

*Robert P. Batzinger,
Phatnaree Srisomphan,
Nutawud Boonroadwongs,
Surachet Wongchomphu,
Payap University Faculty of Business Administration and Science,
Department of Computer Science,
Amphur Muang, Chiang Mai 50000 Thailand*

**DEVELOPING A GITLAB-BASED ISO29110 FRAMEWORK FOR MENTORING
AND MANAGING UNDERGRADUATE CAPSTONE PROJECTS**

*Роберт П. Бетзингер,
Фатнарий Срисомфан,
Нутаевуд Бунродвонс,
Сурачет Вонгчомпхуа,
Паяп университетинин бизнес башкаруу жана илим факультети,
Информатика факультети,
Амфур Муанг, Чиангмай 50000 Таиланд*

**НАСААТЧЫЛЫК ҮЧҮН GITLAB НЕГИЗДЕГЕН ISO29110 АЛКАГЫН ИШТЕП
ЧЫГУУ ЖАНА ОКУУЧУЛАРДЫ БАШКАРУУ CAPSTONE ДОЛБООРЛОР**

*Роберт П. Бетзингер,
Фатнарий Срисомфан,
Нутаевуд Бунродвонс,
Сурачет Вонгчомпхуа,
Факультет делового администрирования и науки университета Паяп,
факультет компьютерных наук,
Амфур Муанг, Чиангмай 50000 Таиланд*

**РАЗРАБОТКА РАМКИ ISO29110 НА ОСНОВЕ GITLAB ДЛЯ НАСТАВНИЧЕСТВА
И УПРАВЛЕНИЯ СТУДЕНТАМИ CAPSTONE ПРОЕКТА**

***Annotation:** This paper describes an attempt to improve the management and the success of student capstone projects through the development and use of an online project management framework compliant with ISO29110. Within this framework, students develop their ideas which are captured incrementally as updates to a GitLab repository. Thus, the process of developing student projects was broken down into a series of small tasks which gathers information as it is needed and recorded the simultaneous development of project code and documentation.*

Student capstone courses were redesigned as an integrated workflow to incrementally document, design, code and test a practical solution to a problem chosen by the student. The process

starts with an initial concept paper that identifies the objectives of the project. These concepts give rise to a proposal that identifies the intended goals as well as the context of the problem and specific requirements of the solution. The project design document establishes the development and testing plan from which the statement of work can be written and serves as the basis for the schedule of work and test plan which drives the pace of development, testing and documentation of the prototype. A comparison of the resulting prototype to the design specifications and user feedback provide a final analysis of the project.

The documentation serves as a portfolio of the entire capstone project. Throughout the process, the student's faculty adviser has access to

the latest version of all files, providing timely indication of progress, effort, and direction of a project. Thus, the framework serves a 3-fold purpose in the training of students: facilitating discussion between mentors and students, managing the project with deadlines, and developing professional communication skills. The framework has had a positive effect on students enrolled in capstone courses.

Аннотация: Бул макалада ISO29110 стандартына шайкеш келген онлайн долбоорду башкаруу базасын иштеп чыгуу жана колдонуу менен студенттердин долбоорлорунун бурчуна коюлуучу менеджментти жана ийгиликти жакшыртуу аракетин баяндалган. Ушул алкакта студенттер өз идеяларын өркүндөтүшөт, алар акырындык менен GitLab репозиторийине жаңыртууларды киргизишет. Ошентип, студенттердин долбоорун иштеп чыгуу процесси бир нече чакан тапшырмаларга бөлүнүп, зарылчылыкка жараша маалыматтарды чогултуп, бир эле мезгилде долбоордун кодун жана документтерин иштеп чыгууну камтыйт.

Студент тандаган көйгөйдү практикалык жол менен документтештирүү, иштеп чыгуу, кодкоо жана текшерүү үчүн, студенттердин курстары интегралдык иш агымына айландырылды. Процесс долбоордун максаттарын аныктаган алгачкы концепциядан башталат. Бул түшүнүктөр болжолдонгон максаттарды, ошондой эле көйгөйдүн контекстин жана чечүү үчүн конкреттүү талаптарды аныктаган сунушту жаратат. Долбоордук документ техникалык тапшырманы түзө турган иштеп чыгуунун жана сыноонун планын белгилейт жана прототиптин иштелип чыгуу, тестирилөө жана документтештирүү темпин аныктоочу иш графиги жана сыноо планы үчүн негиз болуп саналат. Алынган прототипти дизайн өзгөчөлүктөрү жана колдонуучулардын пикирлери менен салыштыруу дизайнга акыркы талдоону камсыз кылат.

Документтер бүт Capstone долбоору үчүн портфолио катары кызмат кылат. Студенттин окутуучусу процесстин жүрүшүндө долбоордун жүрүшү, аракеттери жана багыты жөнүндө өз убагында маалымат берип, бардык файлдардын

акыркы версиясын колдоно алат. Ошентип, түзүм студенттерди окутуунун үч максатын көздөйт: устаттар менен студенттердин ортосундагы талкууну жеңилдетүү, долбоорду өз убагында башкаруу жана кесиптик баарлашуу көндүмдөрүн өнүктүрүү. Бул түзүм Capstone курстарына жазылган студенттерге оң таасирин тийгизди.

