Agile High Technology Course Development in the Enterprise

A Technical Paper prepared for SCTE by

Oliver Jojic
Distinguished Researcher
Comcast
1701 JFK Blvd, Philadelphia, PA
703 677 6067
Oliver_Jojic@comcast.com

Mike Baker
AI Program Director
Comcast
1701 JFK Blvd, Philadelphia, PA
267 207 5865
Mike_Baker@comcast.com

Table of Contents

List of Figures

List of Tables

1. Introduction

This paper describes an effective agile process for teaching *technical* subjects in the workplace. For example, data security, machine learning, cloud computing are technical subjects.

With the advent of Big Data and with today's breakneck speed of technical innovation it is more important than ever to provide the technical workforce with continuous education.

But many questions need answers: how frequently should a full-time employee be diverted from work due to their education? What budget should be assigned to the continuous education of technical employees? How can the teaching material be always kept up to date and relevant to the company?

As we were tasked to teach machine learning to a large portion of the software engineering workforce, we had to ask ourselves these questions and more, and after a few iterations we reached our current process which seems to "just work": no extra budget and an average student's NPS (net promoter score) nearing 100%. The process uses in-house technical experts to design and teach short lessons, and it uses past graduates to act as mentors for new students. In addition, the process has the originally unplanned benefit of favoring networking among employees from different parts of the company.

Encouraged by our successes we recently applied the same teaching process to new technical domains: Data Science and Full Stack DevOps, again, it seems to just work. No extra budget and an average student's NPS in the 90s.

In this paper we answer the above questions and more, we detail our teaching process, and we share some quantified results.

2. Dialectic of Teaching Alternatives

As we embarked on our task of teaching machine learning to a large fraction of the software engineering workforce, we had to evaluate the then current legacy teaching structures. This evaluation led us to attempt a somewhat formal description of teaching alternatives, with the purpose of identifying what would work best for our task.

2.1. Legacy Teaching Structures

Two very distinct teaching programs existed when we started.

The first was akin to a university course but only for the selected, high-performing employees. Over the span of nine months, a carefully selected few employees are gathered in a classroom with an external adjunct teacher with a curriculum that is optimally designed to match Comcast needs. The rather trivial problem with this program is that it is not scalable, not only on the "horizontal" view of fulfilling the education of many employees, but also on the "vertical" view of the continuous education of one employee. Nevertheless, this is a successful and coveted program, and it is being continued today.

The other was open to all employees: it is a selection of online teaching materials. The problem with this totally async approach is that students that can learn on their own don't necessarily need this pre-selection, while students that would need some help do not get it.

Table 1 – The two Legacy Programs

	"Selected Employees" program	**"Open to All" program**
Instructor led	Yes	No
Duration	9 months	Async
Budget	Expensive	Cheap
Scalability	Low	High
Effectiveness	High	Low
Resilience to changing needs *	Ok	Chaotic

** The AI domain is rapidly evolving. How quickly can a program be updated to include the teaching of new concepts or tools*

The stark contrast between these two legacy programs incited us to attempt to formally describe teaching alternatives.

2.1. Attempt at Formalizing the Alternatives

This section attempts to be somewhat formal in listing teaching possibilities. Having a clear view of the pros and cons of all alternatives is helpful in designing the right teaching process.

