exactly what can be found in the industry. The Use cases practices we got has helped me to write PRD's for my projects. I utilized my knowledge of UML diagrams to create work flows for the bot which are afore mentioned. The rigorous schedule in SE lab has helped me to make deliverables for every week and achieve them.
- “The courses which i completed in the Institute helped and aided a lot during this Internship . The course of Software Engineering and Digital Systems were closely related to my work. The concepts taught in the courses were extremely beneficial for the project work.”

# 2018

- “The courses offered by my institute backed me up along the way. The software development models studied in Software Engineering helped me choose a path to work on the development, both during the development of proof-of-concept and extension of a product to use this technology.”
- “development of the software required meticulous application of the principles of software engineering, the course which I had done in my 5th semester helped a lot.”
- “I learned a lot about code management and review. My skills to read and understand code were challenged. I got to experience software engineering principles at various points of time.
- “Several concepts learned as part of coursework, especially about software design principles were very useful. “
- “Software Engineering course helped in reading the SRS documents, design document, ... and in various other ways.”

# 2019

- “As my project was purely based on Software development, the Software Engineering course proved very helpful for me. Thanks to it, I already had some insight over different project management models, Agile software development, Scrum framework, Kanban, Sprint and other similar terminologies.”
- “The Software Engineering course helped a lot through the development of the package. I was able to apply many principles like modularity, abstraction, generality, incremental development and anticipation of change. I also learnt that documentation, using comments, naming conventions, status tracker are very important and I was able to practically relate and understand its importance during my internship”
- “The project had a lot of set-backs, the architecture was changed multiple times. But the final architecture with updates was finally implemented and integrated into their main cluster. The process of updation of architecture was in coherence with the Agile Software Development principles learnt in Software Engineering course & Lab.”
- “Most importantly, the project taught me about delivering products/features within a deadline and also good software engineering practises.”

# 2020

- “In software engineering course, two of the important things that must be kept in mind while designing a software was feasibility and desirability. So, whenever I was trying to make a solution feasibility was the first thing that I was supposed to think. As the solution we design must be feasible and scalable.”
- “The project “We Are Here For You” - An Augmented Reality chatbot, done as part of a Software Engineering course has helped me a lot in completing my internship. Also we used Agile methodology and applied a few other concepts of Software Engineering in the due course of this project.”
- My experience with Software engineering lab was very helpful in understanding the software.
- The curriculum courses like Paradigms of Programming and Software Engineering proved to be helpful for my internship. The concepts of styles of programming, agile software development, team communication, and programming know-how learned during the software development in these courses were quite relevant to the work I did during my internship.



# 2020

- “Software Engineering course helped a lot throughout my internship. The principles learned during the course like write clean, manageable code, make design first, etc. helped me to simplify the work I did. During that course, I had a full go through how a software development project will go, and the lab course corresponding to the course, actually made me understand the use and benefits I will get when applying those principles in development.”
- “For this project, Software engineering theory and lab courses, other basic courses related to programming were helpful.”
- The software engineering course helped me in developing the platform and integrating the plots and automating the tasks.
- The entire theme of the project revolves around what makes it a repository, an engineered software one. All the attributes of the repo reaper quantify software engineering practices to be followed in a project.

# 2020

- “The courses studied in the college helped during this project, especially Software Engineering. Many concepts learnt in software engineering were applied during the internship process. We followed a life cycle model(agile); we gathered the requirements from the mentor, designed the specifications involving construction and iteration, tested with different code samples and modified it based on the feedback every week, sometimes every day through online meetings. We had to submit a weekly status report every week and report the progress every day to the mentor. Finally, we prepared the report document which contained the details of the tool like system requirements, technology stack, features and pseudo-code for the design.”

# The Evolution of Education

1970



Calculate the surface area of the object.

1985



Calculate the surface area of the rectangle.

2000



Calculate the surface area of the rectangle, multiplying the length by the width.

2010



Choose the correct answer.  
What is the surface area of the rectangle?  
☐ 4000  
☐ 600  
☐ 800000

2015



Choose the correct answer.  
What is the surface area of the rectangle?  
☐ Michael Jackson  
☐ Canada  
☐ 600  
☐ Breakfast

2018



Color the rectangle with the color you prefer.

edugog.com

Thank You ☺ ☺

Push your own boundaries! ☺

Contact me for anything!

Join RISHA Lab ☺ [with a caveat]