Project logs

The main reason why you might want to keep a project log is to support you in your work, or a specific project that you are undertaking.

A project log would benefit from recording at least three types of information:

1. Factual information.
2. Development of ideas, plans and designs.
3. Reflection on experiences.

A log is an aide-mémoire that helps you remember facts, ideas, plans, data, and sources.

The log is for you, perhaps to share with colleagues, other members of your team, or in the case of courses, to show to your tutor.

Hohmann description of how to use a project log:

- Keep your notebook with you at all times.
- Use your project log.
- Write legibly.
- Review your notebook regularly.
- Keep accurate dates.
- Keep it safe.
- Use abbreviations if you wish.

What to include ?

You can use the log to:

- make notes, sketch designs, and explore ideas;
- practice writing descriptions;
- reflect upon your experiences.

To ensure that your entries at the log are complete enough, you should consider making many of your entries conform to **the journalist's five Ws** strategy:

- **w**ho (is involved),
- **w**hat (happened),
- **w**hen (did it happen),
- **w**here (did it happen)
- **w**hy (or how).

This strategy is useful because it generates specific answers to these questions.

When recording information that you have read, you should aim to write no more than 150–200 words summarizing the important points.

When you summarize an information that you have read, you should record the following:

- the author (or anon.),
- the work's title,
- the publication date (if the date is not known record n.d.),
- the chapter heading or pages if appropriate,
- the publisher and ISBN,
- or for websites the full URL and the date you accessed the site.

What not to include ?

A project log should not:

- act solely as an appointment book or general diary;
- be written on loose scraps of paper, or in a variety of different places;
- include everything that happens during your day or week, whether relevant or not;
- wander from the subject it was set up to record;
- be a polished piece of writing: it should ‘record as found, when found’.

There are two basic ways in which you can keep a project log: on paper or electronically.

Paper log:

Advantages:

- durability
- convenience
- ease with which you can make an entry.

Disadvantages:

- the difficulty of making backup copies of pages and the difficulty of finding information again.
- pages can easily fall out in a loose-leaf folder.
- pages cannot be added in a bound journal-style notebook.

Electronic log:

you can use a word processor, a diary program or keep your project log online (using a blog or the Open University’s MyStuff facility)

Advantages:

- the ease with which you can copy-and-paste information between software applications.
- you can search and backup your log very easily,
- correct erroneous entries,
- check your spelling,
- and not have to worry about writing legibly.

Disadvantages:

- their availability.
- the permanence of the data.

Laboratory notebooks

the most common form of project log is the laboratory notebook.
laboratory notebooks can be accepted as legal proof of the date of an invention.

Blogs

advantages:

- Entries are time- and date-stamped;
- they can be tagged with keywords
- they are searchable

disadvantages:

- entries in a blog can be edited or amended after they have been posted.
- a blog is intended to be read by anyone and everyone.

MyStuff

- MyStuff is an online personal storage area provided for you by the Open University .
- you can use it to store files or to record notes (much like a blog)
- MyStuff is private so only you have access to it, although you can choose to share items that are stored within MyStuff with your tutor or other students if you wish.
- One way of keeping a project log in MyStuff would be as a series of notes, creating a new note for each new entry in your log.
- you can tag each of your notes with keywords to make it easier to search your log.

Working at a distance

The classical view of a team is that it is a group of people who have shared goals and a common purpose.

Frequent, effective communication is a critical success factor for a virtual team.