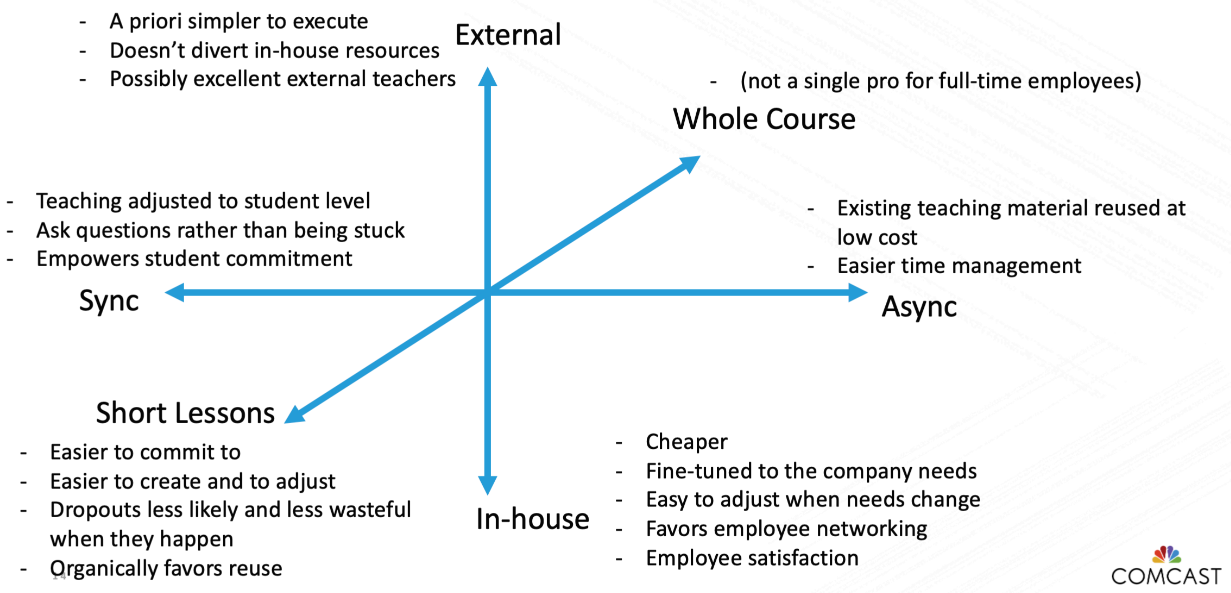

- A priori simpler to execute
- Doesn't divert in-house resources
- Possibly excellent external teachers
External
- (not a single pro for full-time employees)
Whole Course
- Teaching adjusted to student level
- Ask questions rather than being stuck
- Empowers student commitment
- Existing teaching material reused at low cost
- Easier time management
Sync
Async
Short Lessons
- Easier to commit to
- Easier to create and to adjust
- Dropouts less likely and less wasteful when they happen
- Organically favors reuse
In-house
- Cheaper
- Fine-tuned to the company needs
- Easy to adjust when needs change
- Favors employee networking
- Employee satisfaction
COMCAST

Figure 1 – Dialectic of Teaching Alternatives

2.2. Sync / Async dimension

Technical subjects are complex, and students will benefit from all the help they can get.

Sync means teacher and students are interacting in real time. Async means the student is consuming a previously prepared material. The async version may include some domain expert helping the student, for example by answering questions on a messaging app. We will call such an async helper a "soft mentor", or a mentor for short.

Note: in our use of sync/async *the interaction* between the teacher and the students is key. If the information flows in only one direction (teacher to a large class that is not expected to interrupt) then it practically matches async teaching.

Sync: the teacher can adjust what is being taught to the level of the students.

Sync: students do not get stuck because they can ask the teacher for clarification.

Sync: empowers student commitment.

Note: student commitment is key, otherwise continuous learning risks of being a farce, a mean for the student to check some educational boxes. The teacher empowers student commitment by interacting with every student in the class, by asking short, simple questions in some round-robin fashion.

Async: time-management is easier for full-time employees.

Async: existing teaching material can be reused without extra costs.

Conclusion: both sync and async have merits. Possibly both can be used for the successful continuous education of full-time employees: a short sync class followed with a mentored async practice.

2.3. Short lessons / Whole course dimension

Throughout our younger lives we have all been educated with *whole courses*: long duration, well designed material that teaches enough facets about a new subject for the learner to get a good logical understanding of the whole. Yet, long duration courses are clearly not ideal (even incompatible) for full-time employees. So, we explored the alternative: *short* duration *lessons*.

An example of a whole course would be "Machine Learning". An example of a short lesson would be "Random Forests".

What is short? Our experiments showed that up to 8 hours spread over one week is favored by full-time employees.

Short lesson: easier to commit to for full-time employees.

