2.08 Problem solving frameworks

Introduction

Let's illustrate what we mean by a 'problem solving framework'. Here's a possible framework for planning and making a journey between two towns in the UK:

Consult map: find destination town, distance from start town, nearby major cities, etcetera
Investigate possible means of transport: compare... convenience, journey time, cost, choose preferred means of transport
Plan your journey Carry out your plan! (Travel to your destination)

This is quite a sensible high-level framework, but you would need to modify it and add different kinds of detail for each particular journey. For example:

- If you know where your destination town is, you probably won't need to consult a map. (Leave out this part of the framework.)
- If there's only one way of getting there you won't be able to 'compare' several, or 'choose preferred means of transport' but you'll still want to know the cost, and how long the journey will take. (Modify this part of the framework.)
- A plan for a train journey will be significantly different from a plan for going by car, or by air. (Add appropriate detail to this part of the framework.)
- and so on.

The fact that you need to modify the framework to suit each particular 'journey problem' doesn't mean it isn't any use. Most people do use something very like it when they make a new journey for the first time. They don't say to themselves, "I need to use a framework to help me solve this problem: I'd better look in a book, or write one for myself!" They have a journey planning framework stored in their heads, which they use when they need to. They are not really *conscious* of using it.

In the academic and professional worlds, however, understanding and consciously applying problem solving frameworks is commonplace, and essential.

When we speak of Chemistry, or Computing, or History, or Fine Art as 'disciplines', the word 'discipline' means exactly that. For example, a professional chemist does not behave like a child with a chemistry set: she does not say, "Let's mix up the contents of all these bottles and see what happens: ugh, what a horrible smell ... and I've burnt my eyebrows!"

Domain specific frameworks, and general frameworks

The framework suggested earlier for planning and making a journey is clearly not suited to solving problems in general. It is aimed at helping people to solve one particular class of problems, all to do with 'travelling from A to B'. We will call this a 'domain specific' problem solving framework, because it relates to a particular 'domain' or 'subject area'.

A 'general' problem solving framework, on the other hand, as its name suggests, should be applicable to a very wide range of problems, across different domains. The process of 'opening up and closing down', provides a generally applicable framework: it can be useful in all sorts of different situations across many different domains.

The words 'method' or 'methodology' are often used to describe domain specific frameworks: for instance, we speak of 'scientific method', and 'systems development methodologies'. The words 'method', 'methodology', 'framework' are not always used. For example Marxism, Structuralism and Modernism provide frameworks for thinking about history, the arts and society.

Academics and professionals, in every discipline area, use both domain specific and general problem solving frameworks. A major part of their education consists of learning to understand and apply appropriate frameworks.

Highly creative people often break away from the accepted way of doing things, and almost all of us feel a need to do that occasionally. We must be careful not to let the frameworks we habitually use stifle our creativity or that of the people we work with ... frameworks are there to guide us, not put us in chains. On the other hand, if as a student you wish to argue for rejection of accepted methods, you must be able to argue your case against 'tradition'. Since you cannot argue against something you don't understand, you have to learn to use the methods and frameworks of your discipline before you can make a case for rejecting them.

A general framework, developed by members of staff at theUniversityofHertfordshire. They've used an acronym, IDAPER, to name it: partly because this will help you remember the steps that make it up, and partly so that we have a short name to refer to it by.

IDAPER: A general framework for problem solving

The IDAPER framework identifies a sequence of six stages. We show these below, with the sort of questions need to be asked and answered during each stage. There is some additional explanation and advice on the following pages.