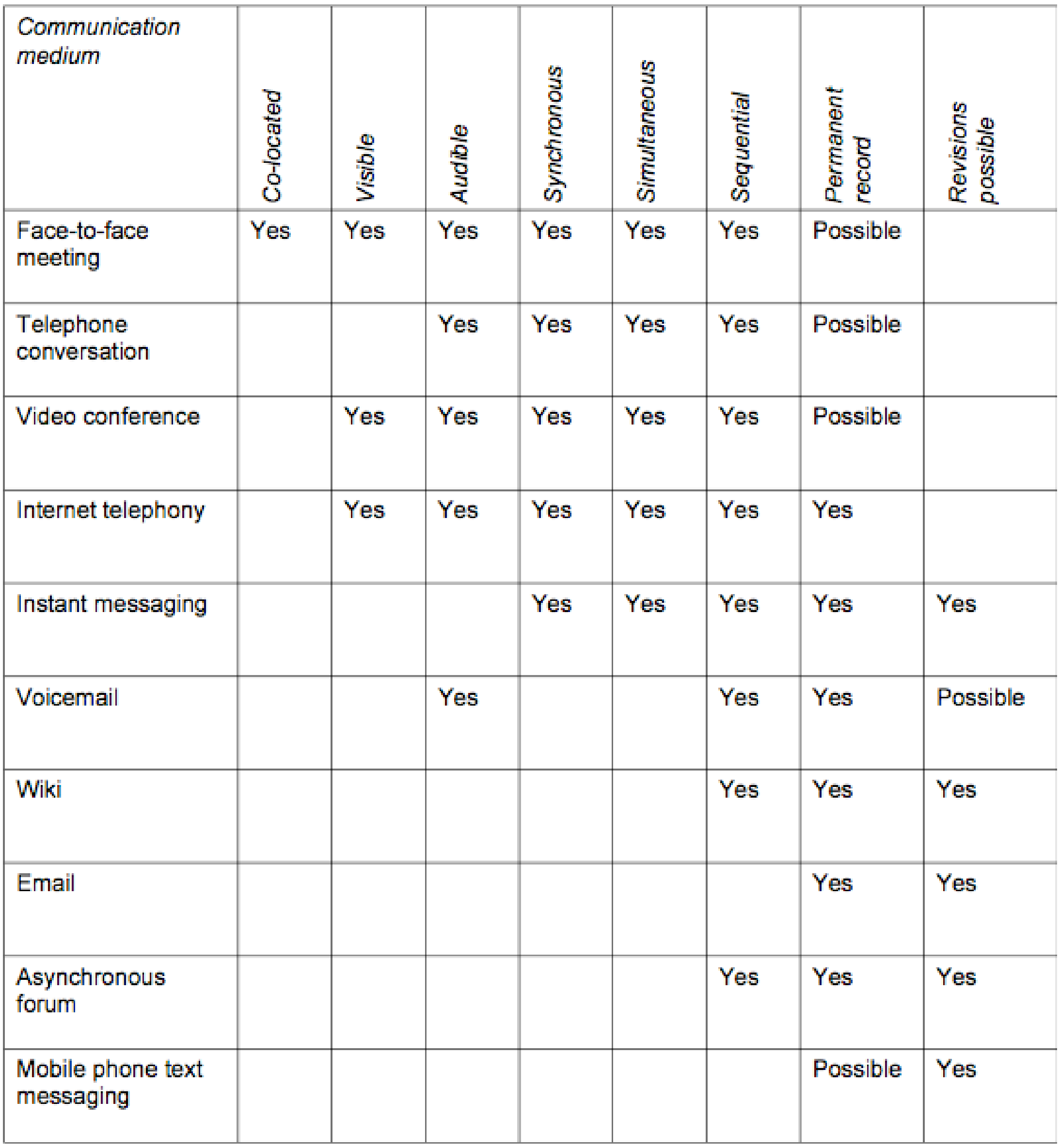
Lipnack and Stamps identified three important practices to the **success of the virtual teams**:

- Exploit diversity within the team.
- Use technology to simulate reality and bring people together in the virtual realm.
- Work on holding the team together through frequent communication.

To work at a distance you need:

- Partitioning the work and keeping the team informed of your progress.
- Collaborate by using asynchronous communication media.
- Be patient of waiting for someone else to complete their part of the project.

