How to Solve Real-Life Problems: Lessons from Air Force Leadership

Martine Ceberio[1], Olga Kosheleva[2] and Vladik Kreinovich[1]

Departments of [1]Computer Science and [2]Teacher Education,
University of Texas at El Paso,
500 W. University, El Paso, TX 79968, USA,
mceberio@utep.edu, olgak@utep.edu, vladik@utep.edu

1. What we do in this talk

- A recent book from two veteran leaders of the US Air Force provides advice on how to deal with real-life problems.
- In the book, this advice is given in storytelling form, based largely on the book authors' personal experience.
- In this talk, we summarize this advice in a systematic and precise way.
- The summarized advice is highlighted by italics.
- Non-italicized text contains our comments.

2. Sometimes, we need to act urgently

- Sometimes, we face an urgent situation, when something happened suddenly, and an immediate action is required.
- In this case, there is not much time for a thorough analysis.
- Then, *we need to do what we can* (Chapter 12).
- We need to *build a plan and follow it* (Chapter 11, p. 103).
- We cannot do much analysis when such a problem appears.
- However, what we *can* do is prepare beforehand for such situations; see below.
- Good news is that in many cases, there is time to think before acting.

3. When we face several problems, which one(s) should we focus on

- In many cases, we face several real-life problems, and it is not realistic to solve all of them.
- In such cases, it is necessary to select a problem that we will focus on solving.
- In this case, a natural advice is as follows.
- To *select a problem in solving which you are most likely to succeed – because it best fits your strengths and talents* (Chapter 4).
- To be able to do that, we need to determine which are our strengths and talents – by analyzing our prior experiences.
- The book calls this advice "find your purpose".

4. Once we selected a problem, first, we need to understand it better

- *When we face a real-life problem, first, we need to understand it better* (p. 30, Section 3).
- We "need to *clearly articulate a problem statement* before acting on a proposed solution" (p. 80, Chapter 9).
- This is similar to what we Computer Science faculty teach students in our Software Engineering classes.
- Usually, the real-life problem is not formulated in precise terms.
- So, the students should not immediately start coding based on their understanding of the problem.
- Such coding is often a waste of resources.
- Indeed, the programmer's initial understanding of the problem is often different from what the user really needs.

5. Once we selected a problem, first, we need to understand it better (cont-d)

- In our 2-semester Software Engineering sequence, students are presented with a real-life problem at the beginning of the 1st semester.
- Usually, the whole first semester is focused on the proper formalization of the problem.
- This is done in dialog with the user(s).
- And only in the second semester, when both the students and the user(s) have a common understanding, the actual coding starts.
- How can we understand the problem better. For this, we need to do the following.
- We need to *collect as much data* about this problem *as possible* (Chapter 3, p. 31).
- We need to *elicit and collect opinions of others*, we need to actively listen to what people say (Chapter 9, p. 82).

6. Once we selected a problem, first, we need to understand it better (cont-d)

- We need to actively *use our own intuition* (Chapter 7) – what the book calls "trust your guts".
- Finally, we need *to take into account what can be done* within given time and with other resource constraints.
- In real-life situations, it is usually enough to satisfy only some of the requirements (Chapter 6, p. 56).

7. Once a problem is clearly formulated, how should we actually solve it

- First, we need to make a gap between the problem and our knowledge narrower.
- Then, we need to decide who to collaborate with when solving the problem, and start solving it.
- If we cannot solve the whole problem, we should solve at least some aspects of it.
- We need to plan what to do if our approach does not work.
- Once we have come up with a proposed solution, we need to test it.
- Once the problem is solved, we need to find the best way to present the solution to all interested parties.

8. How to make gaps narrower

- Real-life problems are usually very complex,
- The gap between our state of knowledge to the desired solution is usually too large to overcome it in one leap.
- To make leaps smaller, we need to come up with some intermediate steps.
- There are two possibilities to do it.
- One possibility is to analyze the problem itself and come up with natural intermediate steps.
- This way, transitions between each step and the next step will be easier.
- This known as *planning* (Chapter 3, p. 30).

9. How to make gaps narrower (cont-d)

- Another possibility is to look for intermediate steps:
  - that do not naturally follow from the analysis of the problem
  - but that would be helpful,
  - in the sense that their solutions would help us solve our problem.
- In other words, we need to look for *similar problems*.
- We must put a special emphasis on similar problems that have already been fully or partially solved (Chapter 3, p. 32).
- And, of course, we can try to use both strategies.

