Experiments in creating Learning outcomes

Kalyan Kumar Banerjee

Objective

Education design must be aligned to the context of the learner. Not just that, it must be paced in tune with the motivation and receiving capability of the learner. If we ignore these, we may still produce outcomes but they won't be a consequence of conscious design.

a. Content of this paper

This paper discusses multiple experiments and focuses on the more successful ones. It tries to draw lessons from both the successful and failed experiments, and propose a model for traits of effective design of education.

The discussion is about creating meaningful outcomes for the less privileged in Indian society. The author proposes diverse pedagogies to meet diverse needs, and argues that what works in an elite segment of society may not work as effectively in another segment. This paper also presents a view of education as if most learning happens outside the chalk and board world.

b. Is this paper relevant to you?

This paper discusses various experiments, and more important, insights from the experiences in these experiments. These insights have been arrived at within a specific context –and while the readers are welcome to adapt these in their environment, they are encouraged to discover what is right in their context. In that sense, treat the discussion here as a set of thought provoking ideas, rather than as a prescription for the perfect education.

The author does not claim these will work for all time for all people, in fact they won't; yet these are demonstrating benefits within a specific context. This paper argues that education is the consequence of interactions of multiple entities in a culture, and it is possible that all elements of a culture are not well understood, nor controlled. Some demonstrable results are discussed, however it is possible there are unknown factors in disguise that have led to the positive results (like the Hawthorne experiment). Rather than dissect the cause, the author suggests the readers embrace the outcomes, choosing whatever works for them; and if it works, does not matter if the factors were wrongly understood – that understanding will be corrected in time if we are observant and have an open mind.

The paper argues for inclusion and diversity of methods. Experiential learning relevant to student's context is also projected – there may be more to learn in a Sleeper class rail journey to Pondicherry, or in appreciating Mangalyaan, than in flying to NASA. It is also more affordable.

Background

In the semester of July-December 2013, we introduced a hands-on course on programming with our 2nd Year CSE (Computer Science and Engineering) students. The primary intent of the course was to ensure no one goes beyond 2nd Year without the ability to write original code. This is a radical goal given the reality of today, where most students beyond some 20 colleges cannot write original code (many of them memorise code, and some copy – exams have known programming questions from books with solved answers).

However, we believed this is an achievable objective, if one has the motivation.

We understood this objective has many hurdles. Among them are these realities:

1. Students come into the class with widely varying capabilities, given the fact they come from diverse backgrounds. How do we cater to everyone?
2. Programming is not learnt inside the classroom. It's learnt by programming, and that's done at home/hostel. (Just as swimming is learnt only by jumping into the pool, not by watching videos, nor by inspirational talks from Olympians.)
3. Programming is best learnt by working with buddies/peers. They critic my code, and I critic and learn from theirs. In a diverse background where many students feel less confident for a variety of reasons, a culture of collaboration to learn needed to be nurtured carefully.

To help us in this objective, we devised several experiments.

1. We created and worked with "10 levels of programming capability". These levels were based on an objective list, with at most 3 simple capabilities in each list. In the first week, students self-assessed themselves and found their "level"; (where students over-positioned themselves, they were moderated.)
2. The students at highest levels (8, 9, and 10) were designated captains. Other students chose their captains (lowest levels, e.g. level 1 and 2, got first choice) – and thus we formed 10 groups of 5 to 6 each. Captains were expected to support and share, so groups learn together.
3. The semester's objectives and coursework were announced at the beginning. Students were given a set of 50 programs, and they were expected to write as many as they could. Every successfully running program would lead to points collected. They will be evaluated based on how many points they collect.
4. Custom objectives were given based on the student level. These objectives were created after asking students at each level – so we had goals that students said they can meet.
5. We set mid-semester targets. Group incentives and class incentives were announced.

After mid-semester target (Aug end, 2013), many of the original objectives were not met. That was very disappointing. However, we found one thing to cheer about – around one quarter of the students exceeded their targets, and were collaborating visibly. Another quarter seemed motivated or interested in the class but forgot all about it after 5pm.

Motivation behind the excursion

This led us to plan a scenario where we have "control" over the students' time, beyond 5pm. Our objective changed from "teaching" or "showing" to motivating. If they are motivated, they will figure out how to learn, including catching faculty for it – by now, they knew faculty was accessible and responsive – so motivation was the issue. So was environment. It's difficult to be the only one doing this stuff, in an environment where everyone copies or memorises code.

This thought process led to a different plan – which is the subject of this document. We needed to take the motivated students away from it all – to a different environment (with none of the usual constraints, nor the usual excuses) where we are with them from early in the morning, till midnight. We wanted to make a difference to their habits, so they spend every hour (relaxation included) in a manner that is useful to their future. The objective is to get them motivated to programming – they should get into this habit because they enjoy it.

The Programming Excursion

The programming excursion took us to Pegasus, by the sea, 22 km north of Pondicherry, on the Chennai ECR. A 24-day marathon effort, where students spend 7 to 12 hours on programming, every day, and rest of the time is spent on other important but secondary goals, all aimed at creating exposure and inspiration. Primary goal is to discover the joy of programming, and become so inspired that one will continue the habit. Just being with same faculty or just doing programming will be boring, hence variety of exposure was planned for. They experienced 10 faculty members from different organizations. All ten exposed students to some perspective that broadened their knowledge and inspired them.

The learning journey in fast-forward left students motivated, who worked well beyond midnight (not their usual behaviour), and showed radical improvement. They successfully wrote a Sudoku Solver, integrating many approaches. Then moved on to write a Buzzword Analysis software - all this from a generation of students not used to writing any original code.

There are elements here for effective education of the future. We began with the conviction that everyone has the capability to perform, provided one has the right environment and the motivation – and we feel happy it worked in this case. The next phase was to ensure they live with the habits we nurtured there.

Diverse perspectives and inspiration

Each of them wrote a program to solve the Sudoku, but that was not the only thing. They interacted with three learning experts from Pegasus and Mindtree, and were exposed to a variety of outbound learning experiences, psychometric instruments, introspection and reflection exercises, group discussions, and brainstorming. Late in the evenings, they played Scrabble, participated in quizzes, and took part in debates. During the 24 days, they were also exposed to

six industry experts from Bangalore who introduced them to a variety of facts, trends, tools, and views. Sundara Nagarajan, veteran industry professional (Technical Director at NetApp and operating systems faculty at IIIT), took them on a technical journey through Linux. N. Ramachandran, Quality Head at Mindtree, used lateral thinking games, role play, participative discussions, Ted Talks and outbound games to introduce them to seven traits of successful professionals. Rajesh Kumar, technical architect at Happiest Minds, demonstrated structured learning in unstructured environment, titled “Hello World to Real World”, peeking into Software Architecture and building real world software. Also introduced students to the technologies of the future like Big Data, Cloud, and the possibilities they create. Ananda Rao Ladi (APAC and Australia Head at Mindtree) introduced them to Purpose, and the need for finding one. Dr Raghunath (head of Mindtree Research and CTO) focused on innovation and took them through an exercise on thinking and algorithms. Sunil Kand of Peak Time Infosystems discussed challenges of the future like urban transportation and engaged them in a problem solving journey. All of them shared their life’s experiences and their belief systems. While all of these appeared different, they delivered **some common messages – of purpose, hard work, building expertise**. With diverse facts, and in diverse ways, they created aspiration. Each of them left them inspired, and energised to continue their programming exercise.

Essential Successes

What the students really got from this:

1. **Joy of Programming:** they can now effortlessly program well beyond midnight – they have imbibed the magic, understood this is a joyful activity, just like music, dance, or football.
2. **Discipline of Programming:** they have practiced and learnt three critical elements of a programmer, viz.
    a. Breaking down a big problem into smaller, easily code-able parts: most average professionals get anxious when presented with a real-world complex problem; their mind freezes, and they believe they cannot do it. In this excursion, students learnt how to break down a big problem into smaller bite-sized steps, with each step being extremely easy to solve. (If you cannot write a program to solve the entire Sudoku, can you solve one cell, at least? And then, one more?)
    b. The art of debugging: all programmers create bugs (or defects) – only a genius writes code that works right first time. The excellent programmer learns the art of debugging: quickly identifying bugs, understanding why they occurred, and finding solutions. The best of them even enjoy the process of finding defects and debugging. The students were mentored to learn that bugs are normal, received personal coaching with the mentors showing them how bugs in their code can be discovered and corrected. Now, the students cannot be scared by defects any more – they have the confidence to debug and create programs that work.
    c. The art of writing good code, and integrating with others' code: Initially, student programmers write code as they like – with poor appreciation of style and readability, they introduce variables and structures as they choose, and write code that is barely readable. In real world software, this won't work as code we write must be reusable and will probably be integrated with code written by others. During the excursion, students were asked to form groups and integrate their code with code written by their peers – this ensured a struggle where they learnt why style and discipline are important – the need to integrate forced them to adhere to right coding styles.
3. **Purpose:** The students learnt how to dream big, think big, and found the motivation to work hard. This came from exposure. Meeting with six technical professionals, and three specialist learning experts made a critical impact. They got exposed to diverse opinions, multitude of facts, and, not the least, their inner selves. They learnt how much they don't know, and how much they need to learn. They learnt how ordinary people like them have achieved so much (but not without a struggle), and they can do it, too, if they work hard for it.
4. **Thinking and Questioning Mindset:** Students learnt how to think, how to question, and gathered the self confidence to think and question.
5. **Collaboration:** Problems thrown at them were complex enough to need collaboration to complete their tasks meaningfully, and simple enough to think on their own.
6. **Self Awareness:** Diverse exposure, psychometric instruments, personalized mentoring and continuous (but achievable) challenge led them to discover themselves.