Characteristics of various communications media according to the dimensions:



Communication medium	*Co-located*	*Visible*	*Audible*	*Synchronous*	*Simultaneous*	*Sequential*	*Permanent record*	*Revisions possible*
Face-to-face meeting	Yes	Yes	Yes	Yes	Yes	Yes	Possible	
Telephone conversation			Yes	Yes	Yes	Yes	Possible	
Video conference		Yes	Yes	Yes	Yes	Yes	Possible	
Internet telephony		Yes	Yes	Yes	Yes	Yes	Yes	
Instant messaging				Yes	Yes	Yes	Yes	Yes
Voicemail			Yes			Yes	Yes	Possible
Wiki						Yes	Yes	Yes
Email							Yes	Yes
Asynchronous forum						Yes	Yes	Yes
Mobile phone text messaging							Possible	Yes

Part-time study and team working

- you cannot expect that your fellow team members are available and studying at the same time as you are, or even that they are studying at the same time each day or week.

- it can be more difficult for you to schedule synchronous electronic meetings between team members than it might be for a 'full-time' team.

issues that your team will have to be aware of:

- The difficulty of scheduling synchronous meetings.
- Recognizing that progress may be slow because you are all studying part-time.
- Allowing extra time for decision making.
- Partitioning the work, so that you are not dependent on working very closely with remote team members on a difficult problem.

Examples of rules that the team could consider are as follows:

1. Frequently check e mail and team forum.
2. Summarize the result of communication in the forum.
3. Decide communication media.

one of the key factors for virtual team's success is frequent communication. The communication media that you choose to use don't have to be technically sophisticated – your team just has to use them!

Communicating, cooperating and collaborating online

There is two electronic collaboration tools at the M253 course's website:

1. A forum.
2. A wiki.

1- Forums:

Using message threads in a forum:

- Giving messages meaningful, unambiguous, subject lines can greatly enhance your ability to find your way through the many postings that appear in busy forums.
- Short titles are better than long titles because the subject line can be truncated within the forum interface so that the rest of the title does not show up.
- Sometimes, a discussion can move off the topic of the original thread, or several new ideas can be sparked off in the course of a discussion. If this happens, start a new message thread with a new subject line that is more descriptive of the content of the message.

What to use forums for ?

- Your forum is best suited to discussion and debate.
- Decision-making in forums is also possible; however it is best managed by setting time limits on the initial discussion and deadlines by which decisions should be taken.
- Asynchronous meetings can be an effective way of managing progress.
- The compilation of the team report that you have to prepare for each TMA and the ECA. you might find that this aspect of the team's working is best managed in the team wiki (see below).

Disadvantage of forums:

- the difficulty encountered in not being able to create shared documents.
- More time and energy in modification.

A paper by Ben Plumpton named 'How students can make conferencing work' have the following **advices**:

- everyone participates.
- the forum is supportive
- everyone takes initiative and acts as the facilitator sometimes,
- everyone writes carefully
- people trust each other.

2- Wikis:

- A Sandbox, is a page on the wiki that you can use for experiments and to try things out while you get used to working with a wiki.

- Comments on the content are identified with the author and date if you use the wiki comments facility.

- only one person can edit a section of a wiki page at any given moment in time. The system will lock the section automatically and not allow anyone else to access it until the current editor has signed off and released that section of the page.

- When you have added content to a wiki page, the system lets you preview what you have done before you save it permanently. So you must remember that you do need to go back and save it after previewing it, otherwise the system will just lose all your work.

Disadvantages:

1. contributions to the page do not get the author's identity, or the date of the contribution, automatically attached to them. However, there is evidence of who did what, when, in each wiki page's history, which can be found by using the History tab at the top of the page, but it is not very easy to follow.
2. you can delete work entered by other people.
3. only one person can edit a section of a wiki page.

Other collaboration tools:

There are two key points that you need to bear in mind before considering, or while using, other communication tools:

1. a summary of what you discuss, agree, and create must be relayed back to the forum or wiki so that everyone in the team and your tutor, knows, or is able to find out, what is going on.
2. you must obtain team agreement about the use of such a tool.

