1.

*You have started work as an* ***agile coach*** *with a team who follow the Scrum process. The team comprises Lesley (product owner), Kerry (a senior developer) and four developers, Robert, Jody, Jamie and Alex. The team conduct a* ***sprint planning meeting*** *on* ***Friday afternoon*** *after a review meeting in the morning. You are invited to participate as an observer.*

*Lesley begins the meeting by congratulating the team on the work they got done in the previous sprint and briefly summarises the* ***five user stories that were successfully delivered****, including one that she* ***wasn't aware of*** *as it wasn't included in the previous sprint plan. She then summarises the* ***new requirements that were identified by the customer*** *during the sprint review meeting that took place that morning. The team then work on turning these into user stories. This takes about* ***60*** *minutes of discussion resulting in* ***eight new stories*** *added to the backlog of work. A typical example of one of the stories proposed is "As a user I want to be able to* ***edit photos*** *I take directly on the app so that I can create content all in one place."*

*Based on the product roadmap, there are* ***already five user stories*** *assigned to the coming sprint. Lesley chooses* ***five of the new user stories*** *for inclusion as well and labels them as* ***Must Haves*** *(giving 10 in total for the sprint). Lesley takes the remaining new user stories and divides them between the* ***next three future sprints****, according to their lower MoSCoW priority.*

*The team then plays a round of Planning Poker for each story. As time is short and the team member's estimates vary considerably, Lesley* ***just assigns each story an estimate of one week****, so that all 10 can be completed within the coming sprint.* ***Lesley allocates two stories to each of the developers****. The team then spend a further* ***90*** *minutes independently* ***identifying and recording the tasks*** *that will need to be completed in order to implement the user stories. Lesley makes sure each team member makes a personal note of their tasks.*

*Kerry reminds Lesley about their discussion in the retrospective that took place after the review meeting. The team had expressed unhappiness with the number of times* ***commits to their mainline resulted in broken builds*** *when the code was subsequently deployed. Lesley agrees and leads a discussion of the reason for this, which lasts about* ***30*** *minutes. The team decides that they need to* ***configure a continuous integration service*** *to build the project every time a commit is made to the mainline. Lesley* ***wants to be supportive*** *so agrees and adds this to the* ***sprint backlog*** *as well.*

Problems and their solutions:

* Problem: The customer seems to have very little influence on which user stories are done in which sprints, Lesley being the one who prioritises the 5 requirements from the 8 given
  + Solution: Involve the customer in the sprint planning meetings as well to learn their priorities as it is their product that the team is working on.
* Problem: The user stories are allocated to developers only by Lesley, possibly not taking into account the developers’ skills and desires
  + Solution: Have a discussion on which user stories cater more to each team members’ skills and which ones they would prefer to do
* Problem: Assuming all sprints are of similar length, 10 user stories are chosen to be done in the coming sprint in contrast with the 5 done, which possibly means double workload and incomplete tasks
  + Solution: Prioritise tasks to be done in this sprint
* Problem: User stories are all estimated to be the same by Lesley instead of trying to figure out why there is such a discrepancy between team members’ estimates of the tasks
  + Solution: Devote more time into planning poker or another estimation exercise instead of giving up. If there is no more time left in the day, start most immediate tasks but plan another meeting to discuss estimation of the tasks.
* Problem: There is no discussion of the specific schedule of the tasks to be done.
  + Solution: Have all team members set individual deadlines for the tasks and establish a week-by-week schedule.
* Problem: The problem of broken builds being deployed is never addressed, only its fix is found.
  + Solution: Have a team discussion on how viable making a CI service would be and whether it would save time and errors in the long run. If found to be important, prioritise the CI service over some less important feature user stories.
* Problem: There is no discussion about how the work done in the previous sprint would influence work for the next one
  + Solution: Discuss as a team how accurate the estimations for the tasks were in order to more accurately pinpoint what types of tasks are more appropriate for which team members and how quickly and effectively they can do them. Any problems encountered before could also be relevant for the next sprint.
* Problem: No theme set for the coming sprint
  + Solution: Establish a theme based on the types of user stories worked on for the customer to better understand what is being done.
* ------------------------- Kuba’s points:
* Problem: VERY LONG meeting, not including review and retrospective
* Problem: “Friday afternoon”
* Problem: “wants to be supportive”
* Problem: 3 requirements discarded