Аннотация: В данной статье описывается попытка улучшить управление и добиться успеха студенческих проектов, основанных на краеугольных камнях, посредством разработки и использования структуры управления онлайн-проектами, соответствующей стандарту ISO29110. В рамках этой структуры студенты развивают свои идеи, которые постепенно фиксируются в виде обновлений в репозитории GitLab. Таким образом, процесс разработки студенческих проектов был разбит на серию небольших задач, которые собирают информацию по мере необходимости и фиксируют одновременную разработку кода проекта и документации.

Курсы для студентов были переработаны в интегрированный рабочий процесс для постепенного документирования, проектирования, кодирования и тестирования практического решения проблемы, выбранные студентами. Процесс начинается с первоначального концептуального документа, в котором определяются цели проекта. Эти концепции порождают предложение, которое определяет намеченные цели, а также контекст проблемы и конкретные требования к решению. В проектно-конструкторском документе устанавливается план разработки и тестирования, на основе которого может быть составлено техническое задание, и он служит основой для графика работ и плана тестирования, который определяет темпы разработки, тестирования и документирования прототипа. Сравнение полученного прототипа с проектными спецификациями и отзывы пользователей обеспечивают окончательный анализ проекта.

Документация служит портфолио для всего проекта Capstone. На протяжении всего процесса консультант факультета имеет доступ к последней версии всех файлов, обеспечивая своевременную информацию о про-

грессе, усилиях и направлении проекта. Таким образом, структура служит трем целям обучения студентов: содействие дискуссии между наставниками и студентами, управление проектом с соблюдением сроков и развитие профессиональных коммуникативных навыков. Эта структура оказала положительное влияние на студентов, обучающихся на курсах Capstone.

Key words: *mentoring Student Capstone Projects, GitLab File Repository, Design and Implementation Framework, ISO29110 Project Management and Implementation Process.*

Түйүндүү сөздөр: *студенттик Capstone долбоорлоруна насаатчылык, GitLab файл сактагычы, дизайн жана ишке ашыруу негиздери, ISO29110 долбоорун башкаруу жана ишке ашыруу процесси.*

Ключевые слова: *наставничество над проектами Capstone студентов, репозиторий файлов GitLab, структура проектирования и внедрения, управление проектами ISO29110 и процесс их реализации.*

In 4 years of undergraduate study, Computer Science (CS) majors receive detailed instruction on a wide range of theories, skills, technologies, and practices. The content of classroom instruction and assigned readings are designed to cover the breadth of the approved curriculum [1]. Assignments focus on specific textbook examples to illustrate some aspects of the various problem domains. However, the informational load of this instruction within 14-week semesters often does not allow students time to integrate and master the concepts covered. Thus, classroom instruction does not always build student skill and confidence in applying computing principles to solve problems.

a more holistic opportunity to apply what they have learned to deliver a prototype for a solution to an issue in a problem domain of their choosing [2, 3]. The scope of a capstone project is broad enough to demonstrate the student's development and project management skills within the time-frame of 2 semesters. However, reports from various universities around the world has shown that capstone project courses can be a significant challenge to students [4, 5]. Inadequate project testing, overly ambiguous project scope and poor time management contribute to failure of student capstone projects. The Thai Software Industry Promotion Agency has shown that these factors are also hamper local Information Technology (IT) contractors who provide development and system services to Thai government [6–8]. In 2011, only 26% of the government contracts were completed on time and about 50% were over 6 months late [9].

The Thai Ministry of Industry has attempted to increase the competitiveness of the Thai IT industry by adopting and promoting ISO29110 as a national standard for project management of software and system development. This standard was chosen for its simplicity and ability to reduces process complexity into smaller manageable tasks [7, 9]. This reduces cost of implementation and time wasted on repetitive data collection [10]. Studies have shown that this improves the effectiveness of project management and the performance of implementation and deployment processes, especially in very small enterprises¹ (VSE) [6, 8].

Senior capstone courses give students with

¹Very small enterprises (VSE) are defined by Laporte [11] as service providers with fewer than 25 employees.

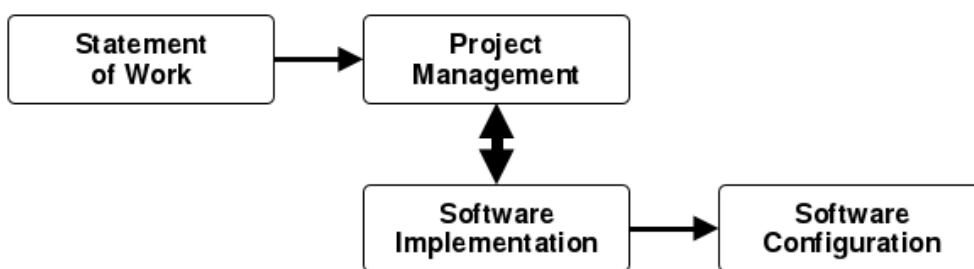


Fig. 1: ISO29110 Project Processes

The success of this standard comes in part from its focus on effective 2-way communication between customers and developers throughout all stages of the project,[11] as shown in Figure 1. This process facilitates the development of a practical description and plan that meets legal and contractual requirements without creating excessive bureaucratic overhead. The focus in the early phases is on defining the work to be done and the means to assess progress and project completion.

