1 Project proposal

Bucheton and Soulé have described the act of teaching as a multi-agenda game of postures requiring good preparation and excellent micro-decisions [5]. In their model, they identify the crucial roles of the teacher in controlling the cadence, the atmosphere, the scaffolding and the relationships during the class, and how each supports the learning objective. And yet many of the teacher's most time-consuming tasks do not take place in the classroom [4]: good preparation and strong follow-ups (for example auto-reflection, marking and parent-teacher interactions) work to support and complement the pillars identified in the student learning process.

Many of these tasks are ripe for automation. As Sweigart writes, "many people spend hours clicking and typing to perform repetitive tasks, unaware that the machine they're using could do their job in seconds if they gave it the right instructions" [9]. For my thesis project of the GymInf formation, I intend to build a suite of tools to support a range of teacher tasks including

- organizing seating plans
  to aid the atmosphere and relationships

- suggesting teacher-student interactions for upcoming classes
  to foster relationships

- building individual student reports
  to regulate cadence and scaffolding, as well as for follow-ups

- creating spreadsheets of marks
  for follow-ups

Teachers will interact with the tool using a GUI: it should be easy to use for all teachers, not just those teachers who are computer literate.

I will implement this project using *Test Driven Development*. This is a style of software development that grew out of the 'Extreme Programming' philosophy of the 1990s, encouraging quick development cycles and active feedback from clients. The first book written on the subject, *TTD by Example*, by Kent Beck, is still the main resource [2]. Kent emphasises the need to thoroughly test the requirements in order to have confidence when refactoring or making changes, as well as the need to make incremental changes (both for confidence and for time to market). TTD programming follows the following cycle:

Red. Write a failing test for a requirement.

Green. As quickly as possible, write code to pass the test.

Refactor. Clean the code, removing duplication and renaming functions.

Coding is driven by testing. In the thesis I will explore testing in more detail.

2 Introduction

Bucheton and Soulé have described the act of teaching as a multiagenda game of postures requiring good preparation and excellent micro-decisions [5]. In their model, they identify the crucial roles of the teacher in controling the cadence, the atmosphere, the scaffolding and the relationships during the class, and how each supports the learning objective. And yet many of the teacher's most time'consuming tasks do not take place in the classroom: good preparation and strong follow-ups (auto-reflection, marking, parent-teacher interactions) work to support and complement the overall success of the student learning process.

For example, teacher responsibilities include the ability

- To plan and implement effective classroom management practices

- To design and implement effective strategies to develop independent learners

- To engage students in active, hands-on, creative problem-based learning

- To build students' ability to work collaboratively with others

- To maintain a safe, orderly environment conducive to learning

- To adapt instruction/support to students' differences in development, learning styles, strengths and needs

- To write student reports to guide changes in instruction and practice, and to improve student learning

Many of these tasks are ripe for automation [9]. I would argue that some of these tasks should **not** be automated. As John Hattie explains in *Visible Learning* [6], *"Expert teachers monitor learning and provide feedback."* In my opinion writing student reports are a perfect example of a necessary evil: although time consuming (and potentially stressful) for the teacher, writing a student report forces the teacher to reflect on the progress of the student and at the same time manage the expectations of all partners - student, teacher, management and parent.

For the Master's thesis project undertaken for the GymInf formation, I intend to build a suite of tools to support a range of teacher tasks including organising seating plans, building individual student reports, suggesting teacher-student interactions for upcoming classes, and creating spreadsheets of marks.

3 System plan

3.1 InOut

We can analyse a system by connecting its (physical or virtual) inputs and outputs. In this project, we have

Inputs	**Outputs**
• Class lists	• Seating plans
• Commentary on students (during class)	• Suggested students for focus (during planning)
• Exam notes	• Average notes for the semester
	• Individual student reports

4   Choice of programming language

5   Testing

6   Test driven development

7   Useful quotes

"The best climate for learning is one in which there is trust. Students often don't like to make mistakes because they fear a negative response from peers. Expert teachers create classrooms in which errors are welcome and learning is cool." [6]

"Since it has been reasonably well established that student affect toward a class is related to student learning, student attitudes toward classroom arrangements are a matter of no small concern when determining a choice of classroom arrangement." [8]

`https://github.com/zedr/clean-code-python#table-of-contents`

References

[1] Anaya, M. (2018). Clean code in Python. Packt Publishing

[2] Beck, K. (2003). Test driven development by example. Addison-Wesley.

[3] Bloom, B. S. (1976). Human characteristics and school learning. McGraw-Hill.

[4] Bryant, J., Heitz, C., Sanghvi, S., & Wagle, D. (2020). How artificial intelligence will impact K-12 teachers. Retrieved May, 12, 2020.

[5] Bucheton D. & Soulé Y. (2009). Les gestes professionnels et le jeu des postures de l'enseignant dans la classe : un multi-agenda de préoccupations enchâssées.

[6] Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning. Routledge.

[7] Martin, R. (2008). Clean Code. Pearson.

[8] McCorskey, J. C., & McVetta, R. W. (1978). Classroom seating arrangements: Instructional communication theory versus student preferences. Communication education, 27(2), 99-111.

[9] Sweigart, A. (2015). Automate the Boring Stuff with Python.