Short lesson: easier to create and to adjust.

Note: we mentioned earlier that the AI field changes at breakneck speed. For example, it is easier to create a new lesson about Attention and Transformers rather than adjusting a well-designed coherent whole course with the new content.

Short lesson: student dropouts are less likely and are less wasteful when they happen.

Short lesson: organically favors reuse.

Note: a whole course about machine learning and a whole course about data science might both have a section about Python. The content about Python could be almost identical, but there is no easy way to reuse here. Instead, with a collection of short lessons, the Python lesson is already there and available: organically favoring reuse.

Short lesson: scalable.

Note: spawning the same short lesson multiple times throughout the year to reach more students is doable because it is short. Spawning a whole course multiple times is hard and limited because it is long duration.

Conclusion: For full-time employees we did not find a single *pro* for the *whole cour*se alternative. In the enterprise, short lessons are always to be favored instead of whole courses. The disconnected nature of a long list of short lessons can be resolved by having the company's educational website somehow link the available short lessons into coherent wholes.

2.4. In-house / External dimension

The course material and its teaching can be developed in-house, or it can be selected from external sources.

Note: The analysis and conclusions about this In-house/External dimension depend heavily on the prior selection between *short lesson* and *whole course* (described in the previous section). Here we assume that the teaching solely consists of short lessons. A few weeks per year an in-house domain expert will have to spend up to 8 hours teaching. This is done without much impact on the employee's regular activities. The design of the short lesson is somewhat more involved, but it happens only once. Sometimes the domain expert selects an existing online source material, in which case the design of the short lesson is even less time consuming.

External: a priori simpler to execute.

In-house: cheaper.

External: doesn't need to divert in-house resources.

In-house: fine-tuned to the company needs.

In-house: easy to adjust the teaching material when the needs change.

In-house: favors networking among employees from different parts of the company.

External: external adjunct teacher can be excellent at teaching.

In-house: sense of satisfaction for the domain expert employee thanks to being selected to teach to the whole company.

Note: after running our agile teaching process for a while, we noticed this sense of pride forwards to the employee's leadership as well.

Conclusion: while there are benefits to both, a company should favor the in-house alternative. For this to be feasible the teaching material should be restricted to short lessons.

3. Student's Practice

To be a practitioner of a new technical skill, a student learns the relevant technical material, but also spends significant time practicing the new skill. Anything long duration that is imposed on full-time workers is best implemented in async mode. Therefore, after completion of a short sync lesson, we do encourage the employees to practice their new skills, but the frequency and total duration of the practice is up to the employee.

Whenever possible, a short sync lesson should be accompanied with async practice exercises (sometimes this can be a simple link to already existing online material).

3.1. Capstone projects

A Capstone project is a bigger, more significant mean of exercising a new skill. Each student decides when and how frequently they will work on their Capstone project. There is no need for a time limit.

The completion of a Capstone project should be a formal event so that the employee rightfully feels a sense of accomplishment. This can be done with a ceremony, grouping together multiple recently completed projects, where each graduate describes their project to the group. At the end a diploma is handed to each employee.

Capstone projects also give structure to an otherwise long list of disjointed short lessons.