ISO29110 is well suited for the IT industry in Thailand where VSEs represent 95% of the Thai IT industry [7]. ISO29110 establishes documented streams of communication as shown in Figure 2. This process makes development a manageable activity and reduces the risk of project failure. Certification for ISO29110 requires VSE’s to demonstrate that their projects are regularly managed and developed as ISO29110-compliant projects. As a result, local IT firms are looking for new hires who are already familiar with this process.

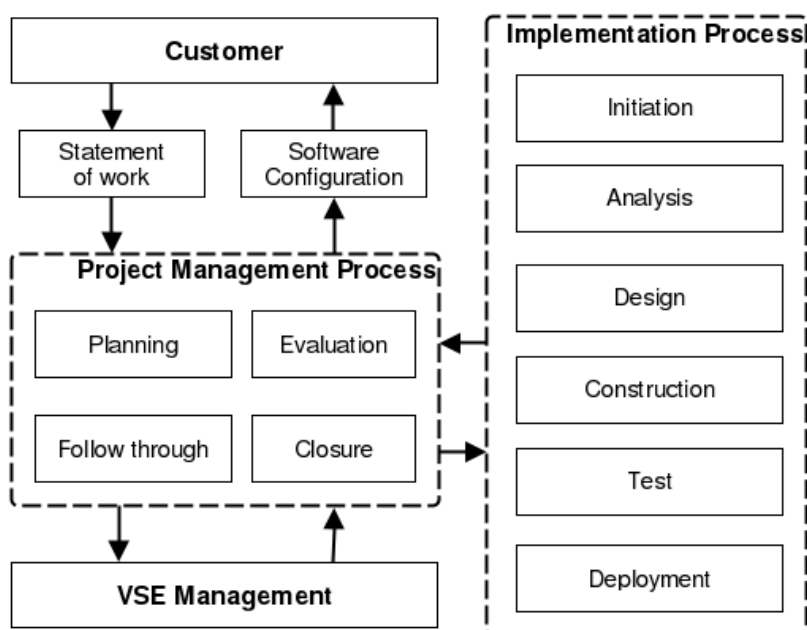


Fig. 2: ISO29110 Software Development Standard for VSE (Adapted from Laporte [11])

ISO29110 is also flexible enough to manage student capstone projects. Students take on the role of IT developers who proposed solution to a problem and negotiate product specifications that are mutually acceptable to a panel of faculty members of the CS department. Throughout the implementation process, students must regularly provide indication of progress and test results. Changes to the Statement of Work are allowed only by mutual consent and approved amendments in the project design and implementation documents. In the final review, the prototype is compared against the agreed product specification. Managing senior projects in a ISO29110-compliant manner provides familiarity with the standard and has the potential of improving job prospects within the local IT industry.

At Payap University (PYU), we have found that a 2-semester sequence provides CS students with sufficient time to design, develop, test and document a practical solution to a real-world problem. The process is broken into two subsequent courses:

- **Computer Project I:**² which focuses on product design and process planning and produces documents that define the project parameters as per the initial stages of a ISO29110-compliant project.

- **Computer Project II:**³ which focuses on the implementation of the approved design and results in the deployment of a tested and documented prototype. The implementation process is subject to weekly scrums and ISO29110-compliant periodic reviews against the proposed specifications. At the end of this course, stu-

dents must deploy their prototypes and demonstrate their understanding of techniques used.

Each student has a faculty member adviser to mentor them through the process and to promote the development of good professional skills that will also help them to successfully complete the project. Throughout both semesters, the student's adviser provides a series of forms and guides to help gather information and implement solutions in a way that builds documentation and a tested solution. Because the results of each phase form the basis for subsequent phases, the design description, documentation, source code, and test results constantly evolve as the project progresses.

The goal is to provide practical experience in ISO29110 project management where the creation of tested, documented solutions evolves during CS499 according to the specifications established in CS398. Workshops, scrums and consultations occur throughout the 2 semesters. Students have the summer break between the two semesters to study and review skills that they will need to complete the project. Effective discussion between student and mentor of both the intended product and the direction of the development on access to the current versions of all student generated files for the project.

ISO29110 related documents undergo several cycles of metamorphosis as the project description is defined and approved. Likewise, the source code and testing modules also change as the project progresses. Changes to the specifications must be carefully controlled by mutual agreement and result in corresponding updates and annotation in the design document.

² **Course description of CS398 at PYU:** Preparation of a project proposal in computer science for which students have ability and interest. Includes literature review, study and selection of software design and development processes; Study and analysis of user needs; Propose solutions to problems; Develop a documented design and development plan. Proposed projects will be presented and student's understanding of the project and its viability will be assessed. Proposals that pass the review process will be implemented in CS499.

³ **Course description of CS499 at PYU:** Study and development of the project established in CS398 resulting in a working prototype that meets design goals and passes the testing plan. The course includes preparation of reports, documentation and presentations.

Thus, source code version control software is essential to ISO29110 to ensure ready access to the latest versions of all documents while providing the means to identify differences between versions. This paper will describe the use of GitLab as a platform to the enhance interaction between the faculty adviser/mentor and the student/developer within the context of ISO29110-compliant project management.