IDENTIFY	**Identify the problem** Is there a problem here that needs to be solved? (or an opportunity to be grasped?) Is it your job to solve it? (or is it best passed on to someone else?)
DEFINE	**Define the problem** What exactly is 'the problem'? What have you got to 'deliver'? What constraints must be met by any adequate solution? Against what criteria should solution(s) be evaluated?
(choose) APPROACH	**Investigate possible strategies, and choose an approach** Investigate: what strategies may help in tackling this problem? (Have you tackled a similar problem before? If so, what useful experience can you bring to this one?) Choose your approach: usually a combination of strategies.
PLAN	**Make a plan** How are you going to put your chosen 'approach' into action? How much time is available? What tasks need to be done? What resources are available? How much time needs to be allocated to each job? What order should the tasks be done in? Which tasks can be done in parallel with each other? When does each task need to be finished by? How are you going to monitor your progress in terms of ...... time-gone-by and amount of work completed? ...... the quality of the work that has been done? In addition, if this is a team project... How are you going to divide up the work? How often, and when, will the team need to meet? What will be the purpose of each meeting?
EXECUTE	**Execute the plan, to 'solve the problem'** (The plan may need modifying as you go along. If it does, make sure everyone involved understands and accepts the modified plan.)
REFLECT	**Afterwards, reflect on the process** What went well? Why? What went wrong? Why? How can you build on strengths, and deal with weaknesses, so that next time you do (even) better?

Some further notes on the IDAPER framework

Identify the problem

It is not always easy, and not always necessary, to separate 'identifying' a problem from 'defining' it, but it's worth having two separate steps for these in the framework. Sometimes we just have an uneasy feeling that there's a problem lurking about, or just around the corner waiting to pounce, or that we are missing some opportunity. We can't 'define it' yet, because we first need to 'identify it'.

Define the problem

By the time you have successfully 'identified' the problem, you may be well on the way to being able to 'define' it. Under most circumstances it is necessary, or at least highly desirable, to write down a definition of the problem. If you can't write it down, then you are probably still confused! The written definition should include the answers to all four questions listed for this step.

Choose an approach

Different problems are best approached in different ways. A general framework like IDAPER cannot tell you what approach to take to any particular problem you use it on. However, it can tell you that before you leap into trying to solve a problem you should think about different ways of going about tackling it, and decide on the best approach.

When you 'choose an approach' to solving a problem, what you are really doing is building (or perhaps borrowing) a more specific framework, one which is more 'domain specific' ... more suited to this particular kind of problem. You may need to combine a number of 'approaches', techniques, strategies (call them what you will) to put together your overall approach to your particular problem.

Past experience can be useful, but it can also be dangerous because it may close our minds to new approaches, or subtle differences between what we have experience of and what we need to do this time. Beware of people who tell you they know how to approach a problem because they've done one 'just like it' before: for example, producing a poster to advertise a gig for a pop group is not 'just like' producing a poster to communicate information about a technical subject.

Make a plan

Good planning is essential to the success of all but the smallest projects. It is particularly important if several people are involved, or you are working to a tight deadline, or you are simultaneously involved with other projects: as a student, often all of these will apply to you! A 'good plan' is one that 'works'. By carrying it out you achieve what you set out to achieve, with a minimum of hassle, stress and wasted effort. It's hard to know for sure if the plan is 'good' until you put it into action: so plenty of experience is needed before you can be confident of planning projects well.

The part of the planning process that is usually done least well is planning the use of time. It is difficult to estimate how long an individual task will take, let alone a whole project consisting of a number of different tasks. Most of us underestimate how long tasks will take. This is one reason why a good piece of computer software is so often shipped with a scrappy, unhelpful user manual: the project team just ran out of time.

Here are some hints to help with planning the use of time.

- Make a list of all the tasks that need to be done to complete the project. Start off by listing major tasks (such as, 'gather information'). Then try to break each of these down into smaller sub-tasks, and (if necessary) break these down again:

it is much easier to estimate how long it will take to do a smallish task than a big, complicated one.

- A plan for work on a group assessment is not 'a good plan' if you get 100% for this assessment but fail one of your other courses as a result! Your plan needs to take account of all the responsibilities you have: your other studies, your job if you have one, your social and family responsibilities.

- Allow time to deal with the unexpected. You, or another team member, may fall ill. The lab you need may be closed for a day at a critical point. Don't plan to finish on the submission date! ... plan to finish well ahead of the deadline. If all goes smoothly, you will be able to have some well-deserved time off.

- Plan to complete an adequate ('passable') solution within half to two-thirds of the available time: plan to add the 'bells and whistles' that will turn it into a top-notch solution after you've got something passable. Often students get obsessed with one area of a project, and fail to produce essential things in other areas ... and they fail, even if what they have done is very good.

