Masters Programmes

Individual Work Assignment Cover Sheet

Submitted by: 2084749

Date: 10th December 2021

Module Title: Programming Solutions for Enterprises

Module Code: IB96D0

Date/Year of Module: 2021-2022

Submission Deadline: Tuesday 10th December 2021, 20:00:00

Word Count: 984 words

Title: Reflection of making the booking system

I declare that this work is being submitted, in accordance with the University's Regulation 11 and the WBS guidelines on plagiarism and collusion. All external references and sources are clearly acknowledged and identified within the contents. No substantial part(s) of the work submitted here has also been submitted in other assessments for accredited courses of study and if this has been done it may result in us being reported for self-plagiarism and an appropriate reduction in marks may be made when marking this piece of work

Contents

1. Introduction

This report provides key points about my personal reflections and what I have learnt from the Programming Solutions for Enterprise module of the Management of Information Systems and Digital Innovation course at Warwick Business School.

In the first section, I outline my experience of learning Python, including my motivations, approach, and successes and failures. This is because adult learners do not exist solely in the classroom; instead, they are individuals with existing knowledge and experiences who participate in courses for reasons including career enhancement, career change and personal growth (Taylor and Hamdy, 2013). For this reason, I do not believe that it is possible to discuss my own learning without discussing my journey to the point at which the course began.

According to Alrashidi, Joy and Ullmann (2019), reflective writing embeds novel experiences into learners' tacit knowledge with the perspective of awareness and an emphasis on future actions. which suitable to problem-solving and algorithms nature of programming works. Therefore, this essay's second and third sections will assess learning objectives with a critical reflection of the relativity of theoretical knowledge.

2. Previous experiences/Learning objectives

Python is a programming language used for automation in many open-source repositories, leading to a robust network effect. It benefits from high readability, but this limits programs' speeds as it optimises human comprehension (Sweigart, 2015). As the broader IT and computing industry focused on automation to leverage productivity and automate repetitive tasks with the rise of Industry 4.0 and blockchain programming, I knew I would gain a marketable skill by developing my expertise in programming. Automation allows more efficient usage of human resources, as employees can maximise the time spent on more creative or complex problem-solving that currently lies beyond the capability of artificial intelligence (Raisch and Krakowski, 2021).

Due to this sea change in how the industry approaches computing, I began learning Python programming in January 2020 to leverage my productivity, create command-line automation tools and create a personal-use system to collect a dataset that quantifies various aspects of my daily life, such as my diet, exercise, wardrobe and time management. Such a customisable personal organisation tool would deliver utility and have potential beyond my personal usage as a saleable product.

According to Newell (2015), to become an expert at a specific skill, a learner should utilise explicit knowledge based on theory and peer learning in combination with practice. Therefore, I first used knowledge from external repositories such as MOOCs and textbooks to internalise the foundations of Python and basic automation. Next, I co-

created a command-line automation program and web-scraping tool with a mentor to take advantage of the transfer of tacit knowledge (Nonaka and Takeuchi, 2008).

3. Learning outcomes

This section will assess learning outcomes and techniques used in making a room-booking system for enterprises. Agile methodology, modularity and open-source repositories were used to improve learning processes and programming logic.

3.1 Modularity

A modularity strategy is used to deconstruct complicated components into compact elements to solve complex problems. Still, developers must work on modules with a single design rule for relevant features, compatibility and functions (Baldwin, Clark and Clark, 2000).

In the intricate enterprise booking system that I worked on, functions are divided into filtering, booking, and the main loop text. Loop text is the central method where all features lead back to the central console with a quit function. The filtering system is used to filter rooms based on availability, location and types of equipment. The booking function provides a user with available times between 9 am and 9 pm.

A modular approach to coding facilitated working within design rules. Working on modules that grew in difficulty gave me opportunities to learn specific subjects, such as code-reusability and the time delta module, which was challenging. In addition,

modularity recycles effort and cognitive loads and provides a higher level of creativity while working, increasing productivity and reducing cognitive fatigue.

3.2 Agile methodology and its importance

Before starting the course, I had not expected to learn agile software development and did not understand its use, hence its omission from my intended education. However, I now appreciate the iterative and reflective approach to development.

Agile methodology leverages constant improvement, feedback loops and emergent strategies by providing daily meetings, iterations, maintaining an overview of projects, making customer-centric solutions, and continuous feedback and planning. Kanban allows constant note-taking within each iteration (Cockburn and Highsmith, 2001).

I found that working iteratively provides a strategic advantage compared to the waterfall method. Emergent planning provides flexibility in bug fixing and updating features if problems arise. Constant improvement is archived with continuous planning and feedback. Therefore, I obtained new knowledge with each iteration, constantly adapted, and completed my program on time.

3.3 Time-bound project-based learning

According to Newell (2015), an epistemology of practice incorporates the internalisation of knowledge through experiential learning. To be an expert at something, one needs to practice, do project-based learning and understand the inner workings of an expertise to utilise it effectively (Kokotsaki, Menzies and Wiggins, 2016).

My attempts to learn the data-analysis tool pandas, CSV manipulation and text manipulation were ineffective due to the absence of challenges and urgency. I learned pandas, Tkinter and CSV database manipulation to make the booking system due to urgency, challenges and demand for the program. This proves that project-based learning is an excellent way to internalise knowledge, which accelerates the learning process.

4. Conclusion

By developing a booking system, I have used novel learning and developmental strategies such as agile methodologies, modularity and project-based learning as tools to further my education. I have improved in Python automation as I gained experience creating systems for customers. In addition, Python, pandas and CSV manipulation proved to be an excellent introduction to data science.

In the future, I would like to achieve more learning goals using agile methodology, modularity and project-based learning by freelancing, doing automation projects and using data analytics to achieve my long terms goals.

5. References

Alrashidi, H., Joy, M. and Ullmann, T. D. (2019) 'A reflective writing framework for computing education', *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, p. 295. doi: 10.1145/3304221.3325580.

Baldwin, C. Y., Clark, K. B. and Clark, K. B. (2000) *Design rules: The power of modularity*. MIT press.

Cockburn, A. and Highsmith, J. (2001) 'Agile software development, the people factor', *Computer*, 34(11), pp. 131–133.

Kokotsaki, D., Menzies, V. and Wiggins, A. (2016) 'Project-based learning: A review of the literature', *Improving schools*, 19(3), pp. 267–277.

Newell, S. (2015) 'Managing knowledge and managing knowledge work: what we know and what the future holds', *Journal of Information Technology*, 30(1), pp. 1–17.

Nonaka, I. and Takeuchi, H. (2008) 'The knowledge creating company. 1995', *Sitio en Internet: http://www. gestiondelconocimiento. …*, (August). Available at: https://scholar.google.co.za/scholar?q=Nonaka+%26+Takeuchi+1995+&btnG=&hl=en&as_sdt=0%2C5#9.

Raisch, S. and Krakowski, S. (2021) 'Artificial intelligence and management: The automation–augmentation paradox', *Academy of Management Review*, 46(1), pp. 192–210. doi: 10.5465/AMR.2018.0072.

Sweigart, A. (2015) *Automate the boring stuff with Python : practical programming for total beginners* . San Francisco: No Starch Press.

Taylor, D. C. M. and Hamdy, H. (2013) 'Adult learning theories: implications for learning and teaching in medical education: AMEE Guide No. 83', *Medical teacher*, 35(11), pp. e1561–e1572.