Critical to Success

Many hypotheses on education were experimented with, and proven through this exercise:

1. **Environment to Fail:** It is the job of educators to provide an environment where students can try, learn, experiment, make mistakes, and fail. The job of the teacher is not to show the right answer, but to ensure the student is motivated to struggle and gradually move towards the right answer, possibly after exhausting a variety of options that do not work, and hopefully with the understanding why they do not work.
2. **Show and Tell:** The faculty repeatedly demonstrates by example – yes, it can be done! And shows in a simple way – so simple that the lowest quadrant believes it is possible. Expert faculty must eschew the temptation to demonstrate magic, in some esoteric way. That leaves the audience impressed and fetches many admirers, but strengthens the divide between the "Knows" and the "Know-nots". The purpose of teaching is to break this wall, and not to build admiration for self.
3. **Custom Advice:** Students do not need generic lectures, content is available in plenty. Students need custom support. When they make a mistake, someone has to understand that specific mistake, show them why they made it, how they could have prevented it, and how to find the way forward.
4. **Incremental challenges:** Depending on the context of the student, challenges need to be customized, just a little above what the student thinks she will achieve easily. When a student has zero confidence in speaking English, she needs to be encouraged to just speak and keep speaking (forget the grammar). She needs to be applauded for speaking, and the content can be dealt with later. Similarly, students need to be encouraged to write original code rather than comment on the style or the bugs – that criticism can come later.
5. **Encouragement to think, question, challenge –** students must be encouraged to question, and must never be asked to shut up – this builds confidence, curiosity, thinking.
6. **Empowerment –** the students in this excursion, after a few days, made their own schedules, provided feedback on their own, and solved their own problems. Students do not need to be "controlled", they need to be liberated from rules that constrain their curiosity or ability to experiment. How else can we nurture the innovative and responsible generation of nation builders?

The excursion exploded the myth that "merit" lies only within the upper echelons of society, those who speak good English and study in the best schools. It demonstrated that students from all sections have the potential to do well in life and in society. However, students cannot be forced to play or be evaluated against an alien process mastered by a select few – the learning environment needs to be set up so that it is conducive to their learning, it needs to take into account their needs, their aspirations, their limitations and their context.

Conclusions from this pilot

This experiment demonstrated that all sections of society have the potential to be nation builders in emergent India. It showed how this can be done. If we do not frame the rules of learning, and the environment of education in an equitable manner, some sections will remain forever discriminated against. However, if the environment is framed to motivate, to inspire, and takes into account individual strengths and weaknesses, we can ensure every student meets her potential and together we create an inclusive, just and vibrant society.

Progress beyond the Pilot

What the students felt about the pilot

The students were inspired, and of course, wanted to repeat the experiment. More important, they wanted more students to be included in the new learning methodologies. They also underlined why "Pegasus" was a stunning success and why some of those successes cannot be replicated in the regular campus environment.

Success factors

The typical class is of one hour and the typical curriculum is divided into one-hour modules. This helps modularize education and makes life easy for the learning administrators. In real world learning, however, the brain takes time to warm up to a concept, and by the time it is ready to engage, it is time for another class and another faculty member. Thus while the one-hour slots are ideally suited for superficial knowledge transfer that does not engage the intellect, they are not meant for deep engagement. When learning delivery is monotonous, students will complain 45 minutes are too many. When engaged, three hours don't seem long enough!

The environment in the hostel, or campus, is not designed for programming excellence. In 2013 when this experiment was conducted, no one in the hostel ("dorm") is programming, and it's difficult to do such things alone. (It is refreshingly different in many IITs, and a few IIITs.) At our Pegasus experiment, everyone was doing it, so it was easier to just conform.

At the excursion, students focused on just one problem for ten days. So it was easy to focus. Is it possible to do the same during the semester, they queried. Instead of spreading our attention among four deep subjects over the four month semester, can we have one subject at a time, month by month?

All these were pointed out by students and not educators, it is interesting to note. The key question, therefore, is: if this simple method produces outcomes, and the traditional methods have been repeatedly found ineffective, expensive and sparse in joy, is there a strong case for change?

The many minis

Through the year 2014, we tried a number of mini experiments, each small on its own, with varying degrees of success. But, together, they tell a story.

1. The empowered hostel

Students aren't motivated to study. The natural response is to control behaviour, largely by negative incentives. If they don't come to class, create an attendance rule, making it expensive to bunk. If they don't study in the evenings, create a Study Hour and strict code of discipline in the hostel. This code has created discipline, but learning outcomes still remained illusory.

We argued for different rules to address different levels of inspiration. While the demotivated students needed the stick to force conformance, the inspired student will probably flower with empowerment and a sense of trust and freedom. The concept of Hostel 1 was precisely this – get the "higher CGPA" students into one hostel, remove the rules, and watch for results. The results have shown that students are as responsible, and as committed to studies – and probably more inspired.

2. Go to their space and make a difference

If students don't go to the library, bring the library to the students. The concept of hostel library had its genesis in this concept, and the empowered Hostel-1 was the first of the beneficiaries. This worked like a charm, creating the case for more such libraries. We also observed the lack of general awareness among both students and faculty, and focused on reading the daily newspaper as a possible solution. The newspapers were kept in the library, but why should students read it?

Again, based on student input, late evening jamborees were organized. From 10pm to 11pm, we started Quizzes and Debates in the hostel. These were received well, with participation from 15 to 40 percent of the total strength. This ensured participation and the sense of competition led to reading newspapers.

3. The programming evenings

As discussed earlier, programming expertise is acquired by programming, and such practice cannot be limited to academic schedules. How do we nurture the programming habit? Designated labs were kept open for programming practice, with presence of faculty to clear hurdles.

4. External Exposure

The author has argued on the need for exposure and inspiration. This came in many small ways: Innovadors' Lab set up a lab for our students with real world learning experiences through software development every weekend. Mindtree and Showhow2 provided real world projects for our students to work with. Edureka provided practitioner-expert to teach Android application building with a show-and-do pedagogy.

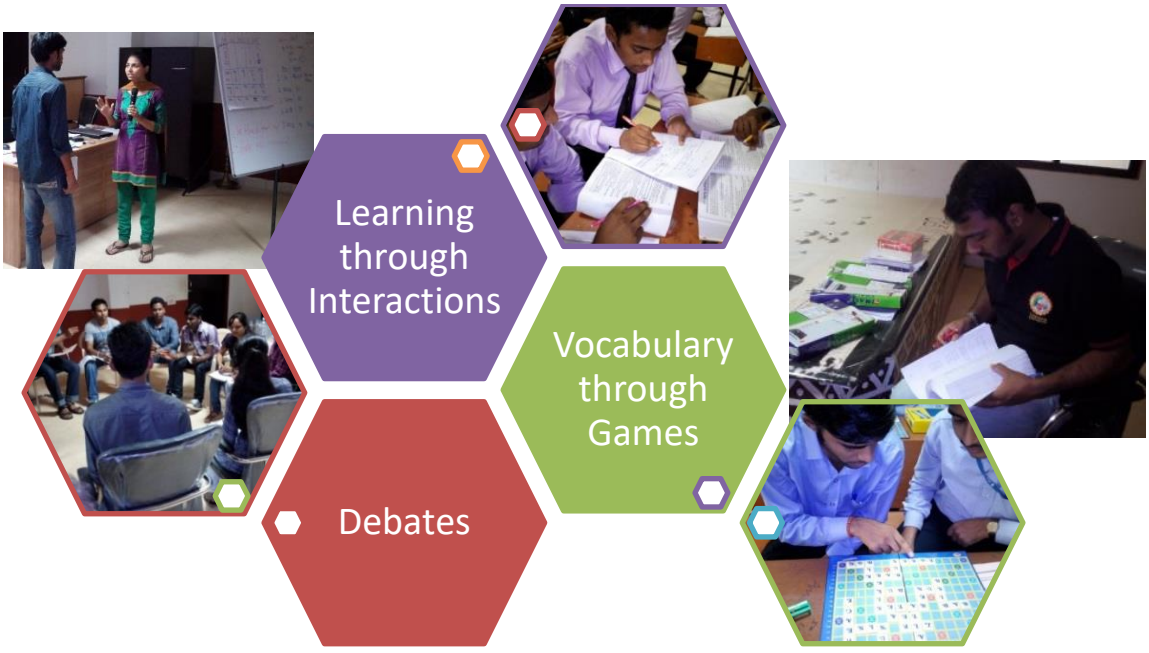
Not all of these created effective learning outcomes – presence of engaged guidance who spent quality time with the students was a critical success factor. While we have argued for

empowerment and freedom, we repeatedly observed at low maturity levels, students cannot just be left alone. These ***methods showed results only when faculty was around to guide, to inspire,*** to show the path when students encountered technical obstacles. In a low maturity (and low confidence) system, many students tend to give up easily – a mentor has to remain constantly engaged, inspire, identify technical obstacles, and gently nudge the lost student towards the right path. And ***remain open to changing the rules***, the methods, the schedules, and even on where to begin.

The bottom line: keep creating external exposure, nurture an environment of real world projects, ***but*** keep watch, inspire and guide – without active support, students will not progress. In that sense, the education in the less privileged context needs more intense engagement compared to the confident learners.

5. Immersion programs - Do more of the good things

The learning excursion resulted in an accelerated learning experience. Can it be repeated in campus (very different environment), with a different set of students, different faculty, and with different goals? We tried two immersion programs for job readiness – where we taught maths, logic, and English – tried to keep the fun experience, with groups, captains, competitions and scoring, quizzes and debates, games. These created learning experiences resulting in useful outcomes. In neutral tests by authentic testing market leaders (CoCubes), our students leapfrogged to top position among more than 100 colleges. In fact, in a span of three months, the average scores of our students came consistently within ranks 1 to 9 among averages of all colleges in the country. So there was data to show the efficacy of these learning methods.



The Immersion programs were cleverly planned. The first one was on a 3-day weekend, starting on Independence Day. The second one started on a Sunday. This ensured the more motivated lot

participated, and improved chances of success. **Motivation is tapped upon as a critical factor** in the learning adventures.

Sometimes, it takes time and effort for the momentum to build up. The evaluation, even self assessment for outcomes, are triggers for building motivation. By design, these come towards the end. Can we design programs where the student's perceived end of duration is the faculty's intermediate milestone? This ensures the faculty can **leverage on the student's end-of-program motivation** to inspire better learning outcomes.

