ALCU Region III Research Journal
ISSN 2672-2143

QuantSci: The Faculty Scheduling Functional Program

Lovell M. Abello, CPA, DBA, IP, LPT, Xero (Advisor, Payroll and Migration) Certified, FRIRes, RIIVPF, Amor I. Barba, MM, Lennard J. De Luna, MBA, Jula V. Ong, MBA, Camille M. Garillo, MSHRM
City College of Angeles, Angeles City, Pampanga, Philippines

A local college in Angeles City, Philippines uses manual system for faculty scheduling. The said scheduling is usually conducted thrice every academic year – 1st and 2nd semesters plus the summer term. It is also worth mentioning that there are a good number of software packages as a system for scheduling, which may be purchased. However, such investment decision may entail a relatively huge amount of capital expenditure. The proponents of this applied research feel the urgent need for the local college to develop a functional program that could facilitate an efficient, cost-effective but reliable scheme of faculty scheduling. Basically, the functional program works by way of an embedded algorithm that performs system of processes - taking into account diverse restrictions/preferences set by system-users. Moreover, the Agile Extreme Programming (XP) software development methodology was utilized. Various stakeholders who were the respondents of the study similarly evaluated the application using an evaluation tool that is based on International Organization for Standardization - the ISO/IEC 25010. The functional program "QuantSci" satisfied the respondents' understanding about the attributes of a quality system. The application is not complicated to use, is also cost-effective, and does not compromise the reliability of results. Because of the development of QuantSci, the objectives of an efficient and reliable scheduling system were satisfied. The ultimate implementation and maintenance of the program is a good start in the promotion of e-governance of the local college.

Keywords: QuantSci, agile extreme programming (XP), faculty scheduling, functional program, e-governance.

Introduction

City College of Angeles (a local college in Angeles City, Pampanga, Philippines) uses manual system for faculty scheduling. Scheduling is usually conducted thrice every academic year – 1st and 2ndsemesters and summer term. It is also worth mentioning that there are a good number of software packages as a system for scheduling, which may be purchased. However, such investment decision may entail a relatively huge amount of capital expenditure notwithstanding the fact that it is not a simple task to request financial subsidy from the local government.

Shaikh and Kasat (2009) suggest the application of e-governance in education. The researchers trust that in the framework of electronic governance such will result in improved transparency, swift information distribution, advanced administrative efficiency and superior public services in sectors including transportation, education, power, health, water, security and the state administration and municipal services.

Also, Heeks (2001) asserted that new information and communication technologies make momentous impact in the achievement of good governance goals. E-governance can transform governance to be more efficient and more effective, and bring other benefits too.

Based on the foregoing, the proponents of this applied research feel the urgent need for the local college to develop a functional program that could facilitate an efficient, cost-effective but reliable

scheme of faculty scheduling. This functional program ensures accuracy, that is, reducing (if not fully eradicating) errors in faculty scheduling. This is made possible because of an embedded algorithm that basically performs system of processes which are (a) selecting and finding vacant time and vacant room, (b) considering an available faculty and (c) taking into account diverse constraints and preferences set by system-users. Software and hardware requirements, research instruments for the study and pertinent human resource department information to complete the inquiries were also incorporated.

Methodology

The study utilized the Research and Development Model and the Agile Extreme Programming (XP) software development methodology during the system development. In today's software industry, technological aptitude and ever-evolving client requirements have steered to more complex software demands. Agile-based software development is progressively being embraced by the software practitioners as it reassures early software development and high value software products. Also, it offers receptiveness to changes in user requirements, providing for their swift absorption during software development (Matharu, Mishra, Singh, and Upadhyay, 2015).

Accordingly, Beck (1999) describes a set of five values that form a foundation for all work performed as part of XP, and these are communication, simplicity, feedback, courage, and respect. The Extreme Programming uses an object-oriented approach as its preferred development paradigm and encompasses a set of rules and practices that occur within the context of four framework activities: planning, designing, coding, and testing (Pressman, 2010). Figure 1 demonstrates the XP process and notes some of the fundamental ideas and tasks that are linked with each framework activity.

