3. Project Communication: Solutions

3-1 What is the difference between a role and a participant?

A participant is an individual. A role corresponds to a set of responsibilities. A single participant can assume many roles. Some roles may be shared among more than one participant.

3–2 Can a role be shared between two or more participants? Why or why not?

Some roles are assigned to more than one participant. For example, a team of developers can be assigned to the same subsystem. Other roles, such as chief architect or project manager, can only be assigned to a single participant in the project. In the end, the issue is one of assigning clear and non-overlapping tasks to single participants: When two or more participants are responsible for exactly the same task, none of them will feel personally responsible for the success of the task.

3–3 What is the difference between a client and an end user?

The client contracts the development of a system. The client is responsible for defining the scope of the system and for financing the project. The end users are the participants who actually uses the system to accomplish their work. In some project, the role of end user and client may be shared by the same participant. In general, however, this is not the case.

3–4 To which roles would you assign the following tasks?

- *Change a subsystem interface to accommodate a new requirement.* [system architect]
- *Communicate the subsystem interface change to other teams.* [API engineer (liaison)]
- *Change the documentation as a result of the interface change.* [editor]
- *Design a test suite to find defects introduced by the change.* [tester]
- *Ensure that the change is completed on schedule.* [project manager or team leader]

3–5 You are responsible for coordinating the development of a system for processing credit applications for a bank. In what roles would the following project participants be able to contribute most to the project?

- *a bank employee responsible for processing credit applications* [end user]
- *the manager of the information technology group at the bank, who contracted the system* [client]
- *a free-lancer who developed similar systems in the past* [developer]
- *a technical writer [editor]*

3–6 Draw a UML activity diagram representing the meeting process described in Section 3.4.3. Focus in particular on the work products generated before and after the meeting, such as the agenda and the meeting minutes. Use swimlanes to represent roles.

Figure 3-1 depicts an example solution. This exercise tests the student's knowledge of the meeting process and of activity diagrams described in Chapter 2.

3–7 What is the difference between a work package and a work product?
A work package is a description of work. A work product is the actual result of work (e.g., a document, a piece of code, etc.).
When is a work package defined?
During planning.
Consider an assignment where two students collaborate to plan and develop a system for sorting lists of names using two different sort algorithms. The deliverables for the assignment are the source code, the system documentation, and a manual for other developers explaining how new sorting algorithms can be integrated into the code. Give examples of work packages and work products in this project.

Examples of work packages:

- Plan development

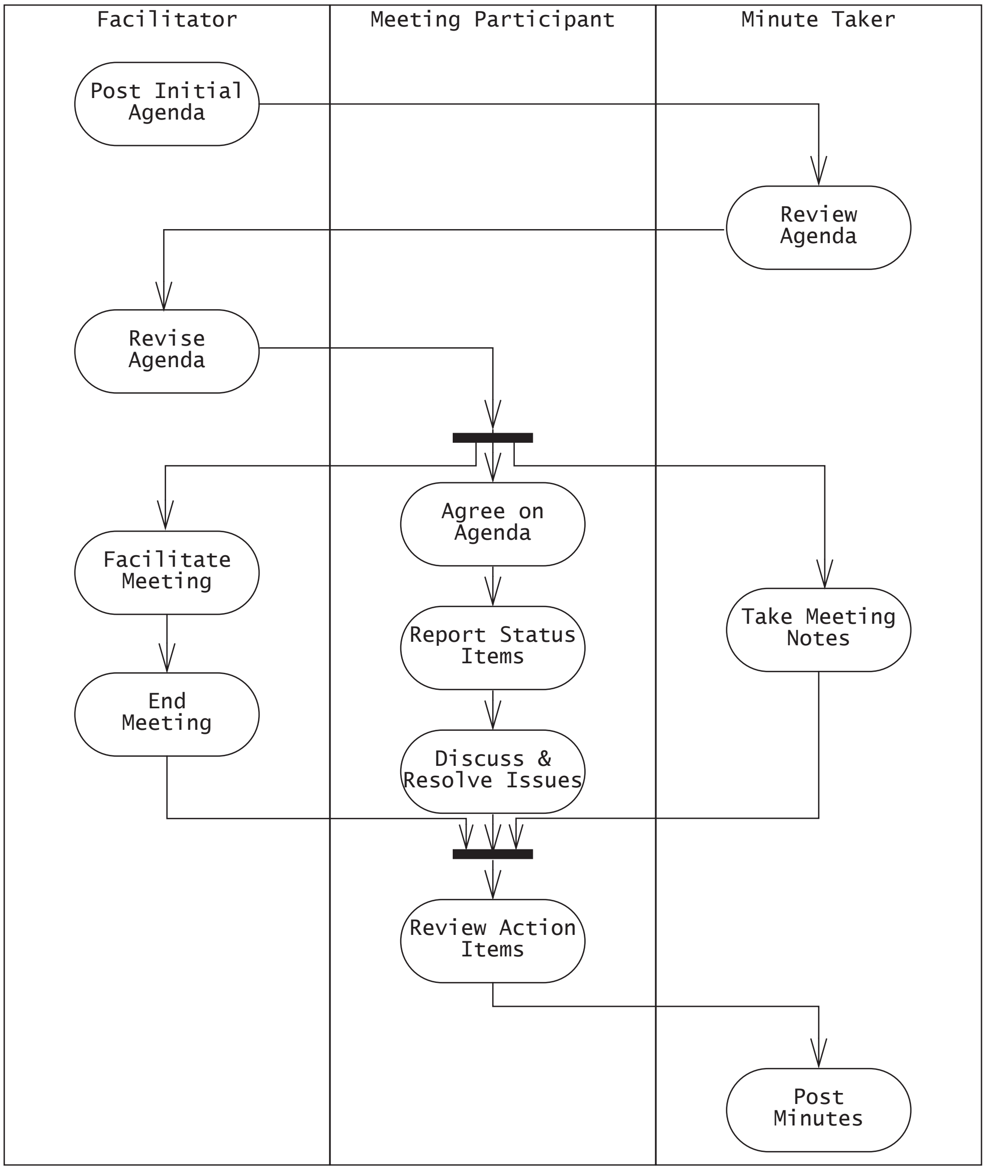

Figure 3-1 An example meeting process (UML activity diagram).