1- Instant messaging:

- instant messaging (otherwise known as online chat) is a form of synchronous communication.

- In some ways, meetings conducted by instant messaging are more flexible than face-to-face meetings. People who arrive late can catch up, you can take breaks from the meeting, and a transcript of what was said can be generated automatically. About the only things that instant messaging cannot do for you is make the decisions and write the minutes of the meeting – these still have to be done by the participants in the meeting!

2- Audio conference:

- It is a form of synchronous voice communication.

- In order to use internet telephony (or VOIP – Voice Over IP as it is sometimes called), you will need:
  1. a microphone
  2. a loudspeakers for your computer
  3. a reasonably fast internet connection.

3- Video conference:

- It is a form of a synchronous communications.

- Some video conferencing softwares:
  - Skype.
  - FlashMeeting:
    - Developed by the Open University.
    - It runs in a web browser, so there is no software to install.
    - It is a half-duplex system.

- In order to use video conferencing, you will need:
  - a microphone
  - a loudspeakers for your computer
  - a reasonably fast internet connection.
  - a webcam (web camera).

Duplex System (simultaneous)	Half-Duplex System
Allows more than one person to talk at the same time.	people have to take it in turns to talk.
Suitable for small meetings.	Suitable for large meetings.

Good practice when communicating electronically

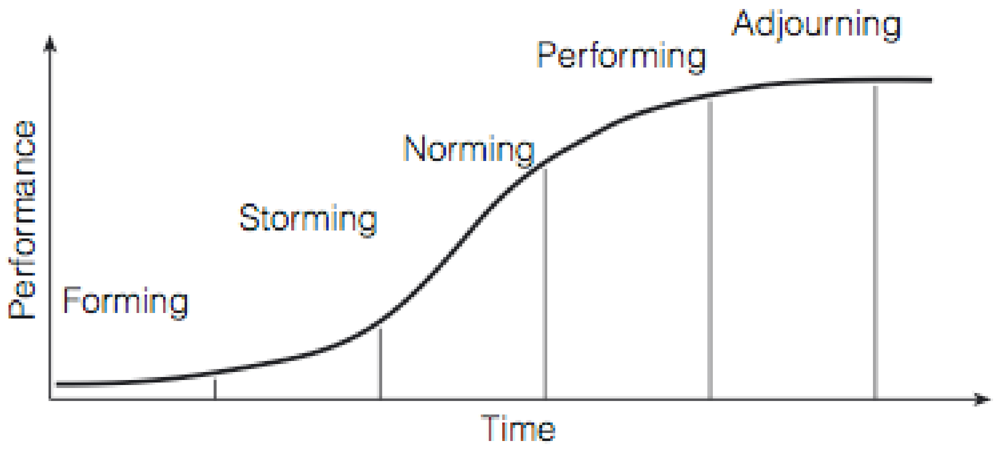
- Remember that you are talking to real people.
- Try to keep your messages short.
- Give your messages a descriptive Subject line.
- Summarize messages when replying to them.
- Check with the sender before quoting from private email.
- Credit other people’s work.
- Signal your emotions in messages.
- Avoid writing all in CAPITALS.
- Send your messages to the right place.

Team formation

Many teams appear to develop in the same way and to follow a predictable pattern of formation and growth. As a member of a team, if you know what the pattern is and can recognize the features of the developmental process in your own team, then:

1. you can understand what is going on – the growing pains of your team;
2. you can take appropriate action to help your team to move on to the next stage of growth;
3. you can try to avoid doing anything inappropriate to upset the development of your team!

Tuckman model of the development of teams:



1- Forming:

- Team members try to get to know each other.
- The team attempts to understand the task or project they have been assigned.
- Individuals within the team will be trying to work out what role they want to play in the team, and what roles they want other team members to play.
- the team will also begin to establish some rules by which the team will operate.

