Mistake-Proofing as the Base for Teaching Principles of Engineering Problem Solving

Zbigniew Prusak
Central Connecticut State University

Abstract

This paper describes how to employ principles of Mistake-Proofing in the teaching of engineering problem solving. The problem solving process starts with an intentionally brief introduction to the concepts of variability, robustness and sensitivity. Through solving a number of problems of increasing complexity or conceptual difficulty, students develop a list of mistake-proofing principles. Development of the list is based on the use of physical and chemical principles as tools for conceptually solving given problems. All solutions are conceptual only and have a goal of mistake-proofing the designs and processes in question. The solutions are then classified according to their mistake-proofing robustness. Student work is intentionally guided by the instructor, and switches between individual work, small team work, and brainstorming done together by the entire class.
Strong emphasis is put on process of defining goals of the redesign process, and critical description of shortcomings of the present stage of product or process design. A particular emphasis however, is put on knowledge of scientific principles which may provide ideal solutions to the problem at hand. In that respect, mistake-proofing of certain designs based on their geometry, as well as production processes prove to be fairly easy to understand and students are able to come up with a variety of solutions. Other advantages, limitations and disadvantages of using Mistake-Proofing as the base for teaching principles of problem solving are also provided and discussed from perspectives of faculty and students. A number of common and specialized devices used in exercises for defining physical principles underlying an engineering problem at hand are also listed in the paper.

1. Introduction

"Problem solving" – is it use of any method (scientific or not) to create something new, something improved; or is it use of sophisticated scientific methods and tools? Who are good problem solvers? Are they skilled users of sophisticated methods, or those who achieve improved or novel solutions using even simplistic methods, or those who employ existing solutions based on a scientific principle pertinent to the problem at hand? An important skill of problem solving is also *not solving of what is not a problem*. But are the students taught methods of recognizing a problem at hand, relating it to the basic physical and chemical principles, and following with creation of solutions based as directly as possible on these principles? For example, in a manufacturing operation a disc-like shape part is sometimes produced too thick and occasionally causes problems in subsequent operations. Is the problem in catching the nonconforming products, sampling production run to predict non-conformances and eliminating them, or is the problem "disc is too thick and what caused it to be too thick"? Do not heal symptoms of problems, eliminate causes is therefore constantly used as a prompt for the students to come up with better and better solutions.

Mathematically sophisticated tools are usually of little help here. "The simpler the better" mantra is repeated in many engineering courses at universities all over the world. Sounds straightforward and simple, but how to make students' minds follow that path of thinking when the vast majority of their time and intellectual effort while studying engineering is spent on learning mathematical principles and procedures with a goal of using them for optimization and numerical assessment of mostly classical, hence canned problems. Most engineering curricula still bare scars of teaching design through engineering drawings (blueprints, not conceptual drawings) and design calculations based on analysis methods that are rooted in specific subjects [1 - 3]. Understandably, due to the nature of learning, engineering problems need to be initially somewhat canned to prompt, if not require, the use of appropriate mathematics-based skills to successfully complete certain aspects of analysis. And here is where one of the biggest problems resurfaces. Functionality of most designs and systems is a conceptual and logical problem; hence the mathematical methods and algorithms are of little use. Additionally, another important aspect of today's engineering education, modern scientific ready-to-use software and hardware are great in enhancing analysis, but usually do very little is solving even simple conceptual and functional problems. From the learning incentives point of view, students are still examined and graded mostly on performance that is based on the mathematical knowledge, memorization of data and procedures for using them. Despite its abstract nature, mathematics-based engineering knowledge is easily quantifiable, and there is a lot of historical experience and academic materials available for teaching it. On the other hand, little learning time and grading effort is accorded to development of creativity, inventiveness and learning logical methods of designing. It is somewhat understandable, since the latter group is difficult to quantify, elusive in teaching and in assessment of student achievements. Petroski sums up the state of teaching how to design "Design has been a notoriously problematic aspect of engineering curriculum" [4].

