Comments

Q1. This compulsory question proved more of a challenge for students than I expected. There were many excellent answers, demonstrating a deep understanding of the agile principles named in the question and the BUFx approaches we studied in the lectures. But there were also a great many very poor answers. Common problems were:

- Not reading the question: a surprising number of candidates seemed determined to answer the question I had asked on last year’s exam, and gave lists of agile practices that corresponded to these principles as their answer. Others thought I was asking whether the use of BUFx approaches violated the principles or not, but failed to say why they had come to the conclusion they reported. Still others used their time to explain to me how much better agile approaches were than BUFx—which was nice, but it wasn’t what the question asked for.

- Too simplistic an understanding of BUFx: a number of candidates talked about BUFx in very strong terms, saying things like “no interaction with the customer is allowed with BUFx” and “once the requirements are documented, no changes are allowed” and even “when even a small requirements change comes along, we have to start the whole process again and scrap everything we have done so far”. (These are minor paraphrases of actual answers.) I could award very few marks for these answers, because they displayed on the whole a lack of understanding of how BUFx approaches work, rather than the clear grasp on the concepts I’d been hoping for.

- Not saying enough to allow me to give full marks: despite the questions being worth around 2 marks each, many candidates made only one point in their answers, meaning I could award them only 1 mark. Some candidates managed to make their one point several times in the same answer, meaning that I could only award 1 mark even though they had written quite a lot of sentences.

Some additional notes on the individual questions are given below.

A) This was perhaps the easiest of the principles to tackle with respect to BUFx, and many people gave excellent answers. After the general points listed above, probably the most common cause of lost marks was in failing to cover all three points from the principle: early delivery, continuous delivery and valuable software. I accepted good coverage of any two of these for full marks, but there were still a number of candidates who managed only to cover one of them.

B) The main cause of lost marks for this question was in claiming that BUFx violates the principle by not allowing changes to requirements. No real approach to software development is going to get away with allowing some degree of change. The real problem here is the cost of change when it does happen.

C) Some candidates stated (rather surprisingly) that no face-to-face communication ever happened in a BUFx project. Others said sensible things about the amount of face-to-face communication with the customer that would occur on such a project, but then asserted roundly that the developers never met each other or had any discussions. A third common cause of error in this sub-question related to stand-up meetings. Some candidates focussed their whole answer on stand-up meetings, ignoring all other kinds of face-to-face interaction that might occur on a project; they then struggled to make any kind of concrete link from stand-ups to BUFx. (Obviously, there’s no reason why a team using BUFx couldn’t meet every day for a quick chat about progress…)

D) Many candidates gave good answers to this question, covering the issue of “maximising the amount of work not done”, but very few managed to convincingly link simplicity to BUFx. Otherwise good answers that failed to mention simplicity earned 1.5 marks out of 2.

E) This was perhaps the question that was answered least well, with candidates demonstrating several major misconceptions about retrospectives and what they are for. Despite my efforts during the course unit, it seems there are still a large number of candidates who are confusing planning meetings (i.e., meetings which make changes to the plan) with retrospectives (meetings which make changes to the team’s process). The other most common cause of lost marks was candidates’ claims that BUFx didn’t allow meetings between the developers for reflection.

Q2. This was the most popular question. All but a handful of candidates answered it, and all but a handful of those who did answered it well.

A) This question aimed to check basic understanding of a range of agile practices. Almost everyone was able to name four such practices, although a small number of candidates were a little confused about what was a practice, what a methodology, and what an underlying concept. Probably the most common cause of lost marks was in failing to be specific enough in the description of the practice. For example, several candidates described a practice that involved something like “the team meeting together to discuss how the project could be made better”, and hoped for the answer “retrospectives”. But, several other practices also fit this description. I would, in this case, have given a mark of 1, rather than 2, to take account of the ambiguity.