Objectives of sprint review meeting:

* Compare tasks done (features and bug fixes) with the ones agreed at the previous meeting
  + Comparison should justify any missed and additional tasks
* Explain any problems encountered
* Provide a demonstration of the current product
* Deliver the current product to the customer

Sprint planning meeting:

* Discuss in a detailed manner key new requirements
  + New features from the review meeting
  + Any problems to solve in the future
  + Priority of the tasks and problem fixes
* Agree on a plan for the next sprint
  + Propose a plan (with estimated times of the tasks and their priority)
  + Modify the plan according to customer’s desires
  + Include time to fix any current problems or implement important quality improvements (CI/CD pipeline, etc)

Roles in a customer meeting:

* Meeting chair
  + Manages the agenda
  + Keeps the meeting to time
  + Constrains discussions
* Product owner (may combine with chair)
* Lead demonstrator
* Note Taker
* Checker

“Requirements influence design decisions and the availability of pre-existing frameworks, and libraries constrain the feasibility of requirements.”

3.

*A team is working on an application to* ***automate*** *the process of conducting a* ***systematic literature review*** *(SLR). An SLR is undertaken when a researcher wants to* ***identify and analyse*** *all the* ***research papers*** *published in peer reviewed conferences and journals* ***about a particular topic****. The review is called systematic because the process should (in principle) be described in a way such that the set of papers identified can be* ***reproduced by independent researchers****.*

*An SLR typically follows several different stages. First, an exploratory search is undertaken in an online paper database with associated API, such as the ACM Digital Library, to identify relevant keywords. For example, for an SLR on Behaviour Driven Development the keywords might be "Behaviour Driven Development", "BDD" and "Gherkin". Then the database is searched fully, and all returned results are recorded, usually in a format like BibTeX. The* ***PDF version of the files*** *will also be retrieved. These results may then be* ***filtered*** *by date range, and publication type, since some databases will return non-peer reviewed documents, such as conference introductions. Any* ***duplicate******results*** *will then be* ***removed****. Next, irrelevant papers are* ***filtered****, normally by identifying irrelevant venues. For example, a paper on the topic of gherkins in the Journal of Applied Ecology is unlikely to be relevant to a review of BDD. The* ***filtering*** *process is then repeated for relevance of the paper title and abstract.*

*Once the set of papers have been identified, a number of* ***analyses*** *may be performed. The researchers may want to produce a* ***graph*** *showing the number of publications per year identified for the topic. The researchers may also* ***classify*** *the papers according to research method such as case study or laboratory study. Other metadata may also be extracted, such as the number of participants involved in a laboratory study, or whether research questions are formally stated. Sometimes it is useful to* ***reorder*** *these steps, for example, by counting the number of papers per year before and after an exclusion criterion has been applied.*

Architecture: Pipe and filter (again!?)

Diagram:

Online Paper DB -O )- Search by Keywords -O )- Date -O )- Publication Type -O )- Duplication Remove  
 I I  
 O   
 U O  
 I I  
Search for Keywords (sent to Search by Keywords)

-O )- Relevancy -O )- Analyses -O )- Requester

Reasons:

* filtering research papers based on different criteria directly translates to using architecture with Filter components on information flow
* modifications on the architecture allow implementing specific features mentioned in the requirements

Modifications:

* *Pull* data flow because the data does not change drastically enough to keep a continuous flow of all repositories and their changes. This way, filters can be more effective because they use output from other filters more without worrying about the original data source.
* Concurrent pipeline? Researchers can see some research papers immediately to start analysis on them, papers can be searched
* Reorderable filters to allow researchers to customise them as required. Except keyword search, which cannot be reordered.
* Branching filters to allow more potential analysis of the papers by the researchers and for different outputs from the filters, including graphs and formats (PDF or BibTeX)