It is interesting and ironic at the same time, that almost all historically significant engineering breakthroughs were achieved by using creativity and knowledge-based logic, not by optimization-based refinement of existing designs [5 - 7]. Prusak [5] provides compilation of results of some studies on persistent underachievements of engineering and technology education in the process of teaching how to find core of a problem and solve it in a methodical way that enhances chances of creating best solutions. The best solutions are again understood as solutions that are not a result of optimization of the existing state of matter, but solutions stemming from the physical or chemical core of the problem as described by basic principles of physics and chemistry.

Historically, engineer has not been a synonym for inventor, but rather for creator [8]. It is therefore unrealistic to expect that engineering and technology education should mold inventors. However, since for the most part engineering education does not loose sight of the fact that

The Best Solution is The Ideal Solution, it is definitely worthwhile to teach various techniques to analyze and solve a problem in order to increase likelihood of achieving The Best Solution.

2. What Level of Problem Complexity to Use?

Petroski [4] advocated high value of using case studies of historical engineering failures as an effective way to teach design via solving a failure problem at hand. However, he also mentions that most of the case studies available in literature usually require possession of substantial background theory, often from various engineering disciplines. Without that prerequisite knowledge, understanding of the problem may be incomplete, and search for The Best Solutions will very likely be unsuccessful. Therefore, due to the complexity of the available case studies (these need to be complex to be worth publishing) they have value when used with well prepared students.

Is studying one complex problem (e.g. a failure) better than studying a multitude of simpler cases? Where does the above stand against the fact that a substantial number of patented inventions are not conceived by trained engineers? In Canadian French there is even a special word for such person, "patenteux" [9]. Patenteux stands for: creator, innovative problem solver, inventor and artist [9, 8]. Based on 10 years of unpublished surveys of engineering technology students at sophomore through senior level engineering technology courses at Central Connecticut State University (CCSU), the students express strong preference for repetitive learning based on practical examples. That fact alone clearly indicates a preference for multiple small projects rather than one complex and time consuming project. The flow of teaching activities described in this paper reflects experiences gathered in teaching Design for Manufacturing senior level course in Engineering Technology.

The above thoughts can be summed up as three goals of design:

1. Product/process executes tasks necessary from final product point of view.
2. Product/process does not execute tasks unnecessary from final product point of view.
3. Developmental goals are ideal solutions not optimized (compromised) ones.

These goals are given to the students at the beginning of mistake-proofing based design activities and all the solutions the students come up with are verified against them.

3. Reasons for Using Mistake-Proofing

With the assumption that multiple but rather simple design projects are better suited for the students due to a wide range of their background knowledge, mistake-proofing improvements were judged to be the most powerful tool in teaching principal aspects of engineering problem solving. Due to the nature of mistake-proofing, aiming at clear distinction between good and bad (accept – reject, Go – No Go), the superior mistake-proofing solutions easily stand out from the inferior ones. Using mistake-proofing concepts it is also fairly easy to keep focus on achieving "the simpler the better" solutions. Additionally, all the mistake-proofing solutions call for some form of innovation and not for optimization of the existing solution. However, a good knowledge of natural effects (primarily physical, and occasionally also chemical) is required in order to describe the problem at hand, understand it and come up with feasible solutions.

Additionally, engineering contradictions (defined as a state of design parameters in which improvement to one parameter of the system deteriorates another parameter) are well suited for incorporation into the problems. Solution to a design problem at hand cannot be achieved while continuously using these existing two parameters. Any improvement achieved by optimization, leads to a compromise in influence of these two parameters. Again, there is a need for fundamental improvement of the system based on its physical fundamentals.

A recent study on best practices conducted by the PMDA (Product Development & Management Association) indicates that companies with the most successful new product development programs [10] have the below common characteristics:

1. Have more up-front activities at the beginning of the development process
2. Do more up-front thinking and subsequently less tweaking once an idea is implemented
3. Strive for simplification since simple systems are less prone to failures, and give fewer opportunities to tweak

Looking at the use of mistake-proofing from axiomatic design point of view [11, 12], especially the second axiom (the Information Axiom – requires minimization of the information content of the design) is well encompassed. Additionally, the first axiom (the Independence Axiom – requires independence of functional requirements), which parallels the earlier described engineering design contradictions, is also encompassed, especially in simple problems.