In 2005, Linus Torvalds created Git to improve the handling of the source code of Linux being developed and maintained by a growing international community. The goal was to create a simple utility that could support large distributed projects asynchronously. Developers using Git could create and test code patches in a parallel working branch before merging the approved code in to the distributed branch [12]. Git was an instant success with large computer companies like Apple, Microsoft and Google as it could manage and merge branches of large code sets very quickly. The number of public and private Git repositories exploded with the launch of cloud based services on GitHub in 2008 and the open-source Git server software from GitLab in 2012. By 2016 Git had become the most popular version control system, curating over 80% of the world's open source projects. Today IT companies even use oncloud Git repositories to distribute programming challenges to potential applicants in order to assess their problem solving and programming skills.

As Git gained popularity in the software industry, universities quickly recognized the value of Git in the classroom for distributing the basic description and structures used in assignments, and for collecting the student's completed homework [13]. GitHub provides access to working examples of new programming languages, facilitates participation in open-source projects, distributes useful apps and creates a professional portfolio for job applications. CS and Software Engineering Departments are increasing aware of the need to integrate Git skill training into the core curriculum [14].

PYU CS has used cloud-based GitLab community services to manage Git repositories of student capstone projects. Each repository captures the documentation, source code, assets and test modules of a student capstone project. Five years ago, GitLab was chosen over GitHub because the default private project mode of GitLab. In addition, GitLab had an open-source server version that could be run on local servers. Subsequent capstone projects have benefited from the use of newer GitLab features, such as support for formatted documentation, a file system to capture and organize project files, tracking of milestones and issues, and the Application Programming Interface (API) which was used to automate the initiation, scheduling and summation of student projects [15]. Since then, both GitHub and GitLab platforms have evolved in similar ways but have different sets of features and services. Because GitLab has proven highly reliable and expandable, there has not been any compelling reason for us to migrate to GitHub at this time.

Description of the Framework

The framework for student projects arose from 3 core principles shared by members of the CS Department at PYU, namely:

1. ***Documentation is a means for capturing the key information that defines a project.*** Information requested at each stage of the project should give rise to working documents that form the basis for subsequent discussions, development and testing.

2. ***Information is gathered to develop deeper understanding of both the problem and its solution.*** Information captured in small manageable increments progressively contributes to both the content of the final documentation as well as the context for the development of assets, source code and unit tests.

The end product is a fully tested and documented solution to a problem. The final documentation must be consistent with the final product and should serve as a practical guide for installation, usage, and future development.

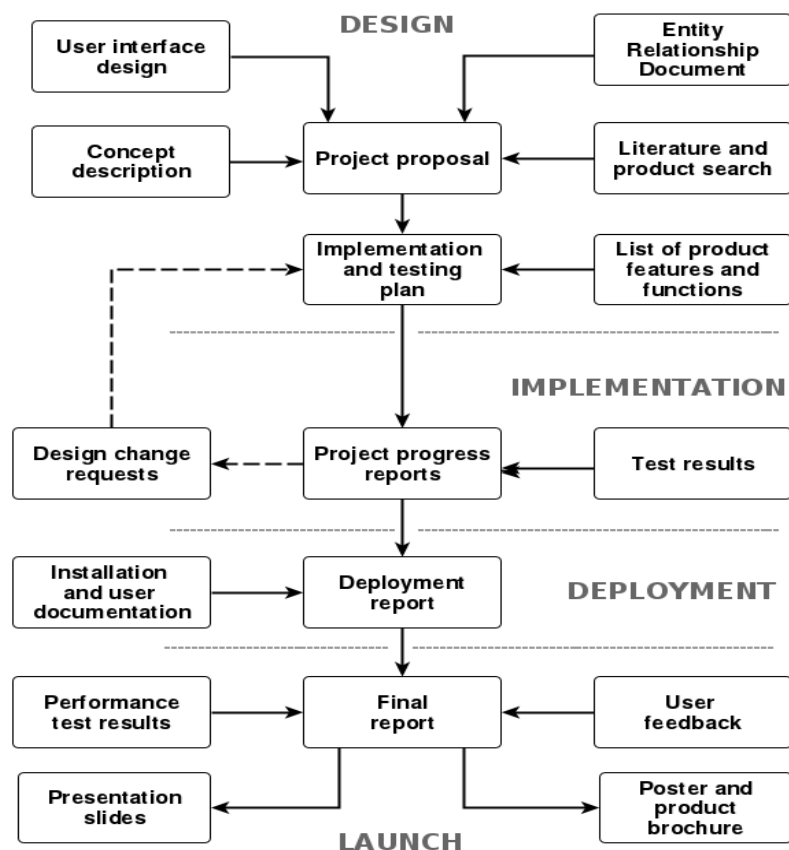


Fig. 3: Workflow of student capstone project documents

The 17 documents of the framework shown in Figure 3 aim to raise critical questions to stimulate the development of working definitions and scenarios needed to create and test practical solutions. Wherever possible, information was captured in hypertext markdown format to encourage reuse of the information as well as facilitate automated monitoring of progress. At each stage of development, the documentation was a critical part of the preparation for consultations and scrums between students and their mentors. The documents of this framework mimic the sequence and content of those used in an ISO29110 project.