Sometimes, when we start planning, we may not be able to break down major tasks into appropriate smaller tasks, because we won't be able to see what those 'smaller tasks' are until we get further into the work. In this case, the best thing to do is to make a high level plan for the whole project, and a detailed plan only for the first few tasks. Play safe: leave plenty of time for major tasks you haven't broken down yet, because you are likely to get your time estimates wrong for these. When you've carried out most of the detailed part of your plan, plan the next few tasks in detail. This is known as having a 'rolling plan'.

Execute the plan

It's no good having a plan unless everyone involved is committed to carrying it out. In team projects, this commitment is crucial: it will make or break the project. Even if you are working on your own you may have motivation problems to deal with.

Every problem is different, so we cannot tell you how to 'execute your plan'. However, there are some important points that do apply generally.

Monitor progress against your plan. Keep checking to make sure you are up to date with the work. How often you check will depend on the size and length of the project.

Check the quality of each deliverable, or part of a deliverable, as it is produced. Make sure it meets all constraints, and measures up favourably against the agreed evaluation criteria.

(Note that, on team projects, it is much easier to ask someone to put in more work on a particular task if you have agreed beforehand how each deliverable will be evaluated. It should then be clear to everyone, including the person who did the work, if it has not been done to the agreed standard.)

You are 'on target' only if the work is both of acceptable quality and ready when it was supposed to be ready.

Be ready to modify the plan if you have to. Things do turn out to take longer than expected; team members do let down the rest of the team by failing to do their work, (and if you are working alone you will sometimes let yourself down like this); genuine problems do arise: people get sick, equipment doesn't work or isn't available when expected.

You need to be ready to modify your plan if you have to. Once it becomes 'unrealistic', don't just leave it as it is. You need to replan the remainder of the project to take account of what's gone wrong.

If you are working in a group, never change the group's plan without consulting everybody, and making sure everyone accepts the revisions. Also note that depending on the reason for the changes there may be quite a bit of bad feeling in the group that you will all need to deal with. On team projects, you should review the plan at each team meeting, and make sure you all still feel it is realistic. If not, do something about it.

Afterwards, reflect on the process

Conscious reflection on experience is a powerful tool in the learning process.

Suppose a world class tennis player suddenly finds 'his serve has gone off'. Last month he was serving 15 to 20 aces a match. Now he's down to 7 or 8, and even worse, he's started serving double faults. Does he just say, "Oh well, better luck next month!" No. He consults with his coach; they watch videos of his good service games and his bad ones; they try to reason out what's going wrong. They then decide on a plan of action. The plan may be that he takes a holiday, and just relaxes a bit: but if it is, they will have good reasons for deciding that.

Professionals don't just 'hope for the best'. Whether they are professional tennis players or professional computer scientists, or professional artists, or professional teachers, or indeed professional students, they must consciously try to learn from experience.

When you finish a task ask yourself the three questions "What did I do well? What did I do badly? and how can I improve?" If the task was a group project, the whole team should reflect individually on the success of the project, then meet to exchange ideas about what went well and badly, and what can be done to improve performance next time.

Make good use of feedback from lecturers. The time lecturers spend evaluating your work is wasted if all you do is look at your mark, and say, "Well, that's ok then!" If the mark is good, make sure you can see why. If the mark is poor, it is more obvious that you need to work out why. Take careful note of all comments; discuss the returned work with other students; after that, if you still don't understand the comments or cannot see how to improve your work, see the lecturer.

This may all seem rather a bore. After all the effort and time spent doing the work, probably the last thing you want is spend more time reflecting on how it went. But this is the best way of learning for the future. One reward is that you will probably need to put in less time, and experience less hassle, over each successive task.

At work as well as at university, you will need to be able to demonstrate that you can evaluate your own work. Almost no one expects a difficult task to be done 'to perfection'; but no one is impressed by someone who is unable to identify why his or her work is less than perfect, and who cannot suggest ways in which it might be improved.

(Source: http://www.herts.ac.uk/envstrat/HILP/gradskill/intellectual/probframe.htm)