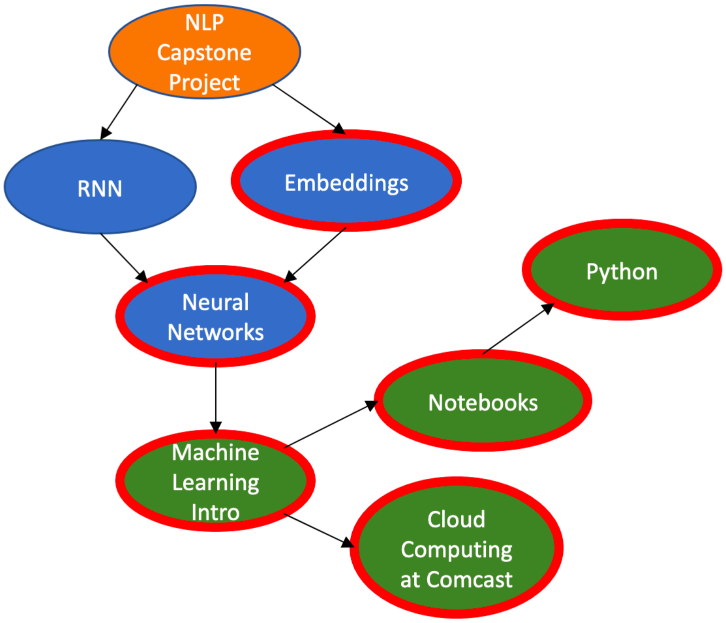


Figure 2 – A Capstone Project and its Soft Prerequisites

Each oval is a short lesson. The arrows represent soft prerequisites. Soft means that an employee does not have to enroll in a short lesson if they already know the subject. On the company's educational website, a Capstone project is roughly equivalent to a short lesson. The entry should list all the soft prerequisites.

Students can enroll to the Capstone project and their duty is to complete the project and to create a presentation of their work to be shared during the graduation ceremony.

While practicing, whether with a Capstone project or with simple exercises, the student can get help from mentors.

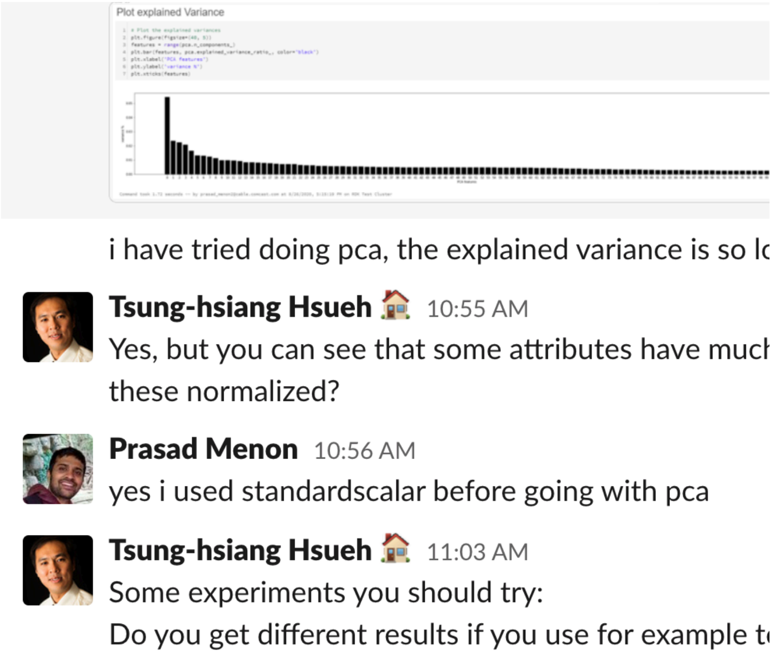
3.1. Mentors

As with any new knowledge, students should practice what they learned if they want to become proficient. Practicing alone can be hard, as the chance of getting stuck is high when dealing with something new. Having mentors to help students practice is extremely useful. The word mentor here should be understood as a soft mentor, or a learning assistant.

This is great in principle, but how is scalability resolved? With continuous learning there is a continual flow of active students. Our solution to the scalability of mentors is to ask select graduates if they are willing to mentor future students. Through time, as the number of students grow, the number of available mentors grow as well, resolving the scalability issue.

A company-wide messaging app (Slack, MS Teams) is used to interact asynchronously between practicing students and mentors. Domain experts (i.e., the teachers) sometimes join a conversation as well.

This process has the originally unplanned benefit of favoring networking among employees from different parts of the company, of building a community that is excited by the same technical domain (machine learning in our case).



Plot explained Variance

```
# Plot the explained variances
plt.figure(figsize=(40, 5))
features = range(pca.n_components_)
plt.bar(features, pca.explained_variance_ratio_, color='black')
plt.xlabel('PCA features')
plt.ylabel('variance %')
plt.xticks(features)
```

Command took 1.72 seconds -- by prasad_menon2@cable.comcast.com at 8/26/2020, 3:13:19 PM on RDK Test Cluster

i have tried doing pca, the explained variance is so lo

Tsung-hsiang Hsueh 🏠 10:55 AM
Yes, but you can see that some attributes have much
these normalized?