6. Exposure to projects

We discussed projects in the context of external exposure. However, not all projects need to be external. It is important to create and nurture a constant stream of internal projects, so willing students remain engaged doing something useful, and creating something useful. The idea of being useful to others enhances confidence and self-esteem, and that's a catalyst to the learning process. During the last year, we had students building applications for the University, doing laptop maintenance, training on and installing Ubuntu, … And all of these created an atmosphere of purpose and motivation – learning leading to confidence, and then to more learning, creating a virtuous circle. The academic system needs to make this project based education the core learning process (and not an exotic showcase pedagogy) – **Corporate Readiness Lab** to teach Communicative English is one example of a formal course using project pedagogy, and it has been far more successful than traditional courses attempting to teach English.

7. Learn by Teaching

The best way to learn is to teach. After a few stunted attempts, there are now a couple of Coding Clubs and Aptitude Clubs, where senior and enthusiastic students are teaching others. That's a good beginning, and another small milestone in creating a learning culture.

8. New Age Teacher program

Teaching in the abundance era, with the internet providing supposedly unlimited access to knowledge and mentors is a different ball game from teaching paradigm evolved over the last 200 years. That is the central theme of this paper.

To succeed with the concepts described here, we need to build capacity – our teachers, schooled in the traditional method, must rise above the limitations of their systems to deliver next generation education. To enable this, Centurion University is at the forefront of building New Age Teacher programs that combine introspection, assignments and projects, outbound experiences, self assessment instruments, brainstorming and live changes in their pedagogy to get assimilated into the new culture.

Two week programs have been running for two summers now, with one-day modules during the semester on multiple behavioral themes. Is there change? Yes, apart from visible positive energy among the participants of the one-day programs, faculty from the two-week summer program have tuned into more interactive, more interesting, and more energizing pedagogy.

9. Involving parents

The parents are an essential part of the learning process. At Centurion University, parents have been involved in multiple ways:

- Regular information on students' attendance ensures parents remain informed, so as to pre-empt matters from going out of control. This has reduced students from straying away right at early stages.
- Parents have phone calls and meetings with faculty who double as Foster Parents for their children. (The Foster Parents are trained on mentoring.)
- Parents are taken into confidence on key initiatives like Job Readiness programs, and informed on consequences when their children do not take part.
- Parents were explained the purpose of our learning experiments, so they are encouraging when their children attend the specially intense programs.

10. Evaluation methods

On one occasion, we gave a 20 question quiz. After evaluated answer sheets were returned, we went through a discussion of all the questions at length, and did not stop till all students said they have understood all questions (and solutions). One month later, we repeated the same question paper all over again. Ideally, all students should have got 100%, right? No one did. This demonstrates one very important principle of learning – just because you have taught once and students believe they have learnt, it does not mean they have learnt. The same concepts must be repeated over and over again for assimilation to occur. **Redundancy is necessary for learning**, unlike in the business world where redundancy is eliminated, for efficiency.

Projects, presentations, vivas, declaring questions in the beginning of the semester, are other evaluation methods that have been used. We have noticed there is a tendency to take lightly all exams but the memory-based exercises in formal settings – and that's a tragedy. In building the right culture for education, building the right evaluation mechanism is critical.

Evaluation is still an open, immature art. How does one grade? On one occasion, three of the best students chose unrealistically tough projects, could not complete, and did not get the best grades. One of them also teaches and supports other students (peers and juniors) – we know he has the best concepts, but did not make the right demonstration at the end. He was graded based on the rules, but we know he deserved better. The evaluation principle does need debate.

What is the purpose of evaluation? Is it not too late to have the most important evaluation at the end, leaving no chance for the student to improve? If students wake up and get serious just before an exam (not ideal scenario, but the reality today), why not have the first difficult examination early on in the semester (i.e. early in the learning process)? This thought is discussed further, in a later section on "Elements of Learning."

11. It's still a long road

It's still early days – while the fog has cleared, building the learning culture is still a long way:

- Only the most motivated are addressed so far; a new experiment in Nov/Dec 2014 (now in progress) includes students from the bottom quadrant – some of the concepts can get shaken then, and new insights will emerge.
- More faculty guides need to join the party – building capacity is critical. These need explicit encouragement and nurturing, apart from capability enhancement programs. **New Age Teacher** initiative in summer followed by more in the semester, are meant to build capacity.

However, all of these methods, together, are showing results. As mentioned earlier, students are writing original code (big departure from the past), 150+ students are comparing well in job readiness tests (compared to less than 25 last year), so there are significant outcomes to show.

The bells that did not ring

When one experiments with diverse tools and ideas, it's very likely that many of these experiments won't work. While the successful pilots provide encouragement, it's important to recognize and document the ones that haven't worked – so far. The list in this section is a compilation of such ideas – these ideas have worked in other contexts, but have not worked in our situation; we need to understand why, and improvise. Such understanding will provide us deeper insight into meaningful education for the excluded, just like the successes.

1. ICT for learning

E-Learning, downloading educational videos, resources from the internet, are all raved about while discussing the new learning. In our context, we are trying – but it has not worked so far. Attempts at e-learning have failed, faculty videos have been somewhat accepted when from an Indian faculty, and the popular stuff in Ted Talks or other great videos have no impact. Lack of comfort with English is one reason, so also the trap of Passive Learning our students are in, this may be a consequence of "push" pedagogy.

2. Group Learning, Peer Learning

Structured, formal groups that are randomly created, don't work – in a group of four, probably two will be passengers. Less than half of the groups show participation from all team members. However:

- Informal groups work.
- Students in the top quadrant of the class are more likely to cooperate in groups than the students in the bottom two quadrants. (This observation won't be true in all contexts.)
- Group learning at the bottom quadrant works when the group is homogeneous and there is active push from faculty.

At the bottom end, there is more resistance to peer learning than to learning from faculty. (This was a surprise, and we need to keep trying). And diverse groups have less chance of success from homogeneous or self-chosen groups.

3. Competitions

Nationwide competitions demotivate rather than inspire the low-confidence students. Appropriate competitions need to be planned for each segment.

Competitions lie in a different "solution path", when motivation and purpose have been negotiated with, to some degree. Making participation in competitions mandatory does not have a lasting impact.

4. There's no silver bullet

Those involved in corporate learning are aware a three-day workshop does not build leaders, nor experts. Building technical or people leaders is hard work. We did an expensive two-day Android workshop, resulting in excellent feedback, but no outcomes. When we demanded a working Android application, they wanted another workshop! The reasons for this have been

dissected many times over, and some of our successful experiments are a consequence of such analyses.

5. Motivating Faculty

Motivating Faculty is another challenge. Around one-third have intrinsic motivation, the desire to make a difference to lives of students triggers their imagination – fortunately, we now have a critical mass to continue our quest.

The barriers

So what are some barriers we identified along the route to meaningful education?

1. Wrong metrics

The metrics need to change from pass percentage and marks to ones that measure learning outcomes. Misleading metrics have just made a mockery of marks, "first class" is achieved rather easily, and students have been misled into believing they have learnt. So we are stuck with hordes of non-learners who know not that they know not.

- ***Ineffective restrictions***

  Corporates and academics alike have restrictions on academic qualifications even when such qualifications (B. Tech., M.Tech. , Ph.D.) don't guarantee anything. In our pursuit of degrees, we have pushed out education in degrees.

On ethical failure of banks, Edward Hayes[1], says "The problem was not my co-workers' ill will or poor character. It was the standards of success for our businesses were impossibly distant from what was good for our clients". Are we in similar predicament where the defined standards expected of faculty and academia are impossibly distant from learning outcomes of students?

2. Fishermen in troubled waters

In these desperate times with gullible parents, students, and colleges running after any mirage of jobs, lots of "training houses" have sprung up, this includes giant brand names, some of them unethically masquerading as "CSR". They are peddling everything from software skills around their platforms and technologies, to interviewing skills and soft skills, and even "leadership skills". Some of them are harmless (but for the expense), but some are doing lasting damage, spreading simplistic notions of leadership or "body language", or tutoring uniform stereotypes of writing a CV or making a robotic introduction of oneself during an interview. Some others pretend anyone can become a leader by following their recipe, irrespective of background, initial capability or exposure. Then there are project sellers peddling themselves as technical trainers. The student copy code and theses, and all stakeholders pretend students have learnt. The genuine students who worked hard to create imperfect outcomes run into trouble.

Unfortunately, more fakes swamp the space and it is difficult for the uninformed to filter the genuine coaches. Besides, the fakes promise the moon but the dedicated mentors are restrained

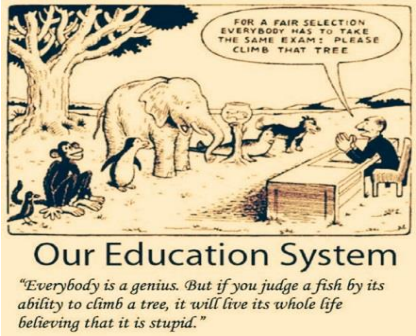
[1] Edward Hadas: "The problem with Finance", Nov 6, 2014, in www.breakingviews.com

about what they commit – and realistic goals (with loads of hard work thrown in) are less seductive than fantasy.

The gullible also get lured in by lottery-ticket like promises, and likening "smart work" as promising a quick route to the pot of gold. Hard work, perseverance, ethics and Gandhi are no longer attractive – while this approach is letting some operators make a livelihood, it is destroying a generation with false dreams.

3. Flat plate for the stork

Why do we have the same course for all, or the same exam for everyone? Abhinav Bindra would have been written off if he were coached by Desh Prem Azad, and he'd not have shot into fame as India's golden mascot if he had to fight Anand at chess. This is easy to understand, yet why do we compel the farmer's son into writing software, leaving him unworthy for both farming and IT?



4. Electricity in Jahangir's court

What if someone peddled electricity in Jahangir's court? No one would appreciate it, apart from "shock" value – and then stay away from it. In overwhelming students with mindless "leadership training"[2] and technical certifications, or pushing differential equations to students who struggle with basic number theory, what are we trying to achieve? Not learning outcomes, for sure.

5. Internet is in English

If we must benefit from the internet revolution, we must be English equipped. "Sanskrit or English" is the wrong debate. This is not a patriotism question but a livelihood question. To earn a living, we need to speak the language of the customer. The fish vendors in Bangalore speak Bengali, not because they love Kannada less but that they value business more. Among nations, China, Japan, Korea have embraced English, why are we frittering away our historic edge? Let us be an economic superpower first, then the world will learn our languages just as scholars did in the past. In Indian metros, we have young students taking Mandarin or Japanese classes. Why?

