COMP3211 Software Engineering

Assignment 1

Author
Wang Yuqi
[redacted]

Lecturer
Prof. Max Yu PEI

BsC in Computer Science
Academic year: 2025/2026 Sem 1

Q1: Software Processes (8 marks)

Q: Please 1) briefly describe one software project you worked with before, 2) decide whether a plan-driven approach or an agile method would be more appropriate for the project, and 3) justify your decision. (< 150 words)

ANSWER:

Description: I once worked on a 3D indoor scene synthesis project. It involves a graident-based optimization method to generate 3D layouts.

Choice: An agile method would be more appropriate for this project.

Justification: The purpose of agile development is to *reduce overheads in the software process (e.g., by limiting documentations) and to be able to* ***respond quickly*** *to changing requirements.* Therefore, an agile method is more fitting, since this project involves rapid iteration through different ideas.
A plan-driven approach, on the other hand, would not only **stale progress** but can also be extremely **short-sighted**, as it is impossible to know beforehand if current ideas will work, and whether new ideas will emerge later on.

Q2: Manifesto for Agile Software Development (12 marks)

Q: Agile software development values individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. Can you think of four situations, one for each pair of these items, where focusing too much on the first item while too little on the second item could lead a software development team to trouble? (< 150 words)

ANSWER:

1. **↑ Individuals & Interactions; ↓ Processes & Tools**

   Case: Develop ML model for sales prediction. No PR code review processes. No tools like `W&B` or `MLFlow` for logging.
   - **Pitfall 1**: New team members cannot reverse engineer design choices, because all commit messages are "fix model" or "tweak params".
   - **Pitfall 2**: New team members cannot reproduce the results, because there are no intermediate results to compare against.

2. **↑ Working Software; ↓ Comprehensive Documentation**

   Case: Developing frontend for a Web3 app. PM requested fancy UI with 3D graphics. The frontend team implemented from scratch with OpenGL. No documentations were written.
   - **Pitfall 1**: As codebase grew, frontend team members' memories of earlier design choices became vague. Development friction increases exponentially.
   - **Pitfall 2**: The frontend team left company. Nobody was able to maintain the codebase. The team had to revert to using legacy UI.

3. **↑ Customer Collaboration; ↓ Contract Negotiation**

   Case: Customer requires a ML solution for fraud detection. No contract negotiation on quantitative results. Overreliance on-site customers' user stories.
   - **Pitfall 1**: The team delivered a solution that fits the grand expectations and narratives by achieving high *Recall* and *AUC-ROC*. But the customer later realized what truly matters is *Value-weighted Recall* (fraud dollars captured), causing major disagreement.

4. **↑ Responding to Change; ↓ Following a Plan**

   Case: A product team developing B2B analytical platforms. The product team defies rigid plans and embrace changes. Their entire product is built out of customer feedback loops.
   - **Pitfall 1**: The eventual product experience is not cohesive. Modules felt disjoint and performance was lacking. Customer is unsatisfied with the delivery despite meeting all requirements on paper.
   - **Pitfall 2**: The team cannot quantify progress and productivity as there is no predefined roadmap to refer to.

Q3: 3. Agile Methods (12 marks)

Ayed et al. investigated the cultural challenges in applying agile methods in Europe and Asia[1] . Pick two challenges in applying agile methods in Asia that were mentioned in the paper and intrigue you, and then 1) briefly describe what the challenges were, 2) indicate where in the paper the challenges were introduced, and 3) explain what you would suggest Asian development teams do to rise to the challenges. (< 300 words)

[1] H. Ayed, B. Vanderose and N. Habra, "Agile cultural challenges in Europe and Asia: insights from practitioners," 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Buenos Aires, Argentina, 2017, pp. 153162, doi: 10.1109/ICSE-SEIP.2017.33.

ANSWER:

- **Challenge 1**: Lack *Team Empowerment*
  1. Description: Asian workers tend to have *command-and-control* mindset and doesn't like to take initiatives and make decisions. The paper also mentioned that they *don't dare to think out of the box* even when granted freedom.
  2. Location: Section IV.B (*Team Empowerment* under *Results*)
  3. My Suggestion: It might be that team members are not used to taking ownership of their work due to education system and thus lack the confidence to take responsibilities. I suggest starting by delegating low-risk tasks to members, provide encouragements, and gradually increase the complexity.
- **Challenge 2**: Lack *Transparency*
  1. Description: The paper states *"team members have a tendency to not expose problems such as non-feasible deadlines or technical difficulties"*, which can be attributed to the high PDI which causes uneasiness with superiors.
  2. Location: Section IV.C (*Team Transparency and Cohesion* under *Results*)
  3. My Suggestion: Team leaders should build a sense of psychological safety between them and team members. For example, they can hold "blameless retrospectives" that focuses on team members' progress not results. By doing so, team members are encouraged to speak up and be transparent about the progress, instead of fearing blame from incomplete results.

Q4: Customer Involvement (10 marks)

One of the problems of having a user closely involved with a software development team is that they "go native." That is, they adopt the outlook of the development team and lose sight of the needs of their user colleagues. Suggest three ways in which you might avoid this problem and discuss the advantages and disadvantages of each approach. (< 150 words)

ANSWER:

Approach 1: Periodically replace on-site user with another user.

- Pros: By limiting interaction duration, this prevents the user from adopting to the team's outlook. Furthermore, diverse perspectives prevent overfitting to one customer's needs.
- Cons: Constantly switching on-site user may cause disruption and inefficiency. Each new user needs time to learn and adapt to the team's tools and processses.

Approach 2: Have multiple on-site users.

- Pros: Constant diverse perspectives without the hassle of switching users.
- Cons: Due to the static nature of this approach, users are still subjected to influence of the team's outlook, though less than single on-site customer.

Approach 3: Have the on-site user representative hold regular meetings with their user colleagues.

- Pros: Grounds perspectives on the users' outlook. Since the user colleagues are not directly involved in the development, they are not subjected to team influences, hence less bias.
- Cons: Constant channeling of message between user colleagues and the development team through a single (or a few) user representative(s) cause latencies.

Q5: Requirements Specification (8 marks)

Suppose you and your friends plan to implement a virtual 3D map mobile application that helps visitors navigate through the PolyU campus, and you are responsible for developing the requirements document for the new system. Based on your experience with existing map applications, please

1. Define one functional user requirement in natural language for the new system's "route suggestion" function. (< 50 words)
2. Define one non-functional user requirement in natural language for the new system regarding its reliability. (< 50 words)

Note: 1) Use "shall" for compulsory requirements and "should" for desirable requirements; 2) Be careful with the differences between functional and non-functional requirements; 3) Make sure requirements are verifiable; 4) Provide an explanation in parentheses if necessary.

ANSWER:

1. ***Route Suggestion* Function**

   The user **shall** be able to input any two locations on campus, and the system **shall** determine the shortest route between them. The system **should** be able to predict the starting location based on GPS location, and **should** also suggest an alternative *accessible route* for the disabled.

2. ***Reliability* Requirement**

   The system **shall** have downtime of $< 1\%$ (e.g., maintainance and software upgrades) around the clock. The system **shall** have an uptime of 100% during peak hours (i.e., 7AM-9PM on Monday to Friday).