In the design phase, students start by proposing a title and an abstract of the basic concepts and issues to be addressed. After discussion, revision, and approval of these basic concepts, students submit a study of their respective problem domain, including a description of existing attempts at a solution and the proposed functionality of their solution. Upon approval, students began to develop the

Project Proposal (which is equivalent to the ISO29110 Statement of Work). Once the scope and specification of the product of the capstone prototype has been approved, students identify, plan and schedule the various phases of implementation and testing. At this point, the Git repository is a complete description and specification of the project that is used as a development contract for the second term.

The activities of the second semester focus on executing the implementation and testing plan. Descriptions are updated as related issues become clear. The students are responsible for finding and implementing their solutions according to their own specs. Features that proved impossible to implement are documented and changes to the specs are proposed for approval. By the end of the semester, students deliver a working prototype of the solution verified by test results, user feedback, and final documentation showing that the final product conforms to the proposed solution.

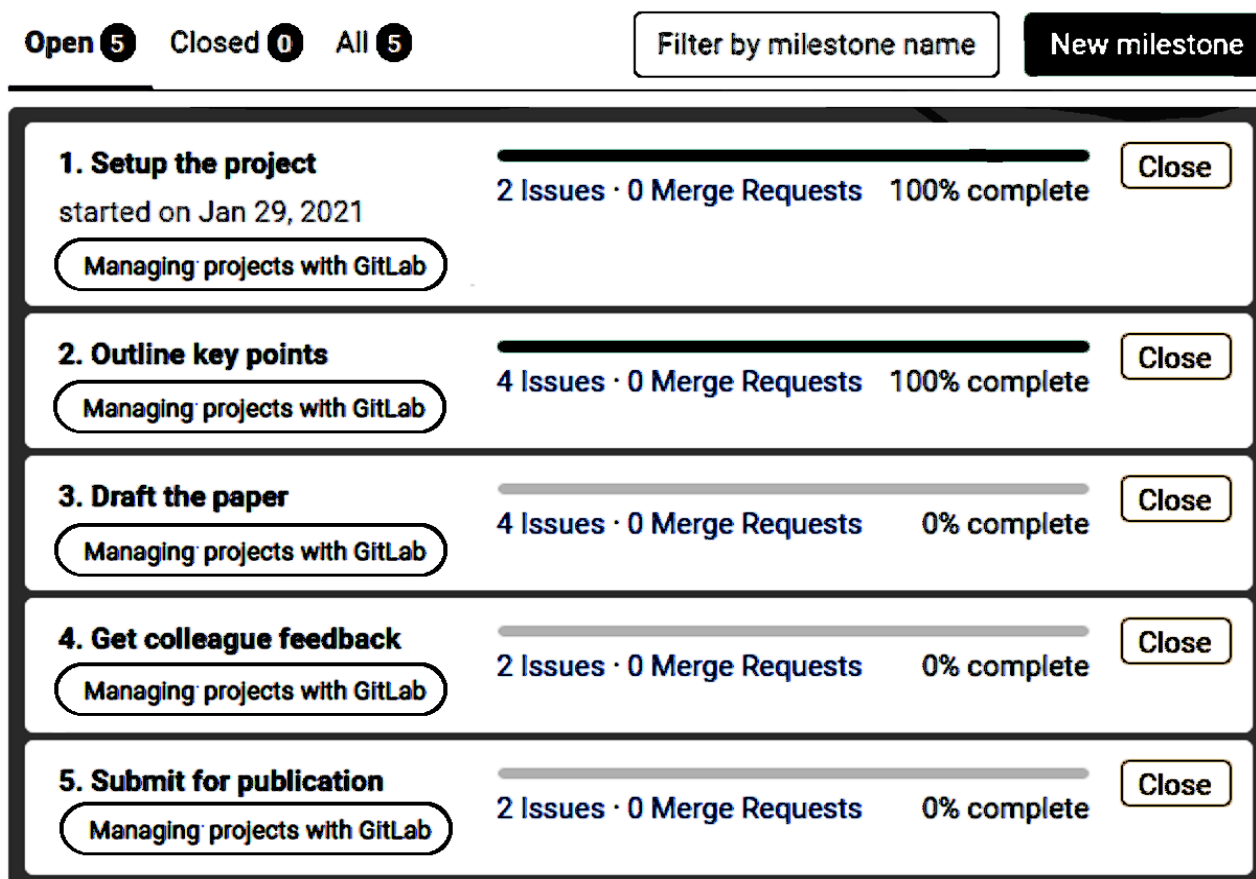


Fig. 4: A Screenshot of the GitLab Dashboard for Milestones

GitLab milestone and issue processing are very useful for managing capstone projects. The GitLab Dashboard for Milestones (shown in Figure 4) was useful for tracking the progress of a project. In this report, the status of each milestone is determined by a tally of the status of the corresponding issues. The individual tasks, functions, and features to be implemented and tested are represented as GitLab Issues and Milestones were used to represent individual stages of the capstone project that must be completed by specific deadlines.

The projects also used the GitLab Workboard (shown in Figure 5) to display and update the status of each issue within a milestone. Issues were represented as tiles that could be dragged and dropped to different

columns representing various stages of completion. While this greatly simplified the reporting of status, students had a tendency to close issues prematurely. During consultations, closed issues were often reopened to handle refactoring and overlooked bugs.