Prasad Menon 10:56 AM
yes i used standardscalar before going with pca

Tsung-hsiang Hsueh 🏠 11:03 AM
Some experiments you should try:
Do you get different results if you use for example t

Figure 3 - Example Interaction between a Student and a Mentor

4. The Agile Teaching Process

The Agile Teaching Process implements the best practices discovered in earlier sections:

- Short lessons rather than long duration courses
- Sync teaching followed with async practice
- In-house domain experts encouraged to teach their expertise
- Past graduates encouraged to mentor new students

The process is best explained by following the journey of all parties involved: administrator, leadership, students, in-house domain experts, and mentors.

4.1. Administrator journey

The administrator typically belongs to the education side of the company. The administrator role is key to ensure that the agile teaching process is successful. As such, the administrator, directly or indirectly, must be in contact with all participants. Administrator responsibilities are highlighted in the subsequent "journey" sections and are not being duplicated here.

4.2. Leadership journey

Leaders must be convinced that it is worthwhile for their expert employees to be sometime diverted from their daily activities to teach their expertise to the rest of the company. This radical meshing of activities across orgs with well-defined boundaries is best started by the administrator introducing the new process as a long-term pilot.

The administrator should provide the leaders with a quarterly update.

4.3. Student journey

Employees are made aware of the available short lessons by browsing the company's educational website. The administrator ensures the website is kept up to date.

How frequently should a short lesson be instantiated? This depends on its popularity: the administrator monitors the number of students that are registered to a class. To maximize the effectiveness of sync teaching a class should be limited to a maximum of roughly 25 students. The administrator will increase or decrease the frequency of a class based on the number of registered students. But, prior to this adjustment, the administrator should verify that the number of registered students indeed reflects the popularity of the class rather than being due to the students not noticing the class; this is particularly true for new classes. The administrator should maintain a growing mailing list of potential students. When the number of students being registered to a class is low, before reducing its frequency, the administrator will email a reminder about the class. In our experience, this often results in many more students registering to the class, proving that the lack of popularity was due to lack of employee awareness rather than a class being instantiated too frequently. This description implies that the teachers must show some flexibility with their calendar planning; this is an agile teaching process, and the administrator should from the start make everyone aware that some agility is to be expected.

For full-time employees, to be able to benefit from sync teaching, the class duration must be short. From our experience, a maximum duration of 8 hours spread over one week works well. Always favor spreading the learning hours over multiple consecutive days. For example, teaching 2 hours during 4 consecutive days is to be preferred to teaching 8 hours within a day. There is only so much a brain can learn in a day.

Courses are recorded allowing students to play or replay the lessons in async mode.

To learn new skills, practice is key. Many short lessons should be accompanied with exercises for the student to complete asynchronously after the class is over. The student is free to do it at any time. Jupyter notebooks are perfect mediums for holding the exercises (and for holding the course as well). If they have difficulties during the completion of the exercises, a student can ask mentors for help. A company-wide messaging app is used for the interaction between students and mentors. The student asks a question and asynchronously a mentor, or more, answers it.

Students can more thoroughly practice their newly learned skills by completing Capstone projects. Capstone projects have their own entries in the company's educational website. A student must register to a Capstone project. There are no teachers for Capstone projects. Instead, the Capstone project's entry in the educational website describes the required knowledge by listing the relevant short lessons. These lessons are only "soft" prerequisites, e.g., if an employee already knows Python, they do not have to register to the Python prerequisite class. There is no time limit for the completion of the Capstone project, it is up to the student. Students working on a Capstone project will frequently communicate with mentors through the messaging app. In addition to completing the project, a student must also prepare a presentation describing how this was done. Once enough Capstone projects are completed, the administrator organizes a graduation ceremony where each student describes their project to the group and at the end receives a diploma.