B) There was some good analysis of the problems going on in the answers for this part of the question, and some very sensible suggestions for how to address the root cause of what is going wrong in each scenario. Marks were lost for answers that read too much into the question. For example, a couple of people referred to the code from team A as being “buggy”, when the question itself gives no indication that this might be the case. (Perhaps these candidates were getting confused with the scenario for team B?)

For the suggestions for corrective actions, I awarded most marks for answers that suggested some concrete change in the behaviour of the team that went right to the heart of the problem as diagnosed. A number of candidates seemed to be rather cross with Team B and its boss. There were several calls for the boss to be fired (!)
Q3. This was the second most popular of the optional questions, being answered by approximately two thirds of customers to easily access the core service offered by the company. being able to send details of discounts was important, without explaining why it was more important than allowing support of the candidate’s preferred release. For example, saying that B was more important than A, because The most common cause of lost marks in answers to this question was vagueness in the arguments provided in dependence of story B on story A, in the absence of any other online services being offered by the company. Several candidates did talk about technical dependences between the stories, when in fact there are none, since I was careful to make them all end-to-end slices (although only a few of a stories were thin end-to-end slices). The most common cause of lost marks in answers to this question was vagueness in the arguments provided in support of the candidate’s preferred release. For example, saying that B was more important than A, because being able to send details of discounts was important, without explaining why it was more important than allowing customers to easily access the core service offered by the company.

Q3. This was the second most popular of the optional questions, being answered by approximately two thirds of candidates. There were many excellent answers, and overall a good understanding of user stories and business value was displayed.

A) Most candidates managed to gain the 4 easy marks from this question. Only a few of you will be surprised to hear that I did not award marks for vague roles such as “user” and “customer” (unless the meaning of these roles was clarified by the sentence of description that followed). Similar roles that I deemed too vague to be worth marks (without clarification) were: manager, office manager, business manager, admin staff, director and accountant. There were also a couple of answers that listed “developer” as a user role, which of course received no marks.

B) This question was answered significantly less well, largely due to an inaccurate reading of what the question was asking for (a change in behaviour that the client wanted to see). A number of candidates listed software functionalities that the selected user role might want to have, giving a technical change rather than a domain-based behaviour change (e.g. “enter details of deliveries in the database” rather than “have fewer missed-delivery attempts”). Some candidates suggested behaviour changes with very low business value, such as having secretaries enter data more quickly. (These candidates were perhaps influenced by the stories in question 2 on this, overlooking the fact that those stories may have been provided as poor examples rather than as something to be copied. Students revising for exams in future years should take note!)

C) There were many great stories written in answer to this question. Almost no one provided a story that was not an end-to-end slice, and many candidates supplied solid business values for their stories. You were helped by the provision of a set of “template” user stories in question 2, and many of you borrowed shamelessly from them to create your own stories for this question (overlooking, as mentioned above, that some of the stories in question 2 might have been poorly written on purpose). I did not penalise anyone for copying from the stories in question 2 where the result was an appropriate and sensible story for the context given in question 3.

The commonest mistakes related to business values (unsurprisingly). There were quite a few stories where the business value given was actual a function (usually the next function step in a process). For example, one story was “I want to be notified when a delivery is made so that I can send the payment request to the customer’s bank”. As a simple rule of thumb, if you could take the business value from your story and write a new one with that same phrase (or a close paraphrase) as the function, then you have probably not written a very good business value.

Another typical error was to write a business value that simply restated the function (e.g. “I want to track my order so that I know where my order is at any time”). Some stories had values where I couldn’t see the relation to the function (e.g. “I want to see the list of missed deliveries so that I can choose which deliveries to buy”). Some had bad functions that were expressed in overly technical language that was unlikely to match how the customer thought about their requirements (e.g. “As a warehouse operator, I want to enter my work into a database, so that …”). In particular, there were a number of stories from candidates who seemed to have a bad case of Excel snobbery: “I
want to enter my data into a proper database, not a spreadsheet". (There was quite a bit of talk about “proper”
databases—as though most non-computer scientists care in the least about these things.) Stories containing these
types of basic error received few, if any, marks. Candidates who attended the lectures will not be surprised to hear
that stories aboutlogin functionality also did not score well.