Advice for distributed teams:

- Individually, you may experience some anxiety and uncertainty over what the project involves, what the other team members will be like, and frustration over the slow start to the project. Often, there is a feeling of wasted time. But please be patient. It takes time for people to get to know one another, to begin working with one another, and to build up levels of trust with one another.

2- Storming:

- A feeling of lack of progress on the task can trigger this phase of team development.
- Sometimes, differences of opinion over the task can arise, and individual's personalities can clash as team members overcome their tentativeness and begin to assert themselves.
- Discussion often centers on team process issues, such as team rules and team roles.
- There may not be much progress on the project!

Advice for distributed teams:

- it is easy for members who feel that they are on the receiving end of unwarranted or

personal criticism to become defensive. this can lead to them withdrawing from the discussion until the debate has become less heated and less personal. In order to draw them back in, it may require a personal, private approach (via e-mail or a telephone call rather than in an open discussion forum) to bring them back into contributing to the team.
- it is important to remember the advice given elsewhere about how to communicate effectively (often called 'netiquette' – short for network etiquette).

3- Norming:

- The team should agree the ways of working and interacting with each other (team rules), and individual team members should agree to the roles they are going to take on in the team and the tasks they will perform (team roles).
- The team also reaches agreement upon the nature of the task and how it is going to tackle it.
- There is real progress upon the task that the team has been assigned.

Advice for distributed teams:

- A useful team rule to establish is the frequency with which team members should check the team conference and their personal e-mail for messages.
- It is also useful to establish when individual team members will be able to contribute to the team – their typical working patterns.

4- Performing:

- The focus of the team has shifted from team process issues to the task that the team is undertaking.
- Individuals within the team know how to work with each other and how the team as a whole operates.
- The team is in a much better position to solve problems.
- There is substantial progress towards the final project goals.

Advice for distributed teams:

- you lack many of the informal cues that are important to keeping a team working productively and easily together – such as how busy someone is, what they are working on, and when they might be on holiday, ill, or otherwise unable to contribute to the team. Therefore, in a distributed team, you have to give rather more explicit indications of how you are progressing in order to replace these informal cues.

5- **Adjourning** (some times called **Mourning**)

- The team may become even more effective as it makes a concerted effort to complete the task before the final deadline.
- there is a possibility that the team could become less effective as members regret the end of the task and the breaking up of relationships that have formed between the members of the team.
- The task is completed, the project comes to an end and the team disbands.

Advice for distributed teams:

- In order to overcome this stage, try to identify some means of moving on, of keeping in touch and of keeping working relationships going – perhaps by something as simple as agreeing to keep in contact by e-mail.

Tuckman's model of team development has a number of other **implications** and consequences of which it is useful to be aware:

- The duration and intensity of the different stages can vary between teams.
- Teams usually have to progress through the earlier stages of development in order to reach the performing stage.
- It is possible for a team to return to a previous stage of development.
- The stages may merge into one another or be repeated as issues recur or new ones emerge.

Team organization

How teams can be organized ?

1- The democratic team

Features:

- Decisions are made by consensus, or if a consensus cannot be reached, by a vote.
- all team members are considered to be equal, which means that the tasks that they undertake must also be considered to be equal.
- common team tasks are rotated between team members so that these tasks do not always fall to the same people.
- Often, democratic teams have no appointed team leader. However, team leader(s) may evolve informally or be recognized by people who are external to the team.
- it is important to ensure that someone within a democratic team does have an overview of the project and the team, even if he or she has no special decision-making powers within the team.

Advantages of democratic teams:

- they have high morale.
- they can be very productive.

Disadvantages of democratic teams:

- they can become inefficient or even break down if no consensus can be reached over critical issues. In such situations it may be necessary to look to an external (usually higher) authority to intervene and help in making the difficult decision.

2- The hierarchical team:

Features:

- one person is in charge of the team, there is a team leader or manager who makes decisions and carries some responsibility for the team.
- In some hierarchical teams, the team leader has a more managerial role, deferring to an informal technical leader on decisions that involve technical knowledge that the manager does not possess.
- Hierarchical teams are very common in large companies. In such companies, a team leader will often be appointed by company management.
- Decisions are made high in the chain of command (sometimes outside the team) and are communicated downwards to the team.

