**DASS – Spring 2023**

**Course Objectives**

* Understand the process of building software, through a live project
* Inculcate software engineering knowledge and skills, and use essential technologies to build a reasonably complex piece of usable and maintainable software.
* Emphasis on structured approach and disciplined process (iterative) to develop software
* Enhance written and oral communication skills

**Instructor**

Ramesh Loganathan

Email: ramesh.loganathan@iiit.ac.in

**Lectures**

Monday and Thursday 9:00-10:30 AM. Tutorial weekly (Monday 6:30-7:30 PM).

**Books/Resources Recommended**

* Software Engineering – A Practitioner’s Approach, 7th Edition, Roger Pressman.
* Student Resources: http://highered.mcgraw-hill.com/sites/0073375977/student\_view0/ - has Chapter Overviews and Practice Quizzes
* Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, 3rd Edition, Craig Larman.
* Object-Oriented Software Engineering: Practical Software Development using UML and Java, Second Edition by Lethbridge and Laganiere
* Fundamentals of Software Engineering, Third Edition by Rajib Mall
* Head-First Object Oriented Analysis and Design, 1st Edition, Brett McLaughlin, Gary Police, and David West
* The Official web2py book: http://www.web2py.com/book
* http://rubyonrails.org/
* http://struts.apache.org/

**Grading distribution (tentative)**

| **Component** | **Percentage (%)** |
| --- | --- |
| Project | 40 |
| Client Feedback (R1 1% + R2 3%) | 5 |
| Coding Assignments (4) | 20 |
| Quizzes (Q1 + Q2, no midterm) | 10 |
| Class Assignments/quizzes | 5 |
| End Exam | 20 |
| **TOTAL** | 100 |
| BONUS | 5 |

There may be some BONUS activities from time to time.

**Traditional Grading (based on previous years – this is not final)**

>= 90.0 : A

>=85.0 and < 90.0 : A-

>=80.0 and < 85.0 : B

>=75.0 and < 80.0 : B-

>=70.0 and < 75.0 : C

>=60.0 and < 70.0 : C-

>=40.0 and < 60.0 : D

< 40.0 : F

Grades shall be on an absolute scale. The cut-off *may* be lowered if the instructors deems it necessary.

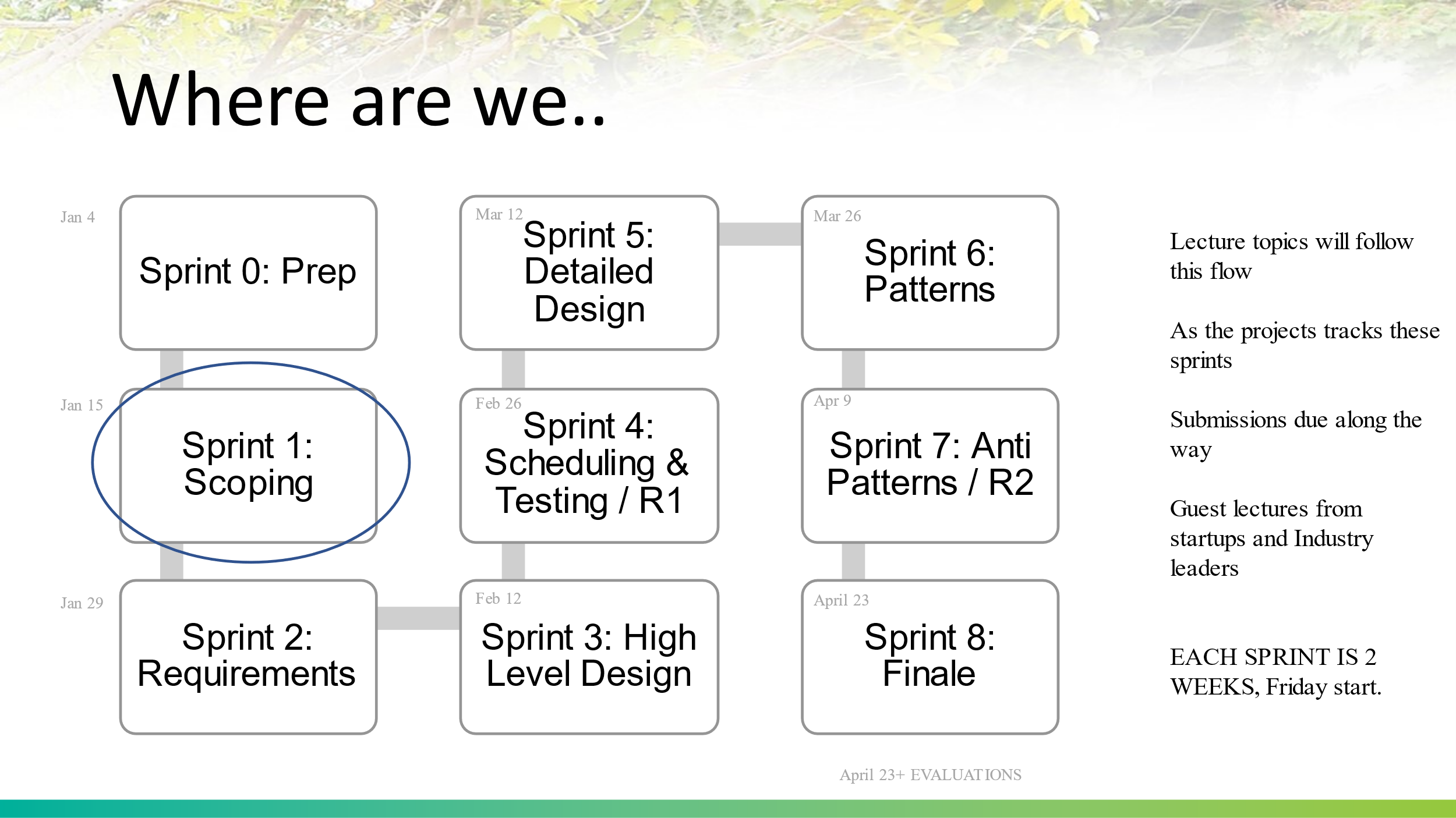
**Exams**

All exams will be given on topics covered up to that point. A comprehensive final exam is given during the finals week. Note that exam topics may cover class activities, assignments, project and text book reading - whether discussed in class or not.