Last but not least, the functional decomposition of a product or process, which is at the kernel of mistake-proofing, is arguably the easiest avenue for understanding them. Since teaching various methods of design concept generation [12 - 15] is difficult and dry when taught theoretically, some methods become almost deductive when solving a design problem.

4. Mistake-Proofing: Introduction and Examples of Using

Mistake-Proofing is based on recognition of the fact that all human beings make occasional mistakes, but these mistakes can be detected and prevented before they create a defect. Application of mistake-proofing results in elimination of human-related defects, reduction in continuous checking, and increase in production output. Concepts underlying mistake-proofing are simple and every student can easily find a personal experience relating to them. That makes introduction to the subject rather quick, easily supported with everyday examples (e.g. electric plugs and outlets, automotive headlight bulbs and their sockets, automotive battery posts and connectors, stamping die guides).

Before students realize it, they solve design problems intuitively or by drawing from life experiences. The challenge starts when they are asked to describe a physical principle used as a base for their solution or solutions, and then find even more solutions based on this principle. This conceptual step back from reality (a solution that has an envisioned embodiment) into abstraction (a physical principle) in order to create a new, not envisioned yet solutions, always proves hard to make. However, after multiple exercises many students buy into using the abstract part of the analysis in order to create novel, at least for them, solutions.

The next step is evaluation of solutions. Concepts and subtle differences between Mistake-Proofing, Fool-Proofing, Idiot-Proofing and Sabotage-Proofing are introduced. They are followed by Jidoka (the concept of fitting machines with devices that self-check, detect defects and stop the process thus enabling operators to do more value-added tasks than mere watching the machines to spot faults). The last concept introduced is Zero Quality Control – the target of mistake-proofing. The far reaching target of that concept is not only zero defects, but also zero concern for Quality Control type inspections that usually carry a human judgment.

Due to the fact that different mistake-proofing techniques have various degree of effectiveness, the subsequent step in evaluation of solutions is development of hierarchy of mistake-proofing techniques. During this exercise that is based on collective analysis of various mistake-proofing solutions students are asked to evaluate the effectiveness of each one. Fairly quickly they come up with the three levels of effectiveness of mistake-proofing devices shown in Table 1. The following are examples of designs used in the exercise:

- electric plugs and outlets for various AC voltages and in different countries
- automotive headlight bulbs and their sockets
- standard automotive electrical connectors
- automotive battery posts and connectors
- injection molding and stamping die guides
- many fasteners used in cellular phones (results of analysis vary widely between manufacturers)
- batteries for cellular phones and camcorders
- electric connectors in consumer electronics
  - supplying power
  - carrying signal type (audio, video)
- PC-external device connectors (for external devices used in personal computers)
- computer external storage devices (5 ¼”floppy, 3.5” floppy, Zip disk, CD, USB stick)
- photographic film cartridges for 35 mm cameras
- storage cards for digital cameras
- use of limit switches in assembly operations for part position
- use of limit switches in machine tool motions
- automatic error detection and alarm to avert an operator
- audio alarm
- visual alarm
- vibratory alarm
- displays
  - analog
  - digital
- counters

- checklists
- safety light curtain (around a machine tool)
- safety enclosure (around a machine tool)
- all-must-be-pressed safety buttons (around a machine tool)
- hand chains (around a machine tool)
- digital scale
- analog scale
- packaging crates
- vibratory bowl feeders
- Go –NoGo gages
- functional GD&T gages
- CMM measurement output read correctly
- verbal commands
- displayed commands with indicated course of action
- bar code reading used for recognition (different applications assessed) on device level and system level)

All of the examples are assessed on device level and some on system level as the device is part of the system.

Table 1. Hierarchy and effectiveness of Mistake-Proofing devices.