D) This question was the least well answered of all parts of this question, although again many candidates gave an
excellent analysis for both estimates. The most common error was failing to remember that the justifications for
both estimates had to be compatible with one another. For example, a typical justification for the low estimate on a
data entry type story was that the database into which the data must go already exists. This would be followed by a
justification for the high estimate that was based on the non-existence of the database. The database could not
both exist and not exist, so it was impossible that these two facts could be used as the justification for both
estimates in a single game of Planning Poker.

Some candidates misread the question, and selected two different stories, one to be the low estimate story and one
the high. A worrying number of candidates also seemed not to know what an estimate in story points represented
(an estimate of the effort, or “size”, of the user story). Instead, they seemed to think that story points were a
measure of the relative value of the stories, and used this question as an opportunity to waffle vaguely about value.
(These candidates appeared to have adopted the technique of memorising answers to last year’s paper, and to be
repeating those here, regardless of what I had actually asked. Students reading this for revision in future years,
please take note!)

The point of this question, of course, was to check understanding of the collaborative “wisdom of crowds” approach
to estimating that agile methods advocate. It is possible for two team members to have very different estimates of
effort while both being completely correct in their analysis, since they will be coming from different viewpoints.

Q4. This was the third most popular of the optional questions, and was answered by around a quarter of
candidates. Everyone who chose to answer this question seemed to have a solid grasp of the principles being
tested. All answers were good, and most earned first class marks. There was only one candidate who tried to write
production code in their fixture, which is a big improvement on previous years.

A) Most candidates were able to suggest something that was both concrete, and cheap to apply in the early stages
of the project. Paper prototyping was the most popular suggestion (the answer I was primarily looking for). Other
sensible suggestions did earn marks, too, when they were well justified. A couple of candidates said they would talk
to the HR staff, to reassure them, which did not earn many marks (unless backed up by a more concrete action).

B) Many candidates gave perfect answers to this question. A few candidates included the Applicant ID column
unnecessarily (including one person who, surprisingly, included it as an output column – which demonstrated
originality, at least). A few others used two columns to give the percentage increase/decrease in execution time
against the baseline, which earned slightly fewer marks than the most minimal solution with just 3 columns. With
respect to the test cases themselves, most solutions were excellent, with just a few candidates suggesting non-
redundant tests or using non-boundary values in their columns.

C) Again, there were some good answers given. Almost everyone remembered the basic structure of decision table
fixture classes well, although there were a handful of candidates who struggled with aspects of how the output
columns should be connected with. The main cause of lost marks here (and it was not a significant one) was in the
production code designs that people suggested, some of which were plausible and some of which looked too
contrived when viewed as production code alone. But, all in all, this question was very well answered.

Q5. Only a small handful of candidates braved this TDD question, though all who did coped reasonably well.

A) Not everyone got this question 100% correct, but it was for the most part answered well.

B) This sub-question was also answered well. The main cause of lost marks was in slightly odd choices for the first
test. The best starting point is the test that converts 0 quacks. Other tests would be reasonable if we were not
doing TDD, since we were, we needed to choose the tests in an order that would help us to drive out the equation
that converts quacks to ducks as we code. For this, we need the “base case” of the equations, which will give us a
key constant after refactoring, for later use.

Other marks were lost for odd suggestions for refactorings, though there were not many of those.

C) This question was answered well, although some candidates seemed to run out of time (or ideas, or both). The
only noteworthy point about the answers is that several candidates said they would refactor the if-else structure into
a HashMap after 2 or 3 tests. I gave some marks for this, though not full marks, as the HashMap only serves to
obscure the equation we are trying to drive out. No one managed to come up a refactoring step that would lead us
in the direction of driving out this equation.