**Assignments/Class Activities**

Impromptu in-class activities and take home assignments will be given throughout the semester to reinforce the material (not) taught in class. Some of these assignments/activities might be individual and some others might be team based. Please note there will be no makeup activities.

**OVERALL FLOW OF THE COURSE (in 2 week sprints):**



**Project**

**Team work**

The focal point of the course is a 3.5 month long project executed by a team of 3-5 students. The goal of the project is to introduce and practice the fundamental software development life-cycle activities of planning, tracking, designing, implementing and delivering an actual software product. Teams will be formed by Week 2 and will remain together throughout the semester. The instructor and TAs will choose the teams. A sincere effort will be made to accommodate one team member of your preference. However, there is NO guarantee you will get a project or team member of your choice.

For most of the students this might be the first time you are developing a team project. Working in teams will be challenging in various ways due to the short amount of time teams are able to meet in person. Teams have to decide on their meeting schedules and stick to it till the end of the semester. At times, the instructor may allocate some time during the class session for project teams to meet/work and clarify questions (if any). Hence, it is critical that you are present for all the classes as well as meetings.

Participating in the team project is perhaps the most valuable experience you will take away from this course. Far more projects fail as a result of poor collaboration and communication within project teams than they do for any technical shortcomings. Becoming an effective team member is a critical career skill and the one of the goals of this course is to gain some experience with effective (and non-effective) teaming practices. The expectation is that all will make an honest effort in participating in team activities to the best of their ability. Should non-constructive conflict arise, your instructor will expect the team to first make an effort to resolve issues internally. Beyond that your instructor will intercede as needed to help in resolving team issues. In extreme cases, if a particular student does not cooperate with, he/she may be **FIRED** from the team. In such cases the student will not have the opportunity to contribute to the project any more and will have to complete the project on his/her own.

Teamwork does not imply an 80/20 rule. Everybody is expected to contribute equally. The instructor may seek explicit **Individual evaluations** from team members at the end of every release. The student’s project grade may be increased/decreased by up to 50% depending on the individual evaluations and TA/faculty mentor evaluations. So make sure you actively contribute towards the success of this project.

**Project Artifacts**

Each team should have a version control repository up by week 2 or week 3. All project artifacts including Project Plan, Requirements Specification, Test Plan, Design document, etc. should be submitted through the repository. More details of version control repository and other artifact submission will be announced soon.

**Project Releases**

This project will be implemented using an iterative incremental process. Each team is expected to produce two project releases – **R1** during Week 11 or 12 (tentative) and **R2** during Week 15 or16 (tentative). The initial project plan should clearly specify the functionality being implemented in the two releases. The release plan should be approved by the mentor (project sponsor) and TA. A release includes the current state of all project artifacts created till that point including executable source code. Each release is accompanied by an in-class team presentation and product demonstration.

**Project Grading**

Your instructor will grade your project using the following point distribution. The course schedule states the expected project deliverables. Project grades are assigned on a team basis with each member initially receiving the same grade. Your instructor will reserve the right to adjust individual grades up or down after each of the project releases based on team member peer evaluations and the instructor’s observations.

Total Points Available - 250

* Status reports & Other Project Deliverables (120 points total)
* Release 1 (55 points)
* Release 2 (75 points)

The assessments provided by each mentor will be taken into consideration for each of these components. So make sure you satisfy your client (faculty mentor) requirements!!!

**Policies & Miscellaneous Items**

1. We have approximately 27 scheduled sessions. Missing one/two sessions can make a significant difference. Most sessions may have some activities (some graded/ some not graded). In addition to grades being lowered as per the university policies, you will also miss credit for activities that took place during the classes you were absent. Bottom line: Be a part of the class physically as well as mentally.
2. No make up exams/assignments/activities will be given. Exceptions may be granted for extreme cases as per university policies.
3. Be professional. Practice common courtesy when I or your fellow students are speaking.
   1. Avoid colloquial talk
   2. Do not distract other students with chitchats
   3. No cellphones/email/IM/Tweeting/browsing/game play or use the computers in any other unprofessional manner
4. You are encouraged to actively interact with your fellow students in class. However, Plagiarism of any kind is a strict **NO**. If the content is deemed as plagiarized/copied, the person who has copied the content will get a straight **F** for the course.

**Other Resources**

**UML Tools:**

**1.**  [**http://modeling-languages.com/uml-tools/**](http://modeling-languages.com/uml-tools/)

**2.**  [**http://www.sereferences.com/uml-tools.php**](http://www.sereferences.com/uml-tools.php)

**Design Patterns:**

**1.**  [**http://st-www.cs.uiuc.edu/users/patterns/**](http://st-www.cs.uiuc.edu/users/patterns/)

**2.**  [**http://www.vincehuston.org/dp/**](http://www.vincehuston.org/dp/)

**3.**  [**http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html**](http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html)