Device type	Symbol	Level of effectiveness	Description, examples
Prevention	P	Most effective, on the level of fool-proofing	Only one way is right: one way assembly, one way process throughput
Detection	D	Less effective, can be circumvented	Process attributes (weight, voltage, pressure, etc.) Physical characteristics (contact/non-contact switches, use of jigs, pins, etc.) Operator or machine motion
Judgment	J	Least effective, may be even subjective	Visual controls Audio controls Color coding Status indicators Counters Checklists

5. Projects

5.1 Examples of Projects

Figure 1 shows part for which students need to do the following documenting steps of analysis, design and evaluation:

- design functional gage
- mistake-proof the gage
- mistake-proof the part

Figure 2 illustrates one of the solutions. Figure 3 shows example of a system in need of mistake-proofing. The system consists of mechanical scale, shipping boxes, finished parts and workers manually packing the parts into the boxes. The workers determine by weight if the proper number of parts have been packed (alignment of needle and the mark). In reality, an incorrect number of parts are occasionally shipped. Two design goals are given:

- mistake-proof the existing setup
- redesign the process (with mistake-proofing in mind)

5.2 Notes on Outcomes

The positive of using mistake-proofing for teaching design process is the relative simplicity of problem definition, and usually no need to work on system level. It is due to the fact that work on device level executing just the required function gives best and most robust results. Evaluation of multiple solutions students come up with is fairly quick since the primary concerns are robustness of executed function, robustness of the device, its simplicity and cost.

One of the noted negatives is that students often come up with multiple solutions quickly and persistently take shortcuts avoiding methodical assessment of the problem at hand and evaluation of solutions. Methodical follow through of at least some phases of a structured and disciplined approach to problem solving, such as DMAIC (Define, Measure, Analyze, Improve, Control) [16], was somewhat difficult to execute due to the above mentioned students' unwillingness to make an effort to work in a structured way. Another observed problem is a set of several design perennials [5]: formulation of engineering contradictions in a given design, development of metrics as well as choice of tools and procedures to evaluate final solutions.

'The simpler the better' had to be reminded constantly, even after completion of few exercises. The drive to add sensors, computerized logic and actuators has proven very difficult to uproot. It follows however the engineering reality that for improving an under performing system or device, a detection device is almost always conceptually easier to design than a prevention device.

Some design problems were solved in small groups of 3 to 5 students. Conversations within a group were intentionally stimulated by the assigned work type, which switches between individual work, small team work, and brainstorming done together by the entire class. After individual work period, each team member must informally present his/her findings and solutions to the rest of the group. Multiple opportunities to repeat the design process, even though in a much abbreviated form, increase buying into the rigors of conceptual design work and needs of keeping at least a small documentation and use it for presenting ideas to peers.

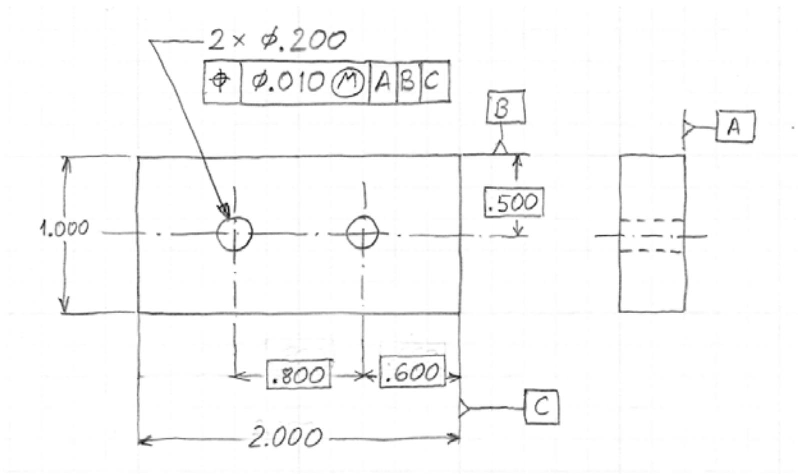

Figure 1. Example of a part used for analysis, design and evaluation.

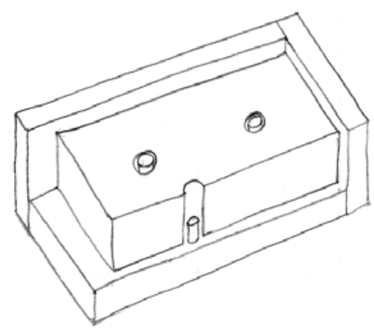

Figure 2. Example of gage for functional verification of the part.