6. Tell me what is "cheese", anyway?

For concepts like "Who moved your cheese" to make sense, we first need to define what is cheese. Eighty percent of the students from a random selection are silent when you question them about their goals beyond their engineering degree. When purpose is unknown, inspiration, motivation or time management are mere theoretical pursuits. Students, however, rapidly respond to the concept of Purpose – and one can see quick behavioral changes when the purpose of Purpose is addressed.

---

[2] By all means leadership training is important. But it cannot be developed by outsourcing this to a training house providing canned content to all, irrespective of background. It requires patience, custom mentoring and time, to nurture leadership and expertise. Money is not a substitute for engagement.

7. Low trust environment

Trust is critical element if education has to result in nation building. In the absence of trust, we build controlling processes like "external exams", and attendance (with medical certificates).

8. Social Context

Neither students nor faculty can be blamed for the trap they find themselves in. They are mere prisoners of a social context that has peddled several unwinnable concepts:

- Copying is seen as a smart way to a degree, and bribing as the quick way to a job.
- Contempt for "hands on" or producing tangible outcomes; the most desirable job is one that involves sitting at a desk.
- That downloading content of someone else is equivalent to being that celebrity.
    - Anyone can copy content, nurturing a culture that creates impact is a different ball game.
- The industry misleads with suggestions of all kinds of random content that academics has now has tied around its syllabus like an albatross. So we teach mainframe, and ERP and sophisticated software tools when students are not clear about bits and bytes.
- Fads like tutoring, coaching classes, workshops, aptitude tests, abound, confusing the student even more. Most of these result in a certificate for anyone who can pay. Parents and students somehow get to believe these certificates matter and that copied projects will impress if they can just be tutored to present they have done it themselves.

9. Uninspired faculty

When one is saddled with uninspired faculty, uninspired students are a natural consequence. The intrinsic motivation for changing lives needs to be a basic attribute for the New Age Teacher. Such a teacher is comfortable with not knowing everything, encourages questions that she cannot answer, and does not mock the student who explores a domain she has not been curious about.

10. Mindless focus on jobs

"If I do this will I get a job"? Such talk kills initiative to learn. Not just that, only the crooks promise a job for any activity they peddle, so there is unwitting gravitation towards crooks.

Somehow, the single track quest for Job has removed Joy out of our lives. Can the focus shift to learning instead? Can we nurture the faith that sustained learning experiences will naturally lead to a respectable livelihood? We need to move our colleges from the job chasing factories into learning institutions. Paradoxically, that will lead to more jobs!

Must redefine what we are doing

This discussion leads to the conclusion that we need to redefine what we are doing. Let programming practice, quizzes, intense discussions, probability, Rubik's cube and Scrabble squeeze out Facebook and Likes, youtube and Kapil Sharma, out of our evenings. The link between colleges and learning, between hard work and success, between ethics and happy living, needs to be re-established. Can we redefine education as preparation for a meaningful life, for joyful pursuit of our interests, for kindling a lifelong passion to make a difference?

Elements of Learning

In this section we discuss interesting elements of learning we experienced through these pilots. While these experiences are authentic in our context, we suspect these insights could be relevant in other learning contexts as well.

1. How to learn?

There are different ways to learn, and no one rule applies to all students, or to all situations. When one is at zero-level in a domain, or one is not confident, directed coaching works best (the traditional chalk-and-talk pedagogy) for a while. However, regular assessments and assignments are still necessary to ensure the learner is active, and questioning enough. At the other end, is complete freedom with the outcomes and schedules specified. However, even in such situations, it is safer to make a project plan with no task more than a half-day. Many students are disorganized and such planning ensures no one is "lost" beyond a few hours.

Within these parameters, many other elements become important to absorb, e.g.

a. Why one must not copy

Many teachers, parents and students **wrongly believe that the aim of learning (or education) is to know the right answer.** That's a misleading approach to learning, and when one tries to achieve this in the quickest possible manner, this attitude becomes a big obstacle to learning.

Why so? Simple. Often, learning stops when one finds the right answer. When you got the right answer, you have got just one – that's it. So education could become a dreary and tiring marathon of memorizing one right answer after another, with no clue of why these are the right data, nor of how to get the next relevant information. And what happens when the next problem is outside our memory list? However, if you could learn ***the process*** of finding the right answer, you can be on top of learning, specially in an uncertain world where your existing reservoir of correct answers is fast becoming obsolete, and the quest for new answers never ends.

Copying short circuits the process of discovery, and so stymies the learner and the learning habit. That's why most tutoring produces knowledgeable imbeciles (rich in information, poor in application) completely at sea with real world situations whenever rules of the game change. The good mentor teaches the process of finding the right answer and encourages the mentee to explore, struggle, fail, and learn. This takes time, usually accompanied with pain, but proves effective in preparing the pupil for real world problems that are typically outside the syllabus.

The best students and the best institutions seem to have figured it out. The smart path seems to be to copy them. Resist the temptation to blindly copy. Copying Bangalore's solutions may not work straightaway in Balangir, just as replicating China may not work in Chennai.

b. How to copy, when to copy, when to seek help

This does not mean one must never copy, or never collaborate. In the era of Open Source, we must learn to leverage from the wisdom of the world, and meaningfully apply in our context. When knowledge exists out there, we must learn where this information exists, and how to copy.

It is also important to learn when to seek help. We must try and explore first, ***understand why we could not solve it on our own***, and then try to find out from others how to progress. When we do this, and we see someone has a superior solution, we learn about ourselves, about the limitations of our learning and our learning processes, and how we can improve. This ensures we become a better learner, and we fare better next time. Just seeking the right answer at the beginning may not lead to such wisdom.

c. How not to help

Do not provide the right answer. Instead help the student discover, and progressively move towards the truth. Do not be rigid and reject other solutions that are sub optimal. Let the student find out.

Do not be in a hurry. Don't get anxious about the student finding the perfect solution. Let the program work first, probably written in a clumsy manner; let performance considerations come later. Let them write working code, think of coding style later. Teach them to think first, let them communicate their original thoughts in any language, forget the English grammar now. We didn't learn cycling without falling down and getting scratches – programming or English communication are tougher, give them freedom and time to learn.

d. How to take notes

Many sincere students are so busy taking notes that they do not understand what was taught. They have excellent notes, copied verbatim, but they have not understood the subject. In an examination system where questions are repeated, "education" becomes a test of memory, such students will probably top in class. That's unfortunate – they worked hard, they have topped, but they have not learnt.

The right way to take notes: First try and understand the concepts being taught, and ask questions when you do not understood. Only after that, take notes, *in your own words*. This way, you have learnt, and you have taken *original* notes, so you will remember. These are *your* notes, not your teacher's.

Dr Abinash Gaya (Centurion Univ at Paralakhemundi) has evolved a good process to ensure students create the habit of taking meaningful notes. Students are not allowed to take notes while he is explaining a concept. Only after the concepts are understood, he will pause – so students can take notes.

2. Struggle

The struggle to learn is an important part of the learning process. The struggle creates a context, creates motivation, and results in non-volatile learning. It is not a coincidence that the struggle results in lasting learning; struggle results only from motivation, motivation comes from motive, and motive ensures we know what and why we are trying to learn. When we know why, and we know the hard way why some concept works, we remember. The struggle is a critical element of the learning process, those good natured parents or target focused tutors who are focused on

short circuiting the struggle, are doing the learner immense harm – if this is done for years, one could become learning disabled for life.

When one is not taught to struggle, nor on how to respond to setbacks, one is left vulnerable for life. The real world does not know you have been carefully cocooned from struggle, and presents you with crises none the less. Real crises later in life leave them devastated, unable to cope. Those who shield the students from learning to struggle are irresponsible educators, leaving the students under-prepared to meet life's challenges effectively. **Preparing for struggle is an essential element of meaningful education**.

a. Fail fast

Most average learners learn best from failure. Failing is a critical precursor to meaningful learning. The good teacher creates an environment where it is safe to fail. If failing is a necessary element of learning, why not fail early in the life cycle of learning, and learn quickly? The longer you take to fail, the more expensive learning can get. So the lesson is to design learning modules such that students fail quickly. This is more radical than it sounds. Typically, the first difficult exam happens during the 2nd Mid Sem; by then two-thirds of the semester is over – and failing becomes demotivating and expensive at this point. In a project, if the expert intervention comes only at evaluation time, it's rather late to correct the student. Even worse, in the name of making it easy for the student, we have eliminated failing from school till class 10. When they fail the first time, it may be too hard to bear, and create a learning drop out immediately. The sail through examination system in many new Universities is similarly responsible for creating citizens unprepared for life's challenges.

b. Take exam first

Many smart students are confident they have understood a concept, when in fact they have not. The first exam wakes them up. If exam leads to learning, why not present the exam right at the beginning? Often this has led to immediate motivation and accelerated learning. This has been shown to happen in our Job Readiness Programs with 2011-15 batch students at Centurion University. The author is aware these observations are in conflict with current trends in the country focused on making life easy for the student. But this paper is focused on ***meaningful*** education.

c. What kind of exams?

There are exams that test for awareness and tests of concepts. These can be taken regularly – often these exams can be taken at the beginning of the learning process. Some of these examinations should have a success benchmark of 100%, and students must repeat the exam till they reach 100%. Such a practice has now been popularized by the e-Learning systems. When automated, collation of responses in such exams can provide useful input to faculty on state of the class – and faculty learns in which areas class is strong, or class is weak. If entire class is strong or weak in a concept, faculty finds it easy to decide on actionable items. However action

items do not jump out if only half the class found the test easy. Depending on the class, one can try peer power – let the confident students teach the ones who don't know, and see how it works.

Later in the learning process on a particular theme, one needs different kinds of exams – where exploration, sharing of thoughts, building on others' thoughts, and creative thinking are more important. A written exam may not be best suited for encouraging these traits. Assignments and projects, colloquium, group brainstorms, and ideas contests will be better suited at this stage.