The GitLab API allowed the development of external time-triggered utilities to track progress of issues of multiple student projects. The graph of the outstanding Issues burn-down (shown in Figure 6) was developed from such a tool that interacts with the milestone data via the API. The workspace activity log also proved useful in measuring the volume and frequency of updates providing an additional indication of the diligence and efficacy of the student development effort.

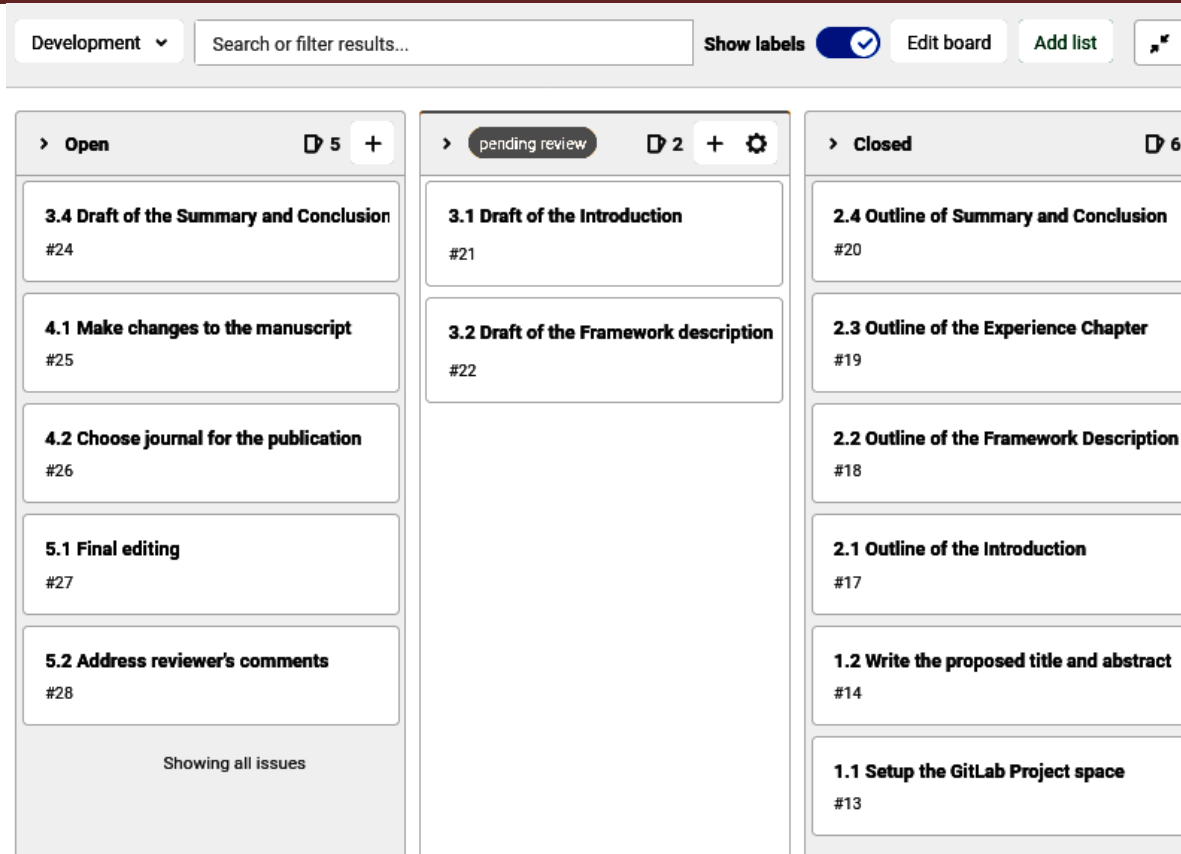


Fig. 5: A Screenshot of a GitLab Workboard

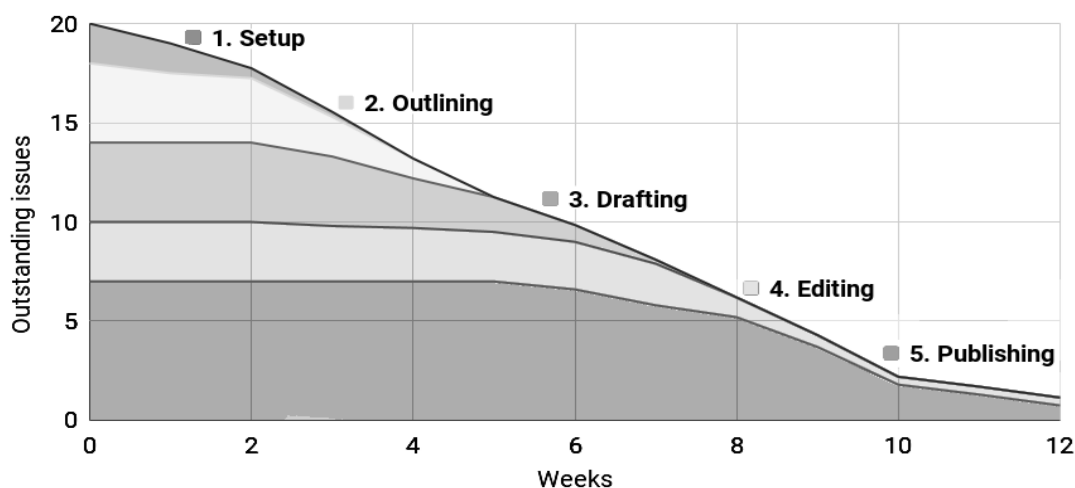


Fig. 6 Burndown of Outstanding Issues by Development Stage

Three sets of guide files were developed to provide key questions and issues to help students develop their thoughts through the various stages of the project:

- **Project Design Guides** were designed to lead the students incrementally through concept development stages leading to the Statement of Work and the corresponding plans for project development and testing.