There are teams in which one person has greater technical skills than the other members of the team (in software engineering practice, such teams are often called chief programmer teams – the chief programmer usually being the person who has the most experience).

3- The ego-less team

- 'everyone does everything' team.
- everyone works on everything and no one has control of any of the tasks within the overall project.
- decision-making has to take place by consensus since there is no leader.

Team roles

Your team needs to agree on who is going to carry out which role. This need not be fixed for the whole duration of the project. You could, for example, rotate the roles so that everyone takes a turn and thus gains experience of the different roles (as in some forms of democratic team).

1- Team leader

Responsibilities:

- has responsibility for clarifying the aims of the meeting and its agenda.
- They should introduce each item on the agenda
- guide the discussion of the items
- summarise the discussion and the decisions taken.
- In a hierarchical team, the team leader also has a strong leadership and decision-making role to play, which includes the partitioning of tasks and allocation of activities.

2- Record keeper (Meeting secretary)

Responsibilities:

- keeps a record of what decisions have been taken, who is doing what, and the date of the next meeting.
- Circulates a summary of the meeting to the rest of the team in the form of meeting minutes.

3- Report coordinator (team secretary)

* This role could fall to the team leader.

Responsibilities:

- coordinating the production of team documents and reports through managing the different versions of the documents that the team produce.

4- Progress chaser

Responsibilities:

- monitor progress, ensuring that everyone is doing what they are supposed to
- ensuring that all the tasks that need to be completed by a particular date are on schedule before the deadline and have been completed once the deadline has passed.

5- Timekeeper

Responsibilities:

- monitors how long is spent on each item in face-to-face meetings and alert the team when the specified period of time has elapsed.

6- Technical director

- Such a person is usually the most technically knowledgeable member of the team.

Responsibilities:

- whose role it is to advise the team and particularly the team leader upon technical issues.

7- Process consultant

Responsibilities:

- advise on the process of working in a team.
- monitor the way in which the discussions are being conducted and advise other members of the team on ways in which these discussions could be improved upon.
- they should make positive suggestions on how the team (or particular individuals within the team) could improve their working practices and relationships with other team members.

8- Client liaison

Responsibilities:

- responsible for dealing with clients.

Note: you do not need all of these roles all of the time, and the same person may fill more than one team role.

Reflective practice

Kolb's **experiential learning cycle**, has four major phases:

1. You start with an experience.
2. You (replay and) interrogate that experience for meaning.
3. You abstract and generalize from the experience and generate a hypothesis about the meaning of what took place.
4. You test this hypothesis in new situations in order to confirm or negate the generalization. Then off you go again on a new cycle...

Three **major approaches** of reflective practice:

1. The experiential application of received theory – trying things out and observing whether what happens is what you would expect to happen.
2. Knowledge-in-action – of making knowledge explicit through observation and reflection. This separates the concepts of 'reflection-in-action' (thinking about it at the time) and 'reflection-on-action' (thinking about it later, setting aside time to think about it).
3. Reflection as critical enquiry in a more generic sense. This is based on questioning not just what happens but also the aims, values and purposes (goals) of the activity being undertaken and the potential effects on the environment (which includes people).

Hazzan and Tomakyo indicate that there are two major factors requiring reflection as a habit of mind in software engineering:

1. The issue of coping with complexity.
2. The issue of coping with (communicating with) colleagues.

So, we need to improve both:

1. our awareness of our own mental processes.
2. our awareness of the mental processes of others.

The reflective process itself:

1. **Before the activity**, we need to take some time thinking about such questions as why we are undertaking a given activity, what assumptions we are making about the skills needed to undertake the activity, how we intend to carry it out, what we hope to get out of it, and how much time and resources we expect to have to devote to it.

