**Discussion date: 19 Nov (Boston time)**

**Next sessions:**

**Daily Scrum Slack (Daily except Sundays): 10am -1015am**

**Daily Scrum Zoom (Sundays): 10am – 11am**

**Confirmation from Richard to meet every other Monday starting 11/15 starting at 11:30-12:15**

**Sprint 2**

**Daily Scrum updates:**

**Yesterday progress**

**Bing:**

**Soheil:**

**Mitchell:**

* **Review CI/CD pipeline**

**Atil:**

**Taryar:**

**Today progress**

**Bing:**

* **Generate 10x unit tests**
* **Fix contact page with real number**

**Soheil:**

**Mitchell:**

* **Continue CI/CD pipeline**

**Atil:**

**Taryar:**

* **Update GIthub README when Bing completes unit tests and fix contact page**
* **Upload sprint 2 minutes to Github minutes**

**Impediments:**

* **Mob programming requirement difficult to achieve during weekday**
  + **Resolution is to meet on Saturday or Sunday**

**Other discussions (post daily scrum):**

**Project part 3: Second Sprint**

| Project part 3: Second Sprint | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| Sprint Planning: You have a forecast of how many story points your team can complete during the sprint. The forecast is documented in your README. |  | 1 pts |
| - The rationale for your forecast is documented in your README. |  | 1 pts |
| - You pulled stories into your sprint backlog. You pulled stories from the top of your product backlog. The aggregate size of the stories does not exceed your forecast. Developers participated in this activity (and in your project README, you made a statement or provided evidence that only Developers participated in this activity). (1 point for each item) |  | 4 pts |
| - All of the stories in your sprint backlog are the right size: the size of each backlog item is less than half your forecast velocity for the sprint. If you noticed PBIs that are too large to get done in a sprint, you refined them by splitting them into smaller stories and estimating the smaller stories. |  | 1 pts |
| - You decomposed user stories into developer tasks (1 point). The tasks are clearly listed in your sprint backlog (1 point). |  | 2 pts |
| - Your sprint backlog, including both the user stories and developer tasks, is represented in a kanban board (1 point). The URL of the kanban board is documented in your README (1 point). Course staff can view the kanban board (1 point). |  | 3 pts |
| - You have a sprint burndown chart. On the x-axis, you display time markers in units of 1 day. On the y-axis, you display story points remaining to get done. There is a linear curve descending from left to right; the leftmost point of the curve is number of story points in your sprint at sprint day 1; the rightmost point is 0 story points remaining on the last day of the sprint. The burndown chart, or the URL of the burndown chart, is document in your README. Course staff can view the burndown chart. (1 point for each sentence you complete) |  | 6 pts |
| Daily Scrum: You have conducted multiple Daily Scrums. You document evidence of at least one daily scrum in your README. |  | 1 pts |
| - In your daily scrums, you discussed: as a group of Developers, what did you do in the last 24 hours that helped you meet the Sprint Goal? In your README, document which team members did which activities. (1 point for each team member's response, up to a maximum of 5 points) |  | 5 pts |
| - In your daily scrums, you discussed: as a group of Developers, what will you do in the next 24 hours to meet the Sprint Goal? In your README, document which team members plan to do which activities. (1 point for each team member's response, up to a maximum of 5 points) |  | 5 pts |
| - In your daily scrums, you discussed: do you see any impediment that prevents the Developers from meeting the Sprint Goal? What are the impediments? What is your impediment removal plan? (1 point for each question answered) |  | 3 pts |
| - Every day during the sprint, update your sprint task board and burndown chart. Provide evidence (URLs or images) in your README to show that you did this. (1 point for each time you updated your tracking indicators, up to a maximum of 2 points) |  | 2 pts |
| Pair- or Mob-Programming: There is evidence that you paired or mobbed on your code. The evidence could be a photo or video of your team working together, or it could be some other kind of evidence. The evidence is included in your README. (1 point for each team member who participated, up to a maximum of 5 points) |  | 5 pts |
| Test-Driven Development: There is evidence that you are building your product test-first. There are at least 20 micro-scale unit tests in your test suite, and they all pass. (1 point for each \*new\* unit test that you created in this sprint, up to a maximum of 10 points) |  | 10 pts |
| Sprint Review: Your team conducts a Sprint Review. |  | 1 pts |
| - Your product increment is working software. The working software is running on a publicly accessible system (this could be a web server, a mobile app store, or something else.) Your README includes evidence that your product increment is working software. Your README includes the URL of the working software, an invitation to test the mobile app, etc. (1 point for each sentence in this rubric element) |  | 4 pts |
| - At least one stakeholder attends your sprint review, and you provide evidence that your stakeholder was there. You revise your product backlog based on the feedback you receive. (1 point for each sentence in this rubric item) |  | 2 pts |
| Continuous Integration: You have a Continuous Integration system running. You only work on the main/trunk/master together—there are no long-lived code branches. The CI system automatically builds your code every time you push to main/trunk/master. The CI system automatically executes all your tests every time it builds the code. You have provided evidence that your CI system exists and behaves properly. (1 point for each sentence in this rubric element) |  | 5 pts |
| Continuous Delivery: You have a Continuous Delivery system running. When the build is "green", the CD system deploys your software to a production environment ("Production"); when the build is "red", the CD system doesn't alter Production. The CD system executes additional tests of your software in Production to ensure Production is up and running successfully after deployment. You have provided evidence that your CD system exists and behaves properly. |  | 5 pts |
| Sprint Retrospective: Your team conducts a Sprint Retrospective. All team members participate in the sprint retrospective. As a team, you identify at least one helpful change to improve your effectiveness together. You make a concrete plan for making that change during the next sprint. (Provide evidence. 1 point for each sentence in this rubric element.) |  | 4 pts |