Written exams can be conducive to asking the "why" questions, and "what are alternative solutions" kind of questions. These tests can come right through the semester (or the learning journey).

d. Learn by doing

"Learn by Doing" is one element of this struggle. Often, the student is presented with a problem she believes she cannot handle. How does one proceed when one does not know how to proceed? That is a key attribute (**Change Agility**) to survive in the uncertain universe. When exposed to any new learning, many low-confidence students freeze, and do not know how to proceed. They need to begin somewhere, make mistakes, learn and improvise (these are the basic **Action Learning** principles). A "show and tell" faculty who takes care to demonstrate while illustrating the common mistakes and how to recover, how to take decisions, and what to learn (motivated learning - to progress towards the right path), will accelerate such learning processes of the student. Students who develop this skill will do well in life, later.

Good teachers will resist the temptation to show the right answer and speed up the "knowledge acquisition" (unwittingly short circuiting the learning process) – they need to remain patient but engaged observers while the student is struggling with a wrong problem or in a "wrong" path. The Corporate Readiness Labs in Centurion focused on some of these aspects – however, faculty who prescribed the steps and monitored compliance rather tightly have very likely inhibited the learning process even though their students might have scored high – that's the paradox of fitting in a traditional examination process (with antiquated objectives) with a practical pedagogy focused on learning outcomes.

Learn by Doing is also a joyful, active method of learning – as we have repeatedly seen in all project based learning that our students have taken up, whereas it is difficult for many students to sustain the motivation in passive learning processes. Captain Ravi (founder of Pegasus Learning Centre) talks of his experience when he started a "driving school". Student drivers were bored with the traditional driving lessons, but showed more motivation when he showed driving as a means to do their daily chores – so the learner will do household work like dropping child in school or going to the grocer, while learning driving. This is more effective, more engaging, but demands more from the faculty.

e. How much should a good teacher "push"?

While this paper tries to promote free thinking, curiosity, and empowerment of the learner, it may also seem pedantic at the same time. Should a good teacher (or parent) push her ideas or let the learner discover? In the name of delegation, many teachers (and parents) abandon responsibility; we must make sure **freedom ≠ abandonment**. It is a teacher's duty to introduce (or expose) the student to a variety of alternatives, guide them through the alternatives (preempt settling down with the first choice), so learners can make an informed choice. In the name of empowerment, we cannot abandon responsibility, saying let the student discover.

How does one keep the balance? Make sure there is open dialog and you receive critical feedback. Make sure the student has the right to choose, against your preferred alternatives. When that happens, you know you are safe. However, there must be some non-negotiables that are evolved together – the logic of those should be debated, and voluntarily accepted by the learners. Once agreed, the teachers must ensure the students keep their commitment, with doors open on re-negotiation.

Teachers must also "push" when failure is expensive or irreversible. Also, students often need to be "pushed" to keep deadlines. We need to maintain that **progress > perfection**, specially in the learning journey, else too many students get lost in complexity.

3. Backup

We have discussed Motivation or lack of it, as a major factor. One way to get across this problem is to think "backup". What's your backup plan, and what's the worst that can happen? At worst, I won't learn anything new – let's accept that, and proceed. Such thinking can liberate the student of his fears and motivate him to seek help and ask questions.

The ambitious student also needs to think backup. Some students get very ambitious, fail completely and land with a thud. That can destroy motivation, and they feel cheated, and deprived of the learning from their mistakes. Learning from failures happens in a safe environment, and the risks must not be too big. One should learn to plan in steps, and ***take risks in a systematic manner***, with clarity on where they will land should they not achieve their most ambitious objectives.

4. How to solve a problem

Students need to be taught the science of problem solving. I will explain this with an example. The problem is to write a program that multiplies two hundred-digit numbers, and displays the product. This is a challenging problem as the standard integer in a regular PC is only a nine digit number. So how do we work with hundred digit numbers? There are some thinking steps here that can be useful in many different contexts:

a. Is this problem solved in another context?

Can I solve this problem on my own, i.e. without writing a program? Of course, I can do it with pen and paper. Then, on principle, it is solvable. Now, how do I do it? My method essentially

involves breaking down the big problem into several one-digit multiplications, and lots of additions.

b. Can I solve a smaller problem?

In the above method, we essentially divided a big problem into a set of much smaller problems. Divide and Conquer!

When that is difficult, we can still begin with a smaller problem. Ok, I cannot calculate product of 100-digit numbers, but can I calculate product of two-digit numbers, *without* using the multiplication operator? Probably, I can. Then, can I try for 3-digit numbers? And so on, iteratively? This is the iterative method.

Sometimes, solving a smaller problem can just be a make-believe step, meant to give us confidence. In Thomas Hardy's 'Far from the Madding Crowd', the hospital is miles away and it's a cold blizzardy evening, but the pregnant and exhausted Fanny Robin tells herself, it's one step away, and I can go one step; and then one more step, thus she reaches the hospital. When we can't solve the entire Sudoku, can we try and fill one cell? If we can't solve the Rubik's cube, can we still try and match one side?

c. Breaking down a complex problem

Solving complex problems is always difficult. The technique is to break them down into smaller steps, and then solve one step at a time. The 100-digit product problem needs to be broken down into multiple parts:

- How do I represent the 100-digit number?
    - What are the alternatives? Which is the best alternative?
- How do I input the number?
- How do I do the product?
- How to display?

The third bullet can be broken down into more parts, of course.

d. Boundary Conditions

In their book Future Think, Edie Wiener and Arnold Brown state that problems are there in the transitions. That is true in life, in business, in society, and in programming as well. In programs, boundaries are the place to look for potential problems. Novice programmers learn this the hard way, but awareness of this can pre-empt problems. The good programmer will check boundary conditions diligently. Array overflows, incorrect initialization, tuning the algorithm for values at the edge (e.g. queue full or queue empty conditions), are examples of working at the boundaries.

Many problems occur when we play at the boundaries. If we shelve jobs for the last moment, it's more likely something will go wrong. If we are habituated to cross at orange lights, chances are we will cross Red sometime. If our money is exhausted on the 30th of every month, we could be broke sometime in the middle of the month.

Playing at the boundaries increases productivity but can be risky – learners must appreciate this.

e. How to debug

This is a key skill for any programmer. It's a critical skill for any professional as well, and the generic term is trouble shooting. Systematic debugging techniques need to be taught. Hand coding, print statements, debug breakpoints, stack trace, using a standard debugging tool, are different ways to debug, any programmer must learn step by step.

5. What should a teacher teach?

a. How knowledge abundance has led to irrelevance of traditional systems

Traditionally, the teacher was a conduit for knowledge transfer. Teacher "taught" in class, students took notes, and examinations tested for knowledge gained. This led to exams becoming tests of memory, students guessing which questions will come, and committed examiners innovating such that students cannot guess. Distortions in this system led to leaking of question papers, predictable repetition of previous years' papers, copying by students, and students' complaints on questions out of syllabus. But for the aberrations, this worked well in an era when good memory was accepted as evidence of scholarship.

In the era of the internet, knowledge abundance, ease of access, and rapid obsolescence of knowledge, memorizing has become almost redundant. Besides, no one can know everything.

It is said that the decade of the '90s created as much content as the human civilization had created from the beginning of literacy till 1990. This means the knowledge of the world has doubled in ten years, in the 1990s. Since then, knowledge of the world is doubling every five years. So trying to teach everything and running after the perfect syllabus is a game you cannot win.

Such exponential explosion of knowledge is specially true in the leading edge disciplines, e.g. medicine, genetics, machine learning, meteorology, computers and software, multidisciplinary domains like image processing, networks, space travel, sustainable energy, etc.

You know an educator or faculty is out of depth, or an education board is clueless if they are still raving about the perfect syllabus ("downloaded from MIT") or the latest content. These are problems of the last century.

b. Involve the student in faculty's own work

The best campuses do it effectively. Such an approach builds content and research data for the faculty, while accelerating learning for the student. This approach has just started in Centurion University.

c. In what order should the good teacher teach?

Rather than focus on transfer of knowledge right at the outset, the teacher will begin with:

- How to learn
- Why to learn
- Where to find what needs to be learnt

- Creating environment where one can learn
- Leading student to environments, teams, sites, or experts who can provide the learning experience to the student.
- Knowledge appropriate to the needs of learning, and the capabilities of the student.

This is not to suggest that knowledge is irrelevant – *it continues to be very relevant and critical element of scholarship*, but since it is easily available, new skills related to **connecting** to authentic, relevant knowledge become critical - pointing to the authentic sources, teaching how to find such sources, or mentoring how to make sense of all the knowledge become more important. Good knowledge is like drinking water. Water is available everywhere, but knowledge of application (e.g. how to make drinkable water available to people), or how to purify and preserve it, becomes more critical. Taking the water metaphor further, we must recognize that water is not always a good thing – in a flood or tsunami, for example. And sea water has its uses, too; and it's not too often that we look for pure, distilled water.

The "Why" of knowledge is critical, as that creates motive to learn. Once student is motivated, learning is easier – since knowledge is easily available, she will discover it on her own – only a good mentor is needed to accelerate the process of discovery, and to guide through the down moments when she makes the mistakes that are necessary steps on the way to relevant learning.

The good teacher will also provide knowledge relevant to the capability of the student. How can man send a rocket to Mars, must have different answers based on whether the learner is a five-year-old, a journalist, an energy scientist or an astronomer. Clearly "standardization" of curriculum is not the answer to issues and concepts we are discussing here. Stubbornness on creating uniformity of syllabus across capabilities, contexts, communities or cultures, reveals a complete lack of understanding of education or learning processes. A lot of students who struggle with the uniform curriculum have just been forced to receive something that is in a form alien to their context. It's not that students are knowledge-challenged, it's just that the curriculum designers are inclusion-challenged or context-challenged.

