

Software Engineering Group Project

COMP2043.GRP

Session 02:
Work Organisation

This Session

- Some notes on team working
 - (Partly based on slides by Prof. Dave Elliman)
- Group meetings
- Software Development Methodology



Team Working (1)

- Teams can be fun!



Team Working (2)

- But sometimes they don't work well . . .
 - Inadequate organisation
 - Low commitment
 - Apathy
 - Conflicts



Characteristics of teams that work

- Balance of member skills
 - and making good use of those skills
- Clear goals
- Clear responsibilities
- Good organisation
- Good communication, including listening
- Commitment to goals: willingness to put group goals before ego and/or comfort
- Mutual respect and valuing



Necessary Roles

- Motivator (initiator)
- Idea generator
- Team worker (“getting the job done”)
- Specialist (technical, writing, . . .)
- Coordinator (administrator)
- Censor (devil’s advocate)
- Mediator (supporter, mentor)
- Monitor (tester)
- Completer-finisher



Formal Roles (1)

- Every team should elect a Team Leader:
 - Overall planning and coordination
 - Motivator
 - Arbiter
 - Main point of contact



Formal Roles (2)

- Additionally, the following are highly recommended:
 - Editor
 - Technical Lead
 - Quality Assurance Lead
 - Repository Master
 - Other useful roles:
 - UI Designer
 - Open Day Producer



Formal Roles (3)

- Editor responsibilities:
 - Document structure
 - Layout (creates templates)
 - Structure of writing process (e.g. draft deadlines, organisation of proof reading)
 - Integration of contributions



Formal Roles (4)

- Technical Lead responsibilities:
 - System architect
 - Identify key technical choices, pros and cons
 - Lead programmer (as projects are not too large)



Formal Roles (5)

- Quality Assurance Lead responsibilities:
 - Making sure requirements are testable
 - Planning for quality assurance, in particular testing
 - Writing test cases
 - Automation of testing, in particular regression testing



Formal Roles (6)

- Repository master responsibilities:
 - Overall responsibility for managing project site and repository
 - Training everyone in how to use the site and associated tools
 - Project website deadline



Formal Roles (7)

- Note:
 - Not all roles relevant all the time
 - Roles can be shared/further subdivided
 - One person can have more than one role
 - Role owners should not be expected to do all work associated with role
 - Rather, think “organiser”
 - Role owners not exempt from helping out with other aspects!
- Everyone should pull their weight all the time!



Some tips

- Be positive: see challenges, not problems
- Work on the assumption that every team member really wants to do his or her best
- If someone does not contribute effectively, try to find out why, and what can be changed to help
- Attempt to handle conflicts within group, but ultimately, don't be afraid to ask supervisor or module convener for help



Group Meetings

- Two kinds of meetings:
 - Informal meetings
 - Only the group members
 - Usually one per week, more if necessary
 - Formal meetings
 - All group members plus the supervisor
 - One meeting per week, about 30 min
 - Compulsory!
- If you cannot make it, apologise to the meeting chair well in advance!



Informal Group Meetings (1)

- Purpose: coordination and getting some real work done
 - Typical activities:
 - Develop a group-wide understanding of what the project is and a consensus about its aims
- Organisational matters:
 - electing group leader
 - division of work
 - developing time plans
 - developing work procedures



Informal Group Meetings (2)

- Design discussions
- Discussions about specific technical problems
- Reviews and inspections:
 - design documents
 - reports
 - code
- Prepare for the formal meetings



Informal Group Meetings (3)

- Of course, a lot of work needs to be done outside meetings, individually or in small subgroups; e.g.:
 - Background research
 - Detailed design
 - Writing design documents and reports
 - Coding
 - Testing & Debugging
- Remember: If you don't put in on average 9h/week, you are not working hard enough!
 - (Coffee-breaks not included. :-))



Formal Group Meetings

- Purposes:
 - Formally monitor progress by reviewing minutes from preceding formal meeting
 - Formally take major design decisions
 - Formally decide on what should be done over the next week, and who is responsible
 - Keep supervisor informed about where the project is going
 - Seek input from supervisor
 - Discuss problems



Chairperson and Secretary

- There should be a Chairperson (or Chair) and a Secretary for each meeting
- These roles should rotate within the group
- The Chair organises and leads the meeting
- The Secretary records what happened and what was decided during the meeting in the minutes



The Chairperson

- The Chair runs the formal meeting:
 - Prepares a written agenda prior to the meeting, makes it available to the group and supervisor (via project site and/or e-mail), and brings printed copies to the meeting
 - Leads the meeting by following the agenda
 - Ensures that the meeting remains focused



The Secretary

- The secretary records the meeting:
- Takes notes during the meeting:
 - Who are present & apologies
 - Summary of major points
 - All decisions
- Compiles these notes into minutes immediately after the meeting
- Makes the minutes available to all group members and the supervisor. They then check that the minutes correctly reflects the meeting



Minutes (1)

- A written summary of a meeting is called the minutes of the meeting
- The minutes help keeping the work organised and focused
- The minutes should be archived:
 - using the facilities for sharing documentation through the project site is a good idea



Minutes (2)

- The minutes should record:
 - Date, time, and place of the meeting
 - Chair, Secretary, who is present
 - Apologies from those who are absent
 - The main points discussed during the meeting
 - All decisions
 - All action points
 - Date, time, place, Chair, and Secretary of the next meeting



Action Points

- Each meeting generates a list of action points. Three parts:
 - What the task is
 - Who is assigned to the task
 - When the task should be finished
- The purpose of the action point list is to:
 - provide a clear and concise record of the work that needs to be done
 - ensure that tasks are not forgotten
 - make it easy to ensure an evenly-distributed workload



Example of Action Points

- John: Find a good Visual Basic Book. Done by: 11 Nov (next group meeting)
- Mark and Sarah: Fix the “sorting bug”. Done by: 8 Nov (urgent)
- All: Finish interim report chapter drafts. Done by: 18 Nov

It can be helpful to clearly identify particularly urgent action points to help ensure they get priority.



Structure of the Formal Meetings (1)

- Typical agenda:
 1. Opening of the meeting
 2. Apologies
 3. Review of progress since last meeting.
 4. . . .
 -
 - n – 2. Any other matters
 - n – 1. Next meeting: Date, Chair, and Secretary
 - n. Closing of the meeting



Structure of the Formal Meetings (2)

- Progress review: be sure to follow up on all outstanding action points
- Review of old action points and other discussion will generate further action points. Record them (e.g. on white board)
- Be sure to review all new action points towards the end of meeting to ensure everyone knows and understands what their tasks are



Personal Logs

- In addition to the formal meeting minutes, it is useful to keep your own personal log
- The log can be used to:
 - keep track of your tasks
 - record how your time is spent
 - note down any ideas you have
 - The log is very useful to
 - organise your own work
 - in group meetings
 - when writing the individual reports



Software Development Methodology

- You can use any appropriate methodology
- Agile methods have been found to work well in the context of the group projects
- Be sure to use prototyping!



Why prototype?

- GRP is a difficult module for many reasons:
 - Large, unstructured task
 - New application domain
 - Medium- to large-scale software development will be a new experience to many of you
 - New people
- Prototyping can help with these issues!



How can prototyping help?

- Helps understand the problem domain and the key difficulties:
 - extremely valuable design input!
- Gives insights about how to best structure the implementation:
 - helps large-scale software development
- Gives something concrete that everyone can try out and have opinions on:
 - ensures everyone is on the same page and pulls in the same direction