2. **During the activity** there is perhaps less chance of conscious reflection, but as we proceed we may find that our subconscious is active in parallel and that useful ideas occur to us as a result of what actually happens, which we had not thought of in advance of undertaking the activity.

3. **After the event** we have the opportunity to ask ourselves such questions as whether the products of the activity are of an acceptable quality, what went well and why, what did not go well and why, what did not go as expected and why, how well the time and resources needed for the activity corresponded to our predictions, and what we would do differently if we had to repeat the activity.

‘My Biggest Mistake’ (Hodges 1996), indicated **four major reasons for failure**:

1. Many people did not set clear goals, or they had too many competing priorities.
2. People did not handle the information available to them adequately – and this was by far the biggest category – They tended to ignore information that did not fit their suspicions or their preconceptions.
3. People tended to act on impulse rather than thinking before they undertook an activity.
4. People failed to observe what they were doing carefully enough, in the sense that they did not check whether what they thought happened actually did happen.

The need for evidence to reflect your practice:

1. based on both qualitative and quantitative data.
2. the collection, analysis and presentation in your project log of evidence.
3. refer to the facts to back up your feelings.

Reflection on reflection (meta-reflection level):

Make sure that the process of reflection itself is assisting you in:

- improving your skills,
- improving the quality of the processes that you are undertaking,
- improving the quality of the products you are developing.

Information gathering

What sort of information are you looking for?

At the very least you need to identify:

1. the potential users of the system to be developed
2. what the system's objectives are from the user's point of view.
3. what features would a user expect the system to provide, and how would they expect to access these features?

where can you go to get the information you need?

1. **Literature searches** (textbooks, journals, reports, case studies, etc).
2. **Evaluation of existing systems.** Accessing existing systems of a similar nature to the one you are investigating.
3. **Interviews**. identify a wide range of people then ask them for their opinions on the nature and scope of a solution to the problem that would be satisfying to them is beneficial.
4. **Questionnaires**. They involve asking individuals to respond to a fixed set of questions set out in a fairly formalized document.
5. **Focus groups**. The idea of getting a group of people together to discuss the problem area. A group discussion often provide a mechanism for getting everyone to think a little more widely than they would have done if operating only as individuals.
6. **Observation**. observing the interactions of few people access a system and noting down what happened.

How to decompose a problem

- In focusing on problem identification you must steer a course between being too general and imprecise about the world in which the problem exists, and being too specific and precise about potential solutions before you understand the problem.

- Another aspect of identifying the problem to be solved is to ensure that you consider it from the point of view of the users of your system.

- If you do not identify the right problem initially then you may build software that correctly solves the wrong problem.

- you can begin to think about how you might proceed to form and describe possible solutions to the problem after:
  1. you have reached the situation where you think that you have understood the overall real world situation and the problem that you are being asked to solve.
  2. you have elicited the real needs of all the stakeholders that you have identified in the problem domain.

- In order to manage the complexity of a problem. you need to decompose the original problem into a collection of simpler sub-problems, each of which can be handled — and hopefully solved more simply — on its own.

Problem frames:

1- The Required Behaviour frame:

- There is some part of the physical world whose behaviour is to be controlled so that it satisfies certain conditions.
- The problem is to build a machine that will impose that control.
- Example: a controller for a set of one-way lights to manage the traffic flow at some road works.

2- The Commanded Behaviour frame:

- there is some part of the physical world whose behaviour is to be controlled in accordance with commands issued by an operator.
- The problem is to build a machine that will accept the operator's commands and impose the control accordingly.
- Example: the controller for your video player.

3- The Information Display frame:

- there is some part of the physical world about whose states and behaviour certain information is continually needed.
- The problem is to build a machine that will obtain this information from the world and present it at the required place in the required form.
- Example:the speed and distance-travelled information provided on a car dashboard display.

4- The Simple Workpiece frame:

- a tool is needed to allow a user to create and edit a certain class of computer processable text, or graphic objects, or similar structures, so that they can be subsequently copied, printed, analysed or used in other ways.
- The problem is to build a machine that can act as this tool.
- Example: a tool to create and update information on an individual's coffee purchases and tasting notes.