In "Hirok Rajar Deshe", an Animal Farm kind of spoof on a West Bengal government that was tinkering with the education system, noted film director Satyajit Rai has a royal rhyme meant to mislead the masses "Gyan orjoner sesh nai, Lekha pora breetha tai" roughly meaning:

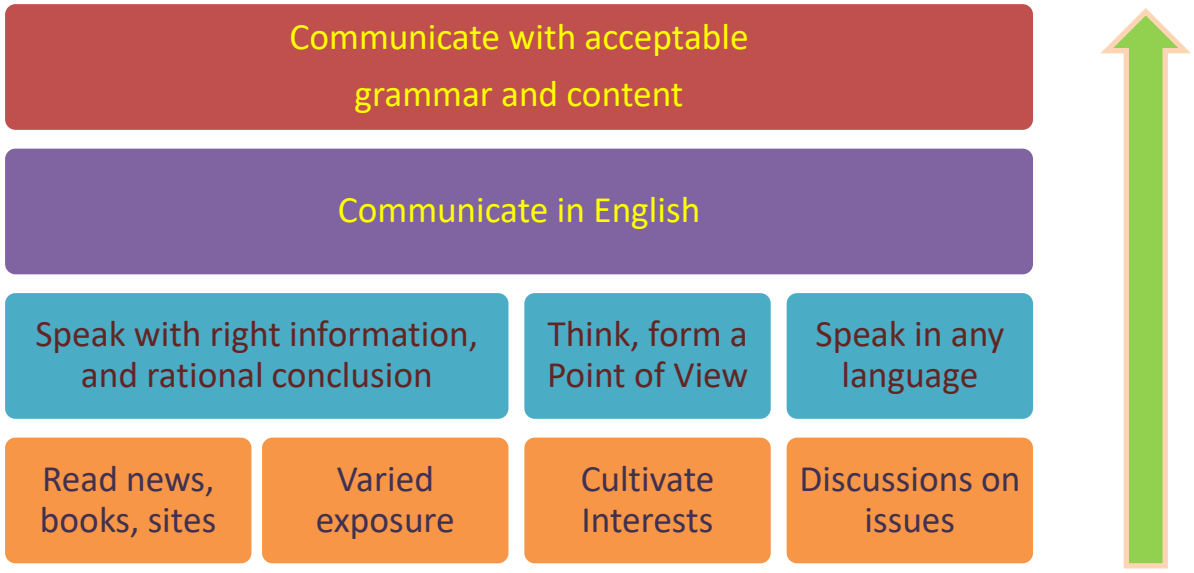
You can never gain all knowledge,
Why then go to College?
(Not a literal translation.)

Creating environment implies building confidence to learn. Where the projected goals seem too high, the guide should be open to breaking the rules and schedules, and begin all over again, maybe from scratch. Better to learn delayed, than stick to schedules and not learn!

d. Identify Core problem, and the Solution path.

The superficial evaluator will tell you the Indian engineer cannot speak English. While that could be true, that may not be the core issue.

The larger problem could be that the student cannot speak because of lack of confidence, or lack of awareness, or both. Lack of exposure and lack of awareness lead to lack of confidence. One can discover that by asking them to speak in any language. Identification of the right problem is needed if we need to find an appropriate solution – just packing everyone off to a "Spoken English" class where everyone parrots the same views on a few chosen topics is not an effective solution.

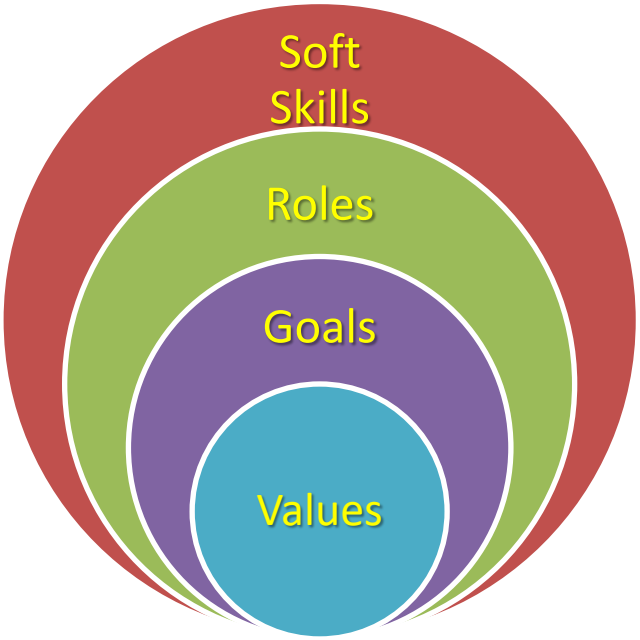


The diagram above suggests the need for a "Solution Path", there isn't one solution. The blocks at the bottom are the foundation that must be built before one moves to the higher stages. Based on the students, some blocks in the solution may not be needed – finding solutions leading to the student's growth is thus as engaging and as interesting as trouble shooting a business problem. These need well thought out custom solutions – unfortunately, many education providers just throw in some "standard and scalable" solution, irrespective of student needs. Often, we also focus on the right things at the wrong time. When the student has just plucked up courage to come and speak, we may kill all initiative by picking on grammar and pronunciation – as the diagram (on solution path) above suggests, grammar can be tackled for the mature communicator, not one who is at the bottom.

This discussion has been illustrated with a communications example, just to explain in detail. The same approach of Solution Paths is relevant for any area. For example, before you teach Android Applications, ensure the student has learnt Java first. Prior to Java programming, ensure the student is confident with programming and with object oriented concepts. And even before that, find out if student is comfortable with thinking and problem solving. In many instances, students will be thrown into one Android workshop after the other, because sales savvy

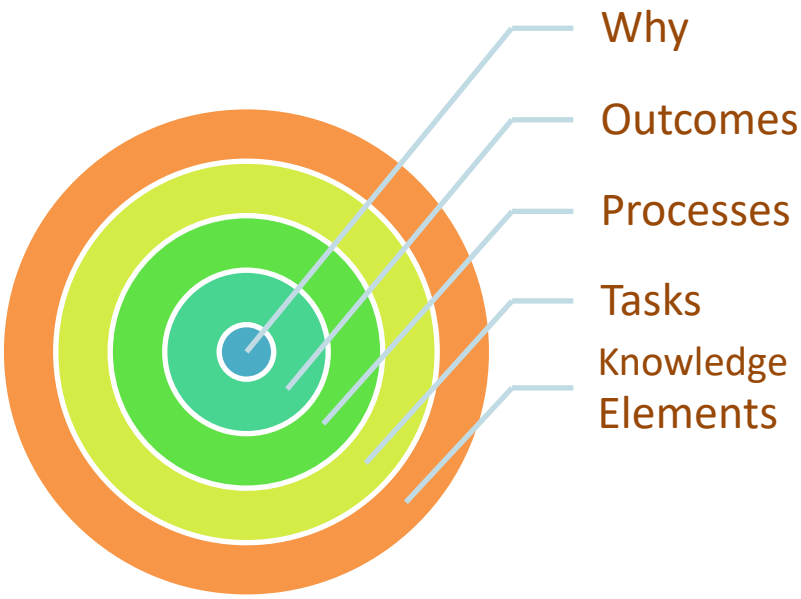
"Training" companies can persuade the college Android will provide jobs. Students keep spending money, but don't progress because they don't have the pre requisites necessary to absorb the workshop. Thinking and problem solving are more critical, and necessary foundations, but colleges and training houses don't provide solutions in that space, as recruiters don't project thinking or problem solving skills as necessary for jobs. Too many "training" centres peddle too many industry-oriented skills, masquerading them as the next step (and last step) before employment. With more than a million unemployed engineering graduates desperate for a job, these sales savvy houses sporting big brands are misleading the gullible to make a killing. These are not the only distortions waiting to be eradicated by the right laws, on our route to meaningful education.

e. Examples of where to focus



Let us assume we want to teach the Software Engineering Manager on Managing People. If we address at Soft Skills level, there will be too many soft skills to teach, e.g. Assigning Tasks, Monitoring Work, Writing reports, Effective Communication, Seeking Customer Feedback, Managing Internal Customers and Peers, Writing appraisals, Managing Resources, How to Reprimand, How to Recognize talent, Building Teams, Sharing your Vision, Mentoring, Delegation, Managing Crises, Managing Conflicts, the list goes on. On the other hand, we could just focus at the Values level – and say Be Fair to People, Keep Commitments, Serve the Customer, Try to Say Yes, Grow People. This list is easier to manage, and there isn't much to "teach" – we may be able to mentor a manager in a few hours in all of these. On the other hand, industry has designed 3 day workshops on all of above soft skills, and still the participants do not learn enough.

This is not to suggest that teaching Soft Skills are irrelevant, but being smart at all soft skills for all time is an endless process. Let us first teach the principles and values underlying these skills, and our smart learner will figure out a lot else on her own. Besides, the specifics of the skills are likely to change with time, with industry, and every time an employee changes a job, leaving them confused. When you teach values first, absorbing the Skills is much easier- it creates the fertile ground for learning all else. *Then* teach the specifics of skills you intended to teach!



If we look at the technical elements of what the Software Engineering Manager needs to teach her people, again there will be multiple knowledge elements that any software engineer must know – and that knowledge will change based on project, domain, and customer. It will be much easier to teach "Why" (this is the customer's problem), and the expected outcomes ("this is what the customer is paying us for"). Typically, many organizations and managers will ignore the two innermost circles (as they are running against time) and instead tutor employees with the three outer circles. Typically, the processes are defined and owned by the Quality Team, the Tasks by the Project Manager, and the outermost circle by the software engineer. On the other hand, the inner circles are owned by Senior Management and the CEO, are likely to be comprehended faster, ***create alignment*** across the board, and less likely to change with time.

Elements of Culture

There has been plenty of discussion on the critical importance of culture and environment in the learning process. In fact, the Learning function in Mindtree (ranked #4 globally on learning processes and outcomes by American Society for Training and Development[3]) is called "Culture and Competence", and not "Training". The following diagram illustrates Culture.

This diagram is self explanatory and an attempt to summarise elements of a learning culture described in different parts of this paper. By no means is this a comprehensive depiction of necessary elements of culture. This is just a hook for Learning providers to begin with, explore, and evolve their own culture by focusing on what works best for them.

It does convey the sense that learning cannot be boxed into a 9 to 5 job, segregated into tight compartments of subjects and one-hour classes. It also conveys not everything that matters can be easily counted. But we must not discard a critical peg of a learning culture just because we have not yet learnt how to measure or evaluate that.

While copying business principles into education has been immensely useful, we must guard ourselves against indiscriminate copying. The principles and intended outcomes of business and education can be different in some significant ways. Finding one unambiguous answer in quickest possible time is a definite objective in business, but it could stymie the learning process in exploratory stages of learning. Besides, management is about measurement and control, education is about learning – as Steve Jobs says in his Stanford Convocation address, the dots can only be filled backwards, and we need to learn with an open mind, hoping that we will figure it out in future.