10. Who to collaborate with

- Since real-life problems are complex, we often need additional collaborators.
- How to select collaborators?
- According to the book (Chapter 6, p. 54), the most important criterion is enthusiasm.
- We need to *select collaborators who are enthusiastically pro-active.*

11. Solve at least some aspects

- Sometimes, we cannot solve all the aspects of the problem.
- In this case, we need to *prioritise some aspects and work on them* (Chapter 3, p. 33).
- This is:
  - in line with the previously mentioned advice
  - to take into account that it is usually enough to satisfy only some of the constraints and requirements.

12. What if this does not work

- If this does not work – *try unusual ways*; it sometimes helps.
- Sometimes it does not help right away.
- However, the resulting experience helps later on, in solving future problems (Chapter 6, p. 57).
- And it makes sense to *take risks*, to concentrate resources on a strategy that has a high probability of failure (Chapter 6, p. 56).
- This may not help us solve this particular problem.
- However, this experience often helps to solve future problems.
- After all, to be successful, a machine learning tool needs to know:
  - for each task – be it playing chess or doing something more practical,
  - not only many positive example, but also an equal number of negative examples.

13. Need to thoroughly test a proposed solution

- Shall we immediately implement the proposed solution in real life?
- No, it is desirable to *test the solution on simulations as much as possible* (Chapter 9, p. 78).

14. How to present a solution

- The *solution should be presented in a clear and explainable way*, “providing clear, concise, and thoughtful guidance” (Chapter 13, p. 125).
- It is also important to *emphasize the imperfection of the proposed solution*.
- Imperfection are inevitable.
- One of the main reasons: real-life problems and real-life situations have a lot of uncertainty (Chapter 13, p. 131).

15. Post-analysis is important

- Solutions to real-life problems rarely work perfectly.
- Once we know the results of using our solution, it is important to *analyze what we could have done better* (Chapter 7).
- Often, the low performance – and even failure – is caused largely by factors that are not under our control.
- However, usually, we can still understand what we could have done better.

16. Comment

- All these steps are similar to what is known as the IDEAL approach to problem solving:
  - Identify the problem – this is what we considered in the beginning of Section 4;
  - Define what we want to find – this is what we mentioned in the last paragraph of Section 4;
  - Explore possible solutions – this is described in the first subsection of Section 5;
  - take Action – this corresponds to Section 6; and
  - Look back – this corresponds to Section 7.

17. How to prepare for future problems

- How can we better prepare for future problems?
- What helps in solving problem is:
  - our own experience and
  - experience of others – that is often reflected in different available tools.
- So, the natural advice is as follows.
- We need to boost our own experience – *when there is an opportunity* to participate in solving a problem, *take it* (Chapter 13).
- The book names it “answer the call”.
- We need to better utilize others’ experience.
- We also need to *collect tools* (Chapter 3), even if they are not useful right now.
- This is an important part of *lifelong learning*.

18. How to prepare for future problems (cont-d)

- The advice about collecting tools is similar to how Stanislaw Lem explains usefulness of mathematics.
- Stanislaw Lem is a renowned science fiction writer.
- He is best known to the US audiences as the author of the novel “Solaris”.
- This novel was the basis for a 2002 film.
- Lem compares mathematics to a crazy tailor that makes suits for alien creatures of all shapes and sizes.
- At each moment of time, most suits remain unused.
- However, once in a while:
  - a weird-shaped creature appears,
  - and the tailor produced a suit that is a perfect fit for this creature.

19. How to prepare for future problems (cont-d)

- In mathematics, this happened many times.
- Here are some examples.
- There was a very theoretical notion on non-Euclidean geometry, in particular, of Riemannian geometry.
- urned out to be a perfect fit for Einstein's ideas about gravity and space-time.
- There were abstract mathematical notions:
  - of imaginary and complex numbers,
  - of matrix algebra, and
  - of infinite-dimensional analogs of the Euclidean space.
- They turned out to be a pefect fit for quantum physics, etc.

20. What if there are no visible problems

- Sometimes, in some areas, it may look like there are no real-life problems.
- Experience shows that such an absence of problems is an illusion, what the book calls *blind spots*.
- Serious problems surface later – and it is always better to start solving a problem as early as possible.
- From this viewpoint:
  - in situations in which there are no visible problems,
  - we need to *actively search problems* – e.g., by asking for people's opinions (Chapter 5).

21. Bibliography

- S. Lem, *Summa Technologiae*, University of Minnesota Press, Minneapolis, Minnesota, USA, 2014.
- H. Wilson and D. Goldfein, *Get Back Up: Lessons in Servant Leadership*, The University of Texas at El Paso Press, El Paso, Texas, USA, 2025.

22. Acknowledgments

This work was supported by:

- the AT&T Fellowship in Information Technology,
- the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany,
- the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329,
- the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590),
- the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand,
- the Ho Chi Minh City University of Banking, Vietnam, and
- Thang Long University, Hanoi, Vietnam.