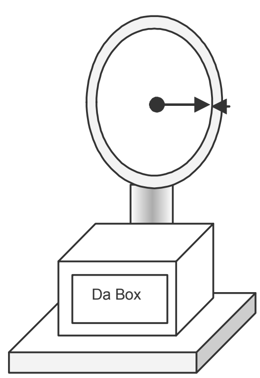

Figure 3. Example of a system consisting of mechanical scale, shipping boxes, finished parts and workers manually packing the parts into the boxes. It was used for mistake-proofing design project on level of device and process.

6. Summary

Problems used for the short projects aimed at development of problem solving skills dealt primarily with designs of well known devices and very common manufacturing processes, therefore were easy to understand. No equipment was needed to run the projects. The positive of using mistake-proofing for teaching design process is the relative simplicity of problem definition, and usually no need to work on system level. Work on device level that provides required functions is often sufficient and results in most robust solutions, which are usually directly based on the underlying physical principle. Evaluation of multiple solutions students come up with is fairly quick since the primary concerns are robustness of executed function, robustness of the device, its simplicity and approximate cost. Observations from running multiple short exercises and projects, showed some development of problem analysis and solving skills of at least half of the class population. Requiring simple solutions has fostered some fairly creative solutions. Although some students showed signs of getting bored by the somewhat repetitious nature of the small but fast paced exercises, the vast majority considered that type of learning very thought-provoking and motivating.

Bibliography

1 Ferguson, E.S., *Engineering and the Mind's Eye*, MIT Press, 1992.
2 Prusak, Z., Laboratory Experiments in Process Design and Optimization, *Proceedings of ASEE Annual Conference*, June 24-27, 2001, Albuquerque, NM.
3 Prusak, Z., Challenges to Future Engineering Professionals – How to Prepare Students to Face Them, *Proceedings of ASEE Annual Conference*, June 28- July 1, 1998, Seattle, WA.
4 Petrosky, H., *Design Paradigms – Case Histories of Error and Judgment in Engineering*, Cambridge University Press, 1994.
5 Prusak, Z., Do Not Optimize, Solve the Problem – Development of Critical Thinking Skills in DFM Small Projects, *Proceedings of ASEE Annual Conference*, June 22-25, 2003, Nashville, TN.
6 Rantanen, K., Levels of solutions, URL: http://www.triz-journal.com/archives/1997/12/d/index.htm , viewed on 1/3/2004
7 Zlotin, B., Zusman, A., Managing Innovation Knowledge, URL: http://www.triz-journal.com/archives/1999/04/c/index.htm , viewed on 1/3/2004
8 Prusak, Z., Fostering Inventiveness in Engineering Education – an International Perspective, *Proceedings of ASEE Annual Conference*, June 16-19, 2002, Montreal, QC, Canada.
9 http://www.societehisto.org/Musee/Recits/patenteux.html viewed on 1/6/2004
10 Lean Product Development Group. URL: http://www.lean-product-development.com/issues.htm , viewed on 1/6/2004
11 Axiomatic Design Solutions Inc., General Axiomatic Design Concepts. URL: http://www.axiomaticdesign.com/technology/tech_adt.asp , viewed on 8/30/2003
12 Ullman, D.G., *The Mechanical Design Process*, McGraw-Hill, 1997.
13 Cross, N., Engineering Design Methods – Strategies for Product Design, John Wiley & Sons, 2000.
14 Dixon, J.R., Poli, C., Engineering Design and Design for Manufacturing, Field Stone Publishers, Conway, MA, 1995.
15 Otto, K.N, Wood, K.L., *Product Design, Techniques in Reverse Engineering and New Product Development*, Prentice Hall, 2001.
16 *Rath & Strong's Six Sigma Pocket Guide*, Rath & Strong Management Consultants, Lexington, MA, 2002.

ZBIGNIEW PRUSAK
Zbigniew Prusak is an Associate Professor in the Engineering Technology Department at Central Connecticut State University in New Britain, CT. He teaches courses in Mechanical, Manufacturing and Industrial Systems programs. He has over 10 years of international industrial and research experience in the fields of precision manufacturing, design of mechanical systems and metrology. He is actively involved in joint research with industry. Dr. Prusak received M.S. Mechanical Engineering from Technical University of Krakow and his Ph.D. in Mechanical Engineering from University of Connecticut. E-mail: PrusakZ@ccsu.edu