[3] http://www.mindtree.com/2015-astd-best-award-winner

What should not drive culture

Fear, mere target orientation, celebration of individual performance to the exclusion of enabling and replicable, learning processes, or shying away from disabling team dynamics, or caving in to "market" compulsions must not be allowed to drive our culture.

Learning Enablers and Caveats

In this section we discuss a few learning enablers that need to be built into the psyche of students. All of these, however, come with their balancing caveats, and an overdose of these enablers will prove counter-productive.

1. Positive Attitude.

Positive Attitude is a critical element of learning, and of success, but it not the only thing, just as vitamins are critical for our body, but we can't live on vitamins alone. Unfortunately, we have a surfeit of "inspirational speakers" today who create the illusion of positive thinking as panacea. Lack of positive thinking is definitely a block, but positive thinking alone is not sufficient. It needs to be backed by understanding of self, one's purpose, strengths and limitations, and a determined plan for working on those. As discussed earlier, creating a Solution Path is critical, and Positive Thinking is one critical milestone in that path.

2. Find Your Purpose.

Positive Attitude must be followed by finding your purpose. This is important as purpose becomes the source of motivation and determination. Purpose becomes the centre of time management, and the final arbiter in resolving conflicts.

It is difficult, however, to be clear on life's purpose – and it's ok to remain fuzzy about it, or to change it when it causes more distress than inspiration. However, it's important to understand the need for Purpose, so one struggles, introspects and progressively discovers purpose.

3. Find Your Strengths.

Our strengths need to be aligned to our Purpose; and when they are not, we need to plan so that we build those strengths. Without such planning and perseverance towards building our strengths, our purpose remains as fantasy, and positive attitude merely leads to disillusionment.

4. Choosing the External Mentors

While external mentors are necessary, and they will inspire curiosity with their diverse perspectives, it is important to ensure the values they disseminate are consistent – else the student is left confused. So, focus on diverse views, opinions, facts, methods, styles, but when it gets to values, ensure the **same messages** are assimilated repeatedly, **in different ways**.

5. Take accountability for your own future.

We need to appreciate that:

- We have very little control over more than 95% of events happening around us.
- The successful people focus on the 5% they can influence or control, rather than worry over the 95% they can't.
- It is still important to understand and anticipate and plan for the 95% we cannot control, so we are prepared for them, whether they arrive in our lives as opportunity or crisis.
- We need to take initiative, be assertive about, and sometimes fight for making things happen for us – for initiating positive cycles and opening doors that lead us to opportunities.

Students must be mentored to strike a balance on this, as well. Too much accountability and too much mulling over setbacks can lead to loss of morale. Fights must be chosen well and we must not fritter energies over small issues. There are no hard coded rules here, and choices must be purpose-driven; an understanding mentor is needed to guide students through these.

6. Keep the momentum

It is often easier to initiate a learning activity, than to create the habit of persistently learning something, every day, 7 days a week, 52 weeks a year. Programming, mathematical thinking, or communication in English are such capabilities that cannot be built in a day – one has to keep the practice.

Practice is built on strong motivation ***and*** rigorous discipline in time management. Breaking the routine *just once* (always with a special reason) comes in the way of such practice. Very often well-meaning students have started programming activities or Communication activities, only to stop these abruptly when an exam or some other reasonable excuse came along.

7. Identify your core blocks, not the symptom

The industry repeats ad nauseam that our engineers lack English communication skills. That is an incorrect depiction of the problem, much like accusing a poor man of not contributing to charity. I have persisted with silent students to just speak – speak anything, in any language, irrespective of content. I discovered most students who don't speak in English, won't speak in any other language as well. Not speaking is a symptom – the cause can be low confidence, or lack of information (student has not heard about the topic), low curiosity, or inability to think or make sense out of an event. The first step, as an example, could be to read the newspaper.

8. Live your own dreams, not your parents'.

Many students are living their parents' dreams and are extremely devoted to them. It's heartening to find the family fabric remains robust in the heart of India, but students need to find their purpose (beyond serving their parents) and live their own dreams as well.

9. What kind of outcomes

It's nice to think outcomes but we need clarity on kind? Should the student get specific information in a given time? Is that the right metric, or is retention more important?

When my children were in primary school, I put them into Music classes and Drawing classes. Both learnt Music and Drawing in fast forward, and we were rather proud with their achievement

in a short time. However, this lasted only two years, the moment they were assertive enough to voice their own view, we could not send them to music or drawing classes any more.

Their teachers had succeeded in a big way in transferring lot of good knowledge in a very short time – but my children were not inspired, they dropped out once they learnt they have a choice.

This should point us towards thinking what kind of outcomes we must try and avoid and which ones we must embrace.

As educators, are we able to inspire? Inspire so much that they continue without us, without fear of penalty. Are we able to change habits – habits that lead to new or enhanced learning, and habits that last, habits that survive the engagement with the facilitator.

The author agrees these outcomes not just require deep engagement with students, but are also difficult to measure. If we find it difficult to "measure" habits, we can look at visible change over a period of time.

Learning Enablers and Caveats – parting thoughts

Why is this section relevant in this paper? We have discovered common blocks in low-performing students, and it is the teachers' responsibility to help the student remove or work around obstacles to motivation and learning – without that, the student remains low in inspiration and productivity, leading to unrealized potential.

Students are also victims of living in an illusory environment that misleads on values and beliefs – students do not appreciate the role of hard work in achieving success, don't know what is self esteem, or why copying or bribing one's way is not an acceptable way of living life. If the educator does not correct this, who will?

Elements of the Inclusive, Meaningful Education

This last section is not a complete summary of insights from our experiments in meaningful education. However, we can take these as some guidelines to begin with, with the confidence that such concepts are not mere theory but insights culled from meaningful outcomes.

1. Build positive habits and values.

We have seen that monitoring, checking, and setting rules like "don't do this" are extremely difficult to implement in a sustainable manner.

It is more effective to expose students to fun learning-oriented habits, e.g. try exposing students to Scrabble, easy to read books, Hollywood movies (with sub titles) and Quizzes rather than tell them not to be on Facebook. (Introducing Positives is the way to ease out the negatives.)

2. Beyond Chalk and talk

Very often, effective learning happens beyond lectures in the classroom. Try those.

a. Diverse Strategies

We need to figure out and design diverse strategies for diverse needs. Not just diverse pedagogy, but diverse evaluation mechanisms, diverse curricula. The goal of an evaluation mechanism, education delivery, curriculum design, or any learning process must not be "uniformity" or "scalability" – the overriding focus should be on student learning. Meaningful Learning must be the central focus and primary purpose of education. "Uniformity" or "scalability" or "measurability" must ***follow*** meaningful learning. It is important to acknowledge that uniformity or scalability do not matter if learning is missing.

b. Teaching is all about engaging.

Meaningful engagement leads to understanding the student – understand her motives, values, strengths and limitations, and tune your strategy accordingly. It should be easier for the experienced and motivated teacher to align with the students' current state than for the typical student to align to the teachers' preferred processes.

c. Attitude to make a difference.

Often, the teacher is dismissive of the student, and complains the student is learning disabled, or a disruptive element in class, or sports a bad attitude.

Most of these students can be turned around with empathy and engagement. The teacher's challenge is to lead such students into the learning path. **The teacher's success must be measured by how many students she has inspired into choosing and working on big goals, how many clueless students she has turned around**. If a teacher says she will teach only the most motivated students, she is not needed in our context of meaningful education for the excluded, she can seek appointment in an Oxford or an IIT. Empathy and engagement must be combined with expertise.

3. Beyond Content

Drop the chase for content. Good content exists everywhere, lot of it free. Find those, use those. (This is true for basic concepts, not for frontiers of knowledge.)

Also, content is not everything. Learning Delivery and Student Engagement methods, and student motivation matter more.

a. Don't copy the syllabus.

Stop copying curriculum from an elite institute. It won't work in your context. Customise your curriculum and content, custom deliver these to fit the capabilities of your students. Your students are likely to span across a spectrum of capabilities, so you have the challenge to engage with diverse curriculum, content and pedagogy.

b. Less is More. Teach Less.

Building a humongous curriculum is a meaningless, egoistic exercise that serves no one's interest. We have discussed earlier, trying to catch up with all knowledge is a losing exercise. Teach just the core, and teach these few concepts in more depth, with more real world examples, with more exercises, so students get it.

If student gets the fundamentals right and are motivated, they will figure out the rest.

c. Be mindful of capacity. Do not overload.

The best intentions and the best environment cannot deliver if these are not aligned to the capacity of the student. All students are not the same, all success formulae do not work uniformly. Every student has her physical, emotional and intellectual limitations, one has to design the environment and engagement to fit the capabilities and motivation levels of the student, and slowly build on those. Single minded focus on speed and targets can slow down the learning process and demoralize both the learner and the guide. This does not mean capacity is constant – a good mentor will stretch capacity in a systematic manner, and it can only be done with cooperation from the student.

Such open approach is necessary if we have to nurture bright, innovative nation builders from among our next generation. Uniform, memory led, last century solutions will continue to transform our children into uninspired, clueless robots, as in the past decade.

4. Motivation is the foundation.

Stop focusing on getting the best content. Focus instead on creating motivation. Look at the IITs, Cambridge, or MIT. The key difference is student motivation, build it.

5. Fixed scope, variable time.

Do not progress to the next level till the concepts at the current level are mastered. If student has not understood the level 1 concepts, there's no sense in moving to level 2, and further burden the student. Yes, you are losing time, but do you have a choice? If your academic system does not

allow time for the student to learn, change your system. What's the use of a learning system that is not designed to deliver learning outcomes?

6. Outcomes

Focus on outcomes around inspiration and habit creation – if we can build metrics around these, we can sustain learning. This was discussed in the previous section. Metrics around knowledge creation are important but may not show learning potential, and there may be issues with knowledge retention that we do not measure.