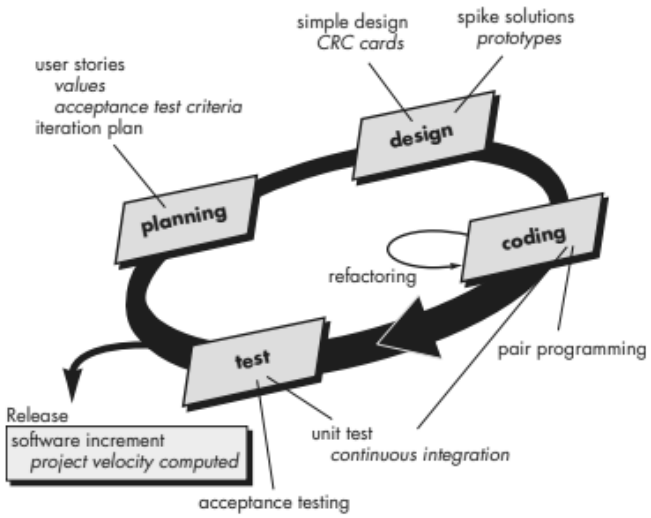

Figure 1. The Agile Extreme Programming Process (Pressman, Roger. S., 2010. Software Engineering, A Practitioner's Approach. McGrawHill, USA.)

One of the key XP activities is planning. The planning activity begins with listening; it is a requirement-collecting activity that permits the technical members of the XP group to comprehend the practical aspect of the program and to get a comprehensive impression for the required output and major features and functionality. Another is XP designing. This activity meticulously follows the "keep it

simple" principle. A simple design is constantly preferred over a more intricate representation. In addition, the design provides implementation guidance for a story as it is being written (nothing less and nothing more). The succeeding XP activity is coding wherein after stories are developed and the primary design work is done, the team does not move to code, but fairly develops a series of unit tests that will exercise each of the story that is to be involved in the current release. After the unit test has been produced, the developer is better able to focus on what must be implemented to pass the test. Once the code has been completed, it can be unit-tested straightaway, thereby providing immediate response to the developers. Finally, the in testing activity, it is said that the creation of unit tests before coding begins is a crucial element of the XP approach. Whenever the individual unit tests are systematized, integration and validation testing of the system can happen on a regular interval. This affords the XP team with an incessant indication of development and also can raise warning flags promptly, if things may go twisted (Pressman, 2010). It is worth mentioning that the aforesaid XP planning activities and pertinent principles guided the researchers in the completion of this project.

Requirements Specifications Analysis

This part of the paper expounds the attributes of the functional program and recapitulates the apparent features from gaining access into the system until the display of end-result through the use of the application.

1. Access into the QuantSci. This application is basically a functional program (Excel-based). A password is required to gain access.
2. Encoding of data. The dean of each institute encodes the room assignments and subject codes. The functional program provides signal if the room has been occupied at the specific time encoded. As mentioned previously, basically the functional program works by way of an embedded algorithm that performs system of processes (selecting and finding vacant time and vacant room, considering an available faculty and taking into account diverse restrictions/preferences set by system-users).
3. The output. After the all data have been encoded, a report [summary of schedule] may be printed.

Logical Specifications

Figure 2. Data Flow Diagram for QuantSci

The data flow shows the main process of using the application. Once the supervisors/deans/heads encoded the data within the system using a laptop/desktop computer with MS Excel application, the record may be saved/stored inside the computer or possibly be forwarded (by means of email/drop box) to other authorities within the local college such as the human resource director, the Vice President for Academic Affairs or the College President. It is noteworthy that even in the absence of Internet connection, the program would work; Internet is only required (as reiterated) when the record shall be forwarded (electronically) to other college authorities.

Physical Specifications

On Software Requirements. The QuantSci program, as asserted beforehand, is a functional program meaning it is MS Excel-based. The basic software specification is the inclusion of the MS Excel application within a laptop/desktop.

Table 1. Recommended Specifications for Hardware Requirements

Device & Equipment	Specifications
Operating System	Windows 1.0
Processor	Intel(R) Core (TM) i5-6200U Processor @ 2.30GHz 2.40 Ghz.
Memory	4GB DDR3L
Storage	1TB SATA HDD

Test Procedures

Before the release of QuantSci, the functional program was tested with different test procedures to ensure that the application would work with its intended function. Accordingly, the application was tested through unit testing, integration testing, system testing and acceptance testing.

Evaluation Criteria and Evaluation Procedures

Various stakeholders – the respondents of the study – evaluated the application using an evaluation tool/questionnaire (Appendix 1) that is based on International Organization for Standardization, the ISO/IEC 25010 [Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE)], which is the foundation of a software product quality evaluation system. SQuaRE has eight product quality characteristics, namely: functionality, efficiency, compatibility, usability, reliability, security, maintainability and portability.