5- The Transformation frame:

- there are some given computer readable input(file)s whose data must be transformed to give certain required output(file)s.
- The problem is to build a machine that will produce the required outputs from the inputs.
- Example: a program to analyse the data relating to an individual's weekly supermarket shopping bills and to identify purchasing patterns, so that a set of special vouchers can be generated to encourage them to buy more of specific items.

there are no hard-and-fast rules that can be laid down for how to achieve a good decomposition or even how to recognise that you have achieved a good one. But there is some possible Heuristics:

- Identify the core problem.
- Identify ancillary problems.
- Use standard decompositions of sub-problems.
- Identify common concerns and difficulties.
- Look for sub-problems with different tempi.
- Look for sub-problems with different moods.
- Look for residual complexity.
- Investigate any need to model users.

- The purpose of the requirements definition phase of system development is to answer the very important, fundamental question: ‘What, exactly, is this system supposed to do?'

- a stakeholder need is: ‘a reflection of the business, personal or operational problem (or opportunity) that must be addressed to justify consideration, purchase or use of a new system.’

- a feature is: ‘a service that the system provides to fulfill one or more stakeholder needs.’

- a use case is: ‘the description of a sequence of actions, performed by a system, which yields a result of value to the user.’
  Consideration and elaboration of use cases moves you closer to your solution in behavioural terms, although it still keeps you away from the consideration of software requirements.

The systems approach

What is a system?

‘A system is a bounded entity which, in its environment, achieves a definite objective through the interaction of its component parts’.

1. **A system has a boundary**; it is finite and manageable.
2. **A system has an environment**; there is something outside that boundary. The system is usually affected by its environment, but the environment is not affected by events occurring in the system.
3. **A system has components** (subsystems); which interact with each other. A subsystem means grouping elements that appear related together physically or logically to deal with it.
4. **A system has an interface**; a set of rules for how a system interacts with the environment in which it operates.
5. **A system has structure**; characteristics of a system that can be divided into subsystems.
6. **A system has synergy**; it has an unexpected or unpredictable effects arising from the collaboration of its components.
7. **A system has a purpose** (objective); a reason for its existence, a set of defined objectives.

Checkland's Soft Systems approach

Professor Peter Checkland, defined soft systems as those in which:

- objectives are hard to define;
- decision taking is uncertain;
- measures of performance are at best qualitative;
- human behaviour is irrational.

Checkland's view was that ’problem solving is dependent on problem structuring'

Checkland's methodology attempted to provide – using systems ideas – a way of seeing diffuse, ill-structured problems in a patterned way, without distorting the original problem in the process.

Checkland's methodology phases:

1. gathering factual material and presenting it in such a way that it provides an overview of the underlying complexity of the situation.
2. we enter the system thinking world and construct a root definition of the system. This describes:
   1. what the system is
   2. what it aims to achieve taking into account the persons who could be affected by it
   3. defining the transformations that could be taking place and the environment that surrounds this particular (human activity) system’.

The root definition of a system should contain the aspects represented by the word CATWOE:

C	Client
A	Actors
T	Transformation
W	World view
O	Owner
E	Environment

a few questions to start you off on the process of system investigation:

Who is your system for?

- Who are the stakeholders? What are their roles? How do they perceive the system?
- stakeholders are individuals or organisations interested in, or affected by, the operation of the system, and their interests and priorities may not coincide.

What does your system have to do?

- clarify the purposes/goals/functionality of the system.
- What is the problem that the system is required to solve?

What is the context of the system?

- What is the environment within which the system will operate?
- What are the boundaries of the system?
- What other systems will it need to interact with?
- How, if at all, will the environment in which the system is deployed affect its use?
- Are there aspects of the environment – factors outside your control – that impose constraints on your system?
- Are there events arising from the environment that could cause the system to fail?