7. Exposure

Exposure to diverse people, diverse environments, diverse possibilities, diverse views and opinions, not only stretches the imagination and broadens horizons, but also inspires one to achieve more – creates motivation and builds self-confidence.

a. Nurture Aspiration

Taking students to different environment, from aspirational work environment to inspirational organizations and their purpose (e.g. Arvind Eye Care at Pondicherry), creates higher aspiration. Aspiration is a necessary driver for achievement. Often, travel and real world exposure to other places will be expensive. Bringing inspired practitioners to discuss their experiences with the students will often do the job just as well. Besides, effective use of books and the internet, and building the right communities can be affordable and effective ways.

b. Nurture Openness

Challenge their assumptions. Students must have freewheeling discussions with a variety of faculty and external professionals who harbour divergent views of the world. Facilitating exposure to variety of perspectives, often conflicting with one another, will open their minds to the reality that there's no one truth, that contradictions co-exist, that there is no fixed formula for success. They need to chart their own path, make their own mistakes, and find their way.

c. Nurture Diversity

It is important for the student to live in relative uncertainty, in the midst of diverse views, without the comfort of having discovered the truth. Too many managers and professionals are uncomfortable with ambiguity, and **scared to be seen as indecisive** – so they jump to a decision too soon. This is not a winner's trait in an uncertain world, and the ability to live with multiple views and often, the need to defer decision making or keeping the option to retreat, must be nurtured.

d. Nurture Turbulence

Exposure to assignments that shake them out of their comfort zones, introducing to variety of faculty with differing styles, are necessary to stretch the self-confident student. The confident students must be forced to challenge their beliefs, and stretch their aspirations. In our experiments, we have had participation from multiple faculty with diverse styles, from Ajit Pasayat to Rakesh Ray and Sasmita to Sangram, students have experienced different approaches. Besides, of course, the variety of external experts they interact with.

8. Segment the syllabus, compress the time.

One critical insight through the learning experiments is the power of focusing on one subject, one theme, one concept, at one time. Right through the experiments, we focused on a few learning objectives delivered in intense schedules.

- Students wrote a Sudoku Solver (involving a few hundred lines of code) in ten days, long days and evenings that stretched beyond midnight. This output does not happen otherwise in even two years. Top quadrant software engineers from many good colleges cannot write a Sudoku solver.
- Students designed Entity Relationship and User Interface for their projects in one and half days. In normal coursework, they don't deliver as good output even in 6 weeks.
- Students learned more in two job-readiness centered immersion programs (3 days and 4 days) than they learn otherwise, over months. More than 50 students delivered nationwide high scores through this pedagogy, compared to one tenth that number through conventional methods the previous year.

7a. What made the difference?

In a conventional semester, there are three problems:

i. Multiple subjects are taught at one time – where does the student focus?[4] Something suffers.

ii. More important, the regular class is over in an hour by the time the student is warming up to a concept. And then the next class begins, the student is gradually getting engaged with the new subject and different concepts, and by the time she is into it, the bell rings and it's time for the next.

iii. Through this routine, the student is not in control. She remains a passive spectator in plans well laid out for her. The best of students will listen well but do not have time to relate, must focus on comprehension or on taking notes and usually choose the latter (as notes are tangible output).

Through our experiments we described here, we had one clear primary focus: Programming (C or Java), or Aptitude Tests preparation, or Android Applications. True, many diverse elements were thrown in, but those were meant to engage the student better by providing relevant variety, and the student often exercised the choice to reject those and focus on the core.

The modern semester delivers 4 courses in 4 months. This was traditionally delivered to the most motivated, best performing students and they coped. Their survival through this design is more a proof of the resilience and motivation of the best students than the effectiveness of this system.

For the less capable student, focus on one objective at a time will be more effective. Can we deliver one course a month, for four months, rather than four courses simultaneously in four months? Can we redesign an ineffective system into one that is shown to generate learning outcomes, and with more joy?

[4] This is not an issue with the most motivated or the most capable students – hence it works fine in the elite schools.

The journey continues …

This paper is a "work in progress" document. The experiences are being shared, even as the learnings are unfolding, and there has been significant learning and fine tuning of tracks and beliefs while this exploration is on. Some of these experiences have now been repeated in multiple contexts – with different faculty, different students, different venues and different learning objectives. All show learning outcomes.

While this paper is being written, a more ambitious initiative is in progress. Students of varied capabilities and motivation are participating in a focused learning excursion, to write Android applications. Different teachers in a different environment are guiding a variety of students – the faculty has taken care to create **diverse goals, based on capability** – and that is an important element of success. Keeping same objective for all students will either bore the most capable or exclude the least prepared.

One cannot wait for this "research" to be over, before one tries to implement in a bigger way. Students' lives do not stop for us, and the semesters blow by, irrespective. Thus exploration, insight, implementation and scale, must go hand in hand, showing incremental outcomes as we progress, with an openness to recognize discordant issues in time and acting on them. Citing ideology, rules or policies to refuse change even when we see the students do not get the desired outcomes, is a crime against the future of humanity. In this century of rapid explosion of knowledge (and rapid obsolescence, too), we need to act even as we learn, else the opportunities will pass us by, without respect for our erudition and growing wisdom. This means we as teachers must be comfortable with the principles we teach and apply them at work, e.g. openness, struggle, or embracing failure as a learning tool.

Thus, in many ways, this is not an epilogue, but an acknowledgement of incompleteness in our wisdom. Consequently, this paper does not have a "conclusion" even though each pilot or initiative results in some conclusions that help us progress to the next milestone.

Acknowledgement

Any successful experiment is modeled after intense efforts, creativity, and endorsement of a large set of people. Some of these were path breaking in the direction of effective education models, and we are bound to have a large number of contributors towards its success. Just a few of them are listed here:

1. Centurion University, the platform where these experiments were designed, piloted, and learnings synthesized.
2. C. K. Dwarkanath: Partner, Prime Implementor, and Co-designer in creating the successful pilot at Pegasus Learning Center in Tamil Nadu, north of Pondicherry.
3. Captain Ravi: Founder and Principal Educator at Pegasus, naturally attracted to innovations in education.
4. Sasmita Nayak, Ajit Pasayat, Rakesh Ray, Sangram Swain, Sabyasachi Mohanty, Biswajit Samal (Assistant Professors and Project Guides, and sponsors/guides of internal projects), and Dr Manas Senapati (HoD), Department of CSE, Centurion University.
5. Dr Haribandhu Panda and Mr Sibakripa Bose: Mentors for these Learning Experiments.
6. Uma Maheshvari: Change Catalyst and Behavioral expert at Mindtree Limited.
7. Col Suresh: Faculty and Inspiration, Pegasus.
8. Rajesh Kumar R.: then Technical expert at Happiest Minds Technologies, Bangalore.
9. Nagarajan S.: Professor at IIIT, Bangalore and Technical Director at NetApp, Bangalore.
10. Sunil Kand: Founder and Head of Peak Time Infosystems, Bangalore.
11. Ananda Rao Ladi: Co-Founder of Parablu, earlier Senior Vice President and Head of Asia Pacific, Mindtree, Bangalore.
12. Ramachandran N.: Co-Founder of Yushu, earlier VP and Head of Quality, KM and Culture and Competence, at Mindtree, Bangalore.
13. Dr Ragunath Govindachari: then Vice President and Head of Mindtree Research, Bangalore, now Co-founder of IOT startup.
14. Faculty and Staff at Pegasus, Tamil Nadu
15. Kumar Devadutta: CEO, Innovadors' Lab
16. Prashant Mehra: Head of Mindtree Foundation, and architect of "I Got Garbage", technology platform for Urban Waste Management
17. Pramod Saini: Co-Founder, iCalibrator.
18. Subhanjan Sarkar: Film maker, technology evangelist, and Founder-CEO of Showhow2, the start up that is redefining Customer Support
19. Lovleen Bhatia and Kapil Tyagi: Co-founders of Brain4ce, and the very successful e-Learning portal edureka.in
20. Amit Sharma: Android faculty at edureka.in

References

1. Career Fountain Blog on “Teaching the Millennials”, www.cutm.ac.in
2. Appendix 1: Ten levels of Programming Capability.
3. Steve Jobs’ Stanford Convocation Address.
4. Education is 360 Degree, address by Kalyan Banerjee at JNTU, Kakinada, May 2014
5. Teaching in the Abundance era, Kalyan Banerjee, Oct 2012
6. http://www.breakingviews.com/edward-hadas-my-career-in-the-financial-jungle/21172302.article, Edward Hadas in “The problem with Finance”, Nov 6, 2014

12th December, 2014.

Appendix 1

Ten levels of basic programming capability

Level 1

1. Write a program to add two numbers.
2. Write a program to multiply two numbers.

Level 2 (Simple "if else" programs)

1. Write a program to check whether the entered number is an odd number or not.
2. Write a program to check whether the entered number is divisible by 7 or not.
3. Write a program to find the largest number among three numbers.

Level 3 (Simple "loop" programs)

1. Write a program to print the series from 1 to 10.
2. Write a program to print first 10 odd numbers.
3. Write a program to print all odd numbers from 15 to 80.
4. Write a program to find out the factorial of a number. (optional)

Level 4 (Simple Array programs)

1. Write a program to enter any 10 integer numbers in an array and print it.
2. Write a program to create an array with first 10 even numbers.
3. Print all elements of an array till you encounter zero.

Level 5 (General integer value programs)

1. Write a program to reverse a number.
2. Given a number, check if 8 is one of the digits in that number.
3. Given a number, say whether it is power of 2.

Level 6 (String programs)

1. Find the length of a string.
2. Reverse a string.
3. Check whether the string is palindrome or not.

Level 7 (Structure program)

1. Write a program to create a structure of a Student to store the student name and marks of 3 subjects.
2. Find the size of above structure.
3. Create an array of above student structure.
4. Find the average of marks of all students in that array of structure.
5. Print name and student who has highest marks. (optional)

Level 8 (Matrix program)

1. Find the highest element in a matrix.

Level 9 (Function program)

1. Write a program to swap two numbers using function.
2. Write a function for factorial of a number.
3. Write a function prime which returns 1 if it is prime, 0 otherwise.

Level 10 (Higher standard programs)

1. Write a program for bubble sort.
2. Write a program to print first 10 Fibonacci numbers.

Note: these levels were modified marginally in 2014, after the experience with the 2013 class.