- **Implementation Guides** were used to capture information and test results for the periodic progress reports.

- **Deployment Guides** were used as a checklist for final testing, documentation, and deployment. These guides also attempted to help student to prepare for the final assessment and successful closure of their capstone project.

Experience with This Framework

The following sections describes how the use of ISO29110-compliant documents within Git repositories has impacted the design, management, implementation, testing and deployment of student capstone projects. Initial experience suggested some improvements and after 5 years, the basic program is stable. Nonetheless, we continue to tweak various aspects of it to better meet the needs of each new cohort of students.

Guide file distribution methods

The development of the handout and guide files proceeded quickly as collaborative effort of the mentors within a GitLab repository for the capstone courses. However, the large number of

files and links between the information in the various guides complicated their distribution. Table 1 summarizes the issues of the 3 distribution methods that were attempted.

The mixed results from the different distribution methods shown here suggested that the timing and volume of the distribution is critical to success of both the design and implementation process. The value of information collected at each stage is diminished when students are overwhelmed by the volume, do not understand the questions, or have insufficient time to consider the issues. There appears to be an optimal pace for the distribution of these guides that encourages student compliance and engagement.

Table 1. Comparison of the effectiveness of 3 methods for distributing guide file

Distribution Method	Strengths	Weaknesses	Popularity w students
Downloading and extracting the contents of a Zip file	<ul style="list-style-type: none"> - Distributing the guide files was easy - The files were loaded correctly into the project directory tree 	<ul style="list-style-type: none"> - Students were overwhelmed by the number of guide files - Students often failed to use the guides to write reports 	+
Guides were distributed as assignments within a learning management system (Moodle or Canvas)	<ul style="list-style-type: none"> - Student were more engaged in the development process - More assignments were submitted on time 	<ul style="list-style-type: none"> - Submitted answers were rarely corrected or updated - Answers did not build on previous responses to related questions - Highest rate of discrepancies between prototype functionality and the documentation 	+++
Merging the guides from a branch of a Git repository	<ul style="list-style-type: none"> - The class Git files could be distributed as needed - Template files were properly filed 	<ul style="list-style-type: none"> - Students were often confused by the functions to merge branches 	-

Initial Development

The ISO29110 process is initiated with a discussion between the client and the service provider to develop a mutually agreed Statement of Work. While the project proposal was designed to serve this same function, about half of the students are uneasy sharing and discussing their ideas incrementally. Instead,

they preferred to submit a completed documents without any review or discussion. This behavior was also seen in the frequency by which the student Git repositories were updated (See Figure 7.).

In addition, students who rarely updated their Git repositories tended to store multiple local copies of each document in a confusing

assortment of directories and filenames with numerous renderings of version labels and timestamps. Comparing the differences between versions in these collections was not a trivial task. Students who ignored the version control functionality of Git found that later stages of development required significantly more effort to test and document solutions.

While time management was an issue with most students, it was often a symptom arising from problematic behaviors, such as a lack of commitment to the planning and design process, overestimation of the student's ability,

underestimation of the complexity of the project, premature focus on the aesthetics of the reports, or other competing activities. In such cases, the statement of work and even consultations did not represent the true status, intentions and directions of the project. Without transparency, issues were not discovered until they became significant problems and threaten the success of the project. Most successful projects were those that evolved and underwent regular review throughout the duration of the project.

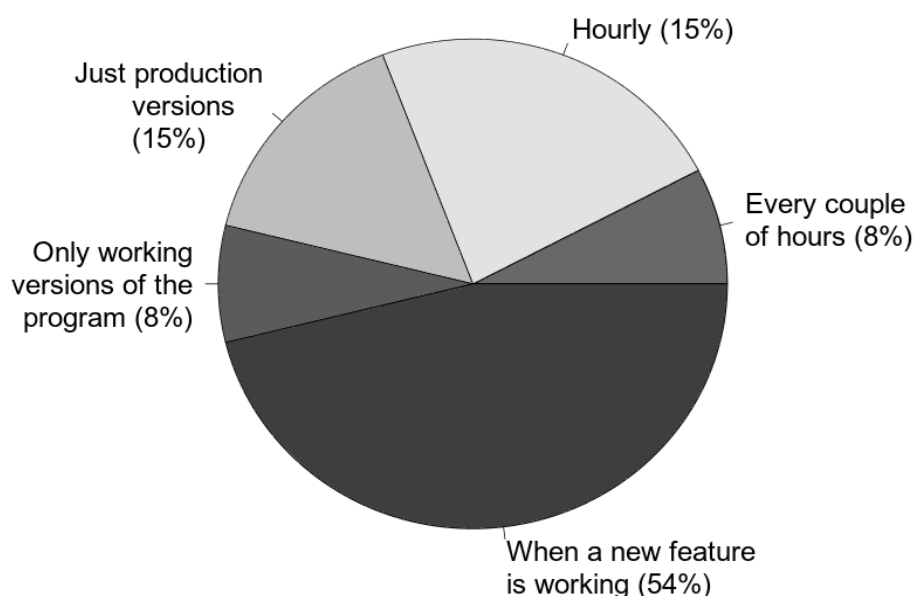


Fig. 7 Student strategies for updating Git repositories

The strategy students developed for solving short weekly CS assignments is significantly different than the approach which leads to successful conclusion of larger term projects. Two years ago, Git repositories were introduced for use with term projects in other CS courses to give students more experience with larger scaled projects. These changes in the curriculum resulted in positive changes in the nature of student use of GitLab as shown in Tables 2 and 3.