- Define and document interface between system and sort algorithms
- Implement algorithms
- Document system

Examples of work products:

- development plan
- source code
- system documentation
- manual for explaining how to integrate new sorting algorithms

3–8 *What is the difference between a cross-functional team and a subsystem team? Provide examples and justify your choices.*

A subsystem team is completely responsible for the development of a subsystem.A cross-functional team is made out of liaison of subsystem teams and is responsible for support tasks, such as documentation and configuration management. Also, the system architecture can is often defined by a cross-functional team made out of liaison from all subsystem teams.

3–9 *As many critical communication events are planned (e.g., client reviews, project reviews, peer reviews), why is there still a need for unplanned communication events (e.g., request for clarification, request for change, issue resolution)?*

There are many reasons for unplanned communication. Some of the more important ones are:

- Reduce latency: While developers could wait for the next project review to request clarification on their tasks or on interfaces provided by other teams, this would slow down development significantly and introduce much rework (e.g., correcting code that was developed under incorrect assumptions).
- React to unexpected events: Any system of a realistic size cannot be completely specified and understood a priori. This results in requirements and design changes during development, which often have to be communicated and negotiated among several stakeholders. While change processes can be set in place to manage this communication, much unplanned and informal communication complements the formal communication events to motivate, clarify, understand, gauge, and negotiate the underlying issues introduced by the change.
- Team building: Formal communication events are rarely conducive to a team atmosphere. It is during unplanned and informal events that participants get know each other and form a community working towards a common goal.

3–10 *Select a random working day in your work week. Log all activities that qualify as communication activities (e.g., talking to friends over coffee, obtaining information from a fellow student, providing information, negotiating, advertising, browsing the web). Which fraction of your working day does communication represent?*

Typically, the fraction of the day spent of communication can easily reach 50% or more. However, there will be a wide variance in the numbers reported by the students, depending on whether or not they are currently doing project work.

3–11 *You are a member of the user interface team. You are responsible for designing and implementing forms collecting information about users of the system (e.g., first name, last name, address, E-mail address, level of expertise). The information you are collecting is stored in the database and used by the reporting subsystem. You are not sure which fields are required information and which are optional. How do you find out?*

This is an open-ended question whose purpose is to have the student think of all the communication paths available. The side effect of this question is that the student will also have to think about the information sources associated with a project. A correct answer should contain at least two of the following points:

- The user interface team member should first check the requirements analysis document, described in Chapter 1.
- If the answer is not in the requirements analysis document, or if the answer is not clear, the document is incomplete or does not exist yet, in which case the user interface team member should ask the responsible analyst on the project. This is a request for change or a request for clarification.
- If the analyst does not have the answer, he or she should investigate with a potential user and possibly get approval from the client. This is also a request for clarification.
- In all cases, the answer should eventually make its way into the requirements analysis document.

3–12 *You have been reassigned from the user interface team to the database team due to staff shortages and replanning. The implementation phase is well underway. In which role would you be most proficient given your knowledge of the user interface design and implementation?*

Liaison of the database team to the user interface team.

3–13 Assume the development environment is Unix workstations, and the documentation team uses the Macintosh platform for writing documentation. The client requires the documents to be available on Windows platforms. Developers produce the design documentation using FrameMaker. The documentation team uses Microsoft Word for the user-level documentation. The client submits corrections on hardcopies and does not need to modify the delivered documents. How could the information flow between the developers, the technical writers, and the client be set up (e.g., format, tools, etc.) such that duplication of files is minimized while everybody's tool preferences and platform requirements are still satisfied?

This is a simple real life question with complicated answers. Several approaches are possible, including:

- The project uses a printable format as a standard interchange format, such as Postscript or PDF (Portable Display Format). This allows every participant to access and view documents. Only the author of a document is able to make changes, however. This makes it difficult to move the document responsibility to move from one team to another (e.g., from a subsystem team to the documentation team).
- The project uses an interchange format such as RTF or XML for all documents. Both formats are recognized by recent versions of both Microsoft Word and FrameMaker. This allows document responsibilities to be transferred between developers and technical writers. Unfortunately, this is currently not a practical solution as both tools understanding of the interchange formats is different and imperfect, resulting in some loss of formatting.

3–14 Which changes in the organization and communication infrastructure would you recommend for a successor of the Ariane 5 project as a consequence of the Ariane 501 failure described at the beginning of this chapter?

This is an open-ended question as many solutions exist and detailed knowledge of the current team organization are not available. Note that this question is only interested in communication or organizational changes, not technical or methodological changes. A correct answer could include the following points:

- Increase communication between the team responsible for system testing and the software developers. This can be done by introducing liaisons on both teams.
- Increase communication between the team responsible for unit testing and for system testing.
- Change documentation requirements for reused components.