4.4. Teacher journey

In this agile teaching process, all teachers are in-house domain experts. It is the domain expert, in coordination with their leaders, that decides what new lesson should be developed. They know what knowledge is needed for the company technical employees. They are the first to notice the advancements in the state-of-the-art or the emergence of new, better, tools. The administrator is responsible for reminding domain experts and their leaders of the company's technical educational needs and of their duty within this agile teaching process. Our experience shows that most domain experts are excited about the opportunity of teaching their art across the company and their leaders are honored by their group being acknowledged as an important source of knowledge within the company. This motivation can be boosted further by explicitly listing the teaching of one's art in the yearly goals of all technical employees above a certain level. We do believe that this process brings a sense of purpose, builds a community, and ultimately strengthens retention of employees.

Once a new lesson has been agreed upon, the administrator helps the expert commit to the design of the lesson and of the accompanying exercises by setting up a schedule to which the expert can comfortably adhere to. The total duration of this schedule varies but it takes around 3 months on average from the original idea down to its completion.

During the course design a specialist helps the expert with good education principles. Two of the most important educational principles are 1) the teacher is not in the class to bombast his knowledge but rather to adjust it to the current level of the students and 2) the teacher maintains students' attention by continuously forcing them to participate with short simple questions in some round-robin fashion.

In our experience failed teachers are rare, as proven by an average NPS from students that is above 90%. When it happens, the administrator gently discards the failed expert teacher and finds a replacement.

4.5. Mentor journey

When practicing new skills, it is crucial to have some assistance to help avoiding “being stuck”. Our use of the term *mentor* in this paper means exactly this: a practice assistant.

The agile teaching process ensures that every student has helpful mentors during practice. How is that scalable?

After graduating a student may be contacted by the administrator about the possibility of becoming a mentor to future students. It is the administrator’s responsibility to notice which students have superior skills, as poorly done mentorship could be counterproductive. The administrator will consult with teachers to confirm the appropriateness of a student to mentorship.

Once selected, and willing, a past graduate is promoted to mentor. The duty of a mentor is simple: monitor the company-wide messaging app for student questions and answer the question and be generally helpful. Thus, the total number of mentors keep growing, resolving the scaling issue.

While not necessarily intuitive, this mentorship by past graduate, in our experience, is working very well. A few star mentors are quickly answering most of the questions (even though the interaction is not expected to be real time) and the students are very happy with the help they are receiving.

In our experience, most contacted graduates were excited to become mentors.

5. Pilot Results

The last 4 years were exciting as we kept fine-tuning the methodology for teaching machine learning at Comcast. In this section we share some numerical results.

5.1. Ability of in-house domain experts to create teaching material.

The pilot switched to the current version (in-house experts preparing short lessons and teaching them) just 2 years ago.

Table 2 – Size of the Catalog of short Lessons

Number of Short Lessons in the Catalog (as of this writing)	Average time to add a new Lesson
25	< 3 months

5.2. Ability of in-house domain experts to teach

The table below shows the average students’ net promoter score in the last 3 months. The score is quite stable over time, between 90 and 94%.

Table 3 – Student NPS

	This Agile Process	Legacy "Democratic"	Legacy "Best and Brightest"	Education global
Student NPS	91%	Not tracked. Mainly constructive feedback	85%	55%

5.3. Ability of the agile teaching process to scale to more students

Yearly course completion or students per year:

Table 4 – Students per Year

	This Agile Process	Legacy "Democratic"	Legacy "Best and Brightest"
Students per year	500+	100+	30

5.4. Retention

Anecdotally the fraction of mentors and teachers that leave the company is less than the average of all employees.

6. Challenges

For convenience we gathered here the challenges to this agile teaching process.

Administering an in-house teaching process is demanding. The tasks include: consulting with business leaders and experts about courses that should be developed. Finding an expert that is willing to design and teach the course. Tracking the progress of the course design. Ensuring that employees are aware of the list of available courses. Sharing results with business leaders.