The said evaluation questionnaires were given to the evaluators. A 5-point Likert scale (Table 2) was used to rate each of the characteristics of the functional program. The evaluation criteria were explained very well to the evaluators. Consequently, the purposive sampling technique (also called judgment sampling) was applied. Purposive sampling is the thoughtful choice of an informant due to the qualities the informant possesses. It is a nonrandom method that does not need core theories or a set

number of informants. Simply put, the researchers resolve what needs to be known and set out to find people who can and are prepared to provide the information by virtue of understanding or experience (Bernard, 2002 and Lewis & Sheppard, 2006).

Regarding ethical considerations in research, all possible ethical issues were taken into account; plus, a consent form (Appendix 2) was adopted.

Table 2. Grand Mean of Scores and the Corresponding Adjective Ratings

Grand Mean	Adjective Rating
5	Outstanding
4	Very Satisfactory
3	Satisfactory
2	Fair
1	Poor

Results and Discussion

Subsequent to the development of QuantSci, the following screenshots from the functional program are evident based on identified features in requirements specification analysis.

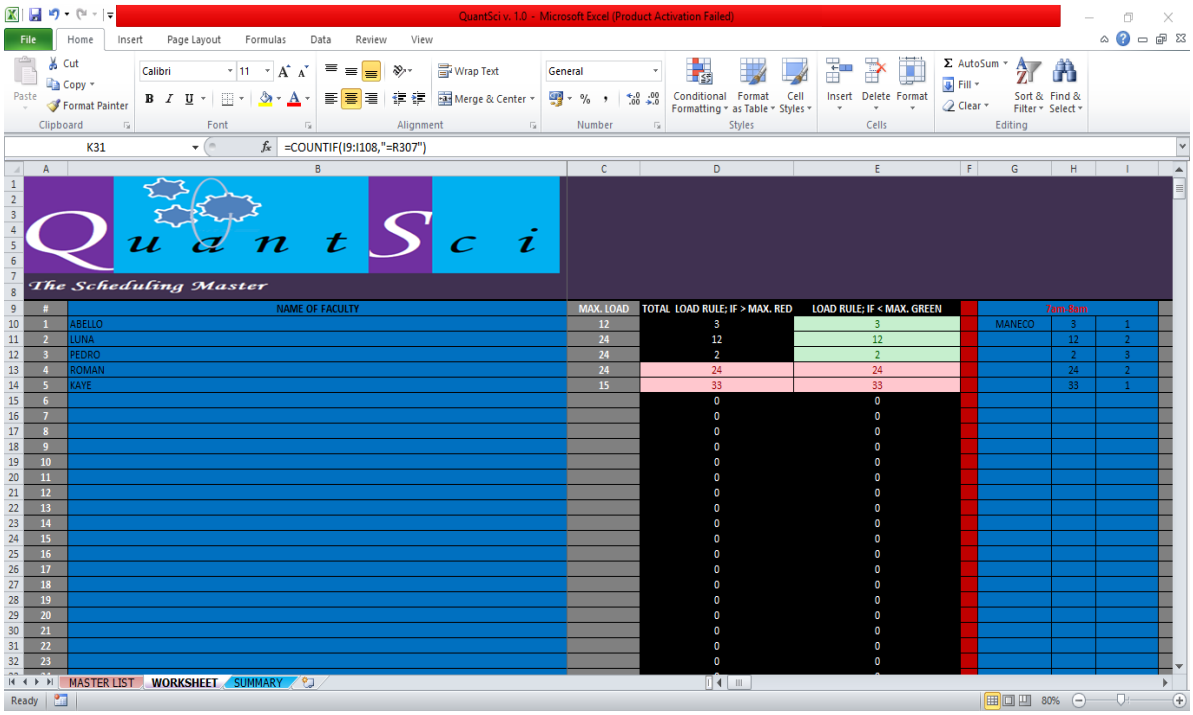

Figure 3. User Interface of QuantSci

The user interface of QuantSci basically consists of three (3) modules: [1] the master list module (consisting of names of all faculty members, the total maximum teaching load, and the current tally of given load), [2] the worksheet module (this is the most crucial part among all modules; this is the part where the users do the encoding of data), and [3] the summary module (this part is the part that may be

printed for verification/review). Figure 4 shows that the functional program prompts a password before a user could gain access into the application.

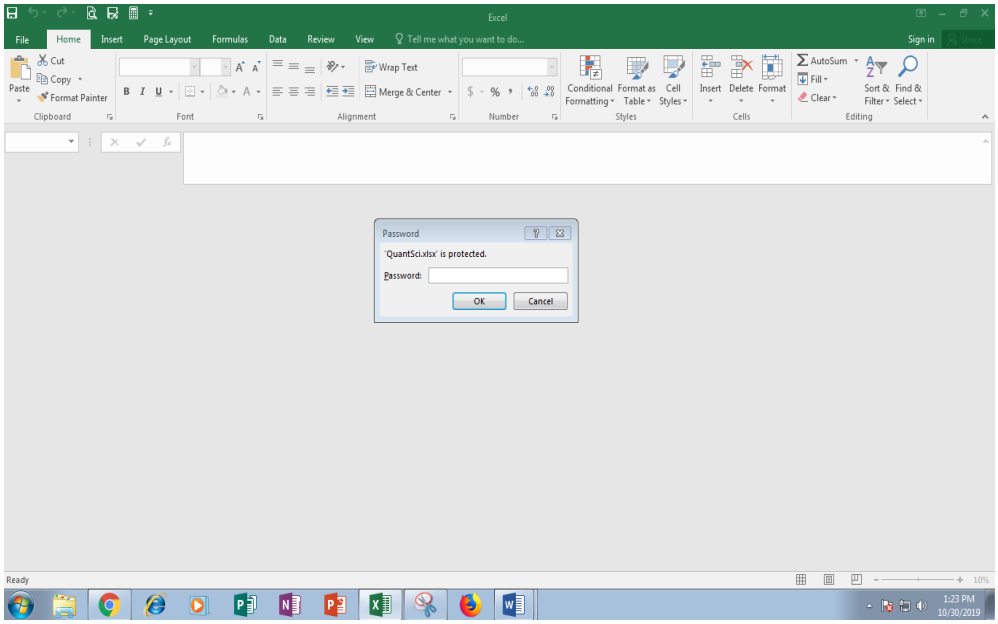

Figure 4. QuantSci Prompts a Password

Moreover, the functional program was evaluated by a total of twenty-four (24) respondents, who included academic program coordinators (17), IT experts (3), institute deans (3), and the Vice President for Academic Affairs (1). The system was evaluated in terms of functionality, efficiency, compatibility, usability, reliability, security, maintainability and portability.

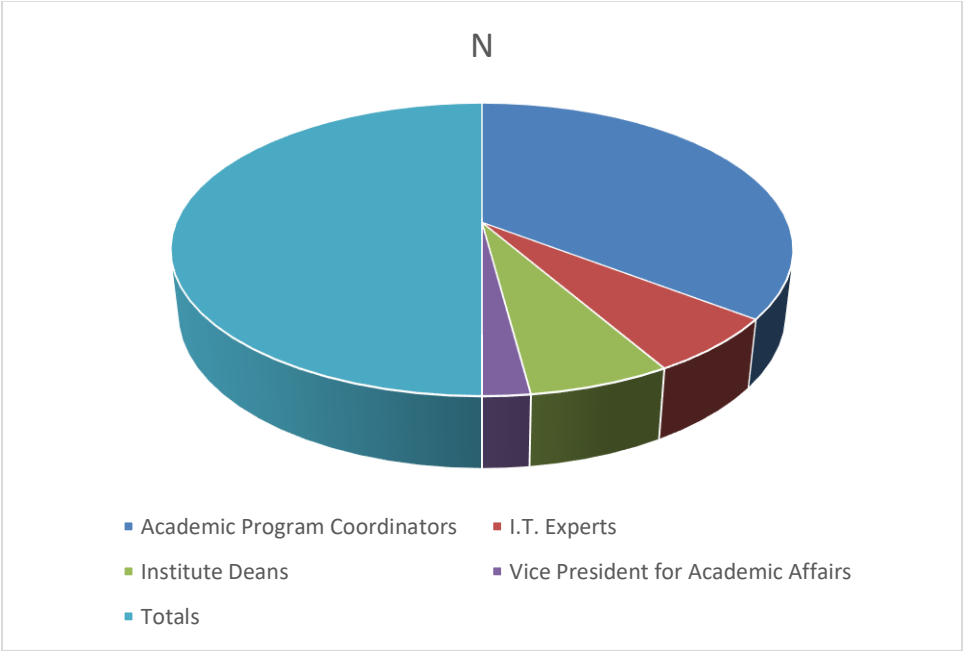

Figure 5. Respondents of the Study

Evaluation questionnaires (as a reiteration) were given to the evaluators. A 5-point Likert scale was used to rate each of the characteristics of the functional program; the evaluation criteria were explained very well to the evaluators. The summary of the outcome is presented in Tables 3 to 6.