In house expert resources are diverted from their regular duties. This is facilitated by leadership agreeing that knowledge sharing is an official part of an expert's duty.

Assumption that domain experts are apt at teaching. This was a major unknown at the beginning of our pilot, but the results are positive. At least in technical fields, most experts can teach efficiently (based on the student NPS scores for experts vs. external faculty classes). Still, we had to cancel about 10% of experts due to their inability to teach.

A minority of employees prefer async training. An easy remedy is to record the classes and let each employee decide whether they prefer to join a sync class or to listen to the async material.

What happens to their class when an expert leaves the company? It happens, and the administrator must find a replacement expert to teach the same class. The replacement expert is made aware that the course material is now theirs and that it is within their rights to adjust the material to their teaching style.

7. Business Impact

For convenience we gathered here the impact to the business of this agile teaching process.

Mastering data driven decision making. Some parts of our big organization have business leaders that push for more data-driven decision making. The progress toward that goal was slow and the reason might have been due to the engineering workforce, on average, having a lacking knowledge in AI/ML. After three years of applying this agile teaching process we have boosted the knowledge of machine learning to more than 500 employees. Many of them are now genuinely excited by this technology, and now push for data-driven decision making from the bottom-up as well. Compared to a few years ago, our company is noticeably more agile in extracting knowledge from data. It is hard to quantify how much of it is due to our application of this agile teaching process.

Low dollar cost. Not having to pay for external faculty is a cost saving. Even async teaching material can be expensive. With this agile teaching process, the dollar cost is replaced with the in-house experts being diverted from performing their regular duties. This "cost" is welcome if leadership agrees that knowledge sharing is an official part of an expert's duty.

Scalability with no cost increase. If it is in the business interest to spread some knowledge quickly, or simply if a class is very successful, the expert will teach more frequently, say 8 hours per month instead of 8 hours twice per year.

Expert employee retention. While anecdotal (this one pilot cannot be considered statistically significant) it seems that designing and teaching a course has beneficial effects for expert employee retention. If true, it is likely due to the combination of two psychological effects: the pride of being selected to design and teach complex material to the rest of the company; as well as the significant increase in connections, sometime even new friendships, between a teacher and their many students.

Significant gains in NPS performance. Based on the students' NPS ratings, their least favorite method of learning is asynchronous, where the company selects some async training material (sometime quite expensive) and the student is left on their own. Next up, with a significant NPS jump, is external faculty teaching. Lastly, the agile teaching process described in this document with its in-house expert-based teaching, has the highest NPS scores.

8. Conclusion

In this paper we described an effective agile process for teaching technical subjects in the workplace.

This process is scalable, it closely tracks the educational needs of the business, it empowers the continuous education of technical employees, it builds a community with the same passion, and it does not require any additional budget.

How? By leveraging the expertise that already exists within the company.

In-house domain experts design and teach short lessons. Because the teaching material is being built by the company's employees, it is always fine-tuned to the company needs. Because each lesson is short, it can easily be scaled up to more iterations for classes in high demand.

Students are grateful because they have a sync teacher and can ask questions but also because the course material is cut in small lessons and easier to fit within their busy schedule. But learning is only half of the path toward proficiency. The other half is practice, and practicing new skills is hard. When to practice is completely up to the student. Yet, the student can still get async help from mentors through a company-wide messaging app. Select graduate students are asked in turn to become mentors, making the whole process scalable. Student's satisfaction is clear given that their average NPS is above 90% promoters.

Lastly, and this benefit was not originally planned, we found that this process forces the interaction between experts, students, and mentors and organically builds a community with the same passion and ultimately is likely to strengthen employee retention.

Abbreviations

SME	Subject matter expert. For clarity, in this paper, we use "in-house domain expert" instead
Mentor	Learning assistant
Sync	Teacher and students are interacting in real time
Async	Student is consuming a previously prepared material
NPS	Net Promoter Score: Students rate a class from 1 to 10. Total = number of student ratings Promoters = number of ratings equal to 9 or 10 Detractors = number of ratings equal to 6 or below NPS = (Promoters – Detractors) / Total