Table 3. Assessment of Academic Program Coordinators

Criteria	Mean	Descriptive Rating
Functionality	4.12	Very Satisfactory
Efficiency	4.00	Very Satisfactory
Compatibility	4.06	Very Satisfactory
Usability	4.59	Very Satisfactory
Reliability	4.18	Very Satisfactory
Security	3.76	Satisfactory
Maintainability	4.12	Very Satisfactory
Portability	4.18	Very Satisfactory
Total	4.13	Very Satisfactory

Table 4. Assessment of I.T. Experts

Criteria	Mean	Descriptive Rating
Functionality	4.00	Very Satisfactory
Efficiency	4.00	Very Satisfactory
Compatibility	4.00	Very Satisfactory
Usability	4.00	Very Satisfactory
Reliability	3.67	Satisfactory
Security	4.00	Very Satisfactory
Maintainability	4.33	Very Satisfactory
Portability	4.33	Very Satisfactory
Total	4.04	Very Satisfactory

Table 5. Assessment of Institute Deans

Criteria	Mean	Descriptive Rating
Functionality	3.33	Satisfactory
Efficiency	3.67	Satisfactory
Compatibility	4.67	Very Satisfactory
Usability	4.00	Very Satisfactory
Reliability	3.67	Satisfactory
Security	4.00	Very Satisfactory
Maintainability	5.00	Outstanding
Portability	4.33	Very Satisfactory
Total	4.08	Very Satisfactory

Table 6. Assessment the Vice President for Academic Affairs

Criteria	Mean	Descriptive Rating
Functionality	4.00	Very Satisfactory
Efficiency	4.00	Very Satisfactory
Compatibility	4.00	Very Satisfactory
Usability	4.00	Very Satisfactory
Reliability	3.00	Satisfactory
Security	3.00	Satisfactory
Maintainability	5.00	Outstanding
Portability	5.00	Outstanding
Total	4.00	Very Satisfactory

In general, QuantSci received a 'very satisfactory' rating from the evaluators (Tables 3 to 6). However, the respondents provided a hint, which is the improvement of the user interface (aesthetics), in terms of color, font, font size and the like.

Conclusions and Recommendations

The functional program "QuantSci" satisfied the respondents' understanding about the attributes of a quality system. The application is not complicated to use, is also cost-effective, and does not compromise the reliability of results. The objectives of an efficient and reliable scheduling system were satisfied. The development, the ultimate implementation and the continuance of this application is a good start in the promotion of e-governance of the local college.

References

1. *Beck, K. (1999). Extreme Programming Explained: Embrace Change. Addison-Wesley.*
2. *Bernard, H.R. (2002). Research Methods in Anthropology: Qualitative and quantitative methods. 3rd edition. Altamira Press, Walnut Creek, California.*
3. *Heeks, R. (2001). Understanding e-Governance for Development. i-Government Working Paper Series, p. 3.*
4. *Information technology — Software product quality. ISO/IEC 2000. Retrieved from https://www.cse.unsw.edu.au/~cs3710/PMmaterials/Resources/9126-1%20Standard.pdf*
5. *Lewis, J.L. & S.R.J. Sheppard (2006). Culture and communication: can landscape visualization improve forest management consultation with indigenous communities? Landscape and Urban Planning 77:291–313.*
6. *Matharu, G. S., et al. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. SIGSOFT Softw. Eng. Notes 40, 1, 1-6. Retrieved from https://doi.org/10.1145/2693208.2693233.*
7. *Pressman, R. S. (2010). Software Engineering, A Practitioner's Approach. McGrawHill, USA.*
8. *Shaikh, N., et al. "E-Governance in Education as an Effective Management Control Tool in Developing Human Resource-Faculty Appraisal System,"* 2009 Second International Conference on Emerging Trends in Engineering & Technology, *Nagpur, 2009, pp. 410-415.*
9. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Retrieved from: https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en.*