Effects of Microsoft Office Usage on the Utility of Documentation.

Key documents of ISO29110 projects are meant to act as instruments of communication to convey goals, intent and achievement in terms that are easily understood by both software developers and their clients. Most students preferred to use Microsoft Office applications save information in proprietary binary files like DOCX, XSLX, and PPTX which does not encourage cut and pasting between the documentation and the assets built in the Integrated Development Environment (IDE).

Table 2. Common uses of GitLab by students

Common uses of GitLab by CS students	in 2016	in 2018
To archive source code	45 %	62 %
To download and study code fragments	35 %	62 %
To get components of a homework assignment	10 %	38 %
To build and distribute a professional portfolio	0 %	15 %
To write papers using wiki or markdown	0 %	8 %
Never used Git	34 %	8 %

Table 3. GitLab Features commonly used by CS Students

Features of GitLab	in 2016	in 2018
File sharing with coworkers	42 %	77 %
File repository	49 %	54 %
Comparing versions	5 %	39 %
Milestone and issue tracking	5 %	31 %
Writing documentation in Markdown	20 %	31 %
Restoring previous versions	0 %	31 %
Tracking progress	0 %	23 %
Running unit tests.	0 %	15 %

In addition, MS Office features like smart quotes alters the text making it harder to compile and use in other applications. In addition, Git version control is unable to do line by line comparisons of MS Office file severely limiting the ability to list the changes made since the last consultation.

Using MS Office for generating documentation and a separate IDE application for the development of the prototype encourages a disconnect between the documentation and the prototype, and between the definition of the project, and day-to-day development and testing of the solution. If documentation and implementation could occur within the same IDE, there would be fewer discrepancies in the prototype. Yet, most students are not familiar with using marked text systems, such as LATEX, Markdown, HTML, or Wiki to generate formatted documents. Mentors also noted that when students quickly hammered out something to meet deadlines, the text was

written with little or no intention of implementing the features described. The best student projects tend to have documentation that was written and updated at each stage of the project. This occurred more often when most of the work including writing the documentation was done in an IDE. Efforts to address these concerns are still ongoing.

Value to student projects.

When students embraced the software engineering aspects of the product documentation and developed the components of the target documents, they discovered that the framework had several advantages:

- **Final documentation was easier to write.** Some students commented that writing the key documents seemed to be a natural integrated part of the process. They found it easier to write documentation when they knew what material was expected and understood its relevance to the project.

- *Consultation with the faculty advisers were more helpful.* Because the advisers had access to answers to the guide files, they were better informed about the intentions and directions of the student and could discuss the design of the project in the early stages. Students found this useful for improving the design and scope of their projects.

- *Project development was easier.* Although capstone projects are the largest projects students experience, many students felt that development seemed to progress along a systematic workflow. They felt that the time spent in design and project description was useful for identifying the main features and functions required. Student found that they were well equipped to describe their development problems and their intention clear enough that respondents on social media could provide hints that effectively addressed the issues. The framework appeared to be an accurate prediction of the effort and hurdles they experienced.

- *Students were better prepared for presentations and project assessments.* Some students felt that the process of writing the target documents incrementally resulted in a more comprehensive understanding of the project. This deep knowledge provided confidence in the handling detailed questions.

- *The project repository was a useful addition to their portfolio.* Some alumni mentioned that the GitLab repository of their senior project was helpful during their job search as it provided functioning samples of their work.

Advantages seen by faculty advisors.

Mentors recognize that successful completion of the capstone project requires skill, vision, and discipline. Instructors for capstone courses found it essential to adopt the role of coaches and mentors who can encourage and prod their students to both define what needs to be done and do whatever it takes to deliver a tested and documented solution. The development process tends to be more intensive

than students experience in other courses. Although many students tend to be shy in communicating their ideas at a time when dialogue was critical, the GitLab repository coupled with project management, support tools and guides created a platform that projects an updated, multi-dimensional image of the student project. This made it possible to review work in progress. Scrums and consultations were more effective when students updated repositories before meetings.

Students that embraced and followed the framework prepared their ideas in advance and understood their project far better than those who waited to the last moment. They were also better at articulating their goals and identifying the areas of the project that needed refinement, resulting in proposed schedules and plans that were complete and realistic. GitLab repositories also provided important hints about progress and threats to success, as well as indications as to whether the development was on schedule and critical deadlines could be met. Integrating development and documentation into the implementation process tended to result in cleaner, readable code, inclusion of more test functions, and fewer discrepancies with the proposed project. Conversely, failure to integrate design and documentation into the development process increased the risk of failure, increased false starts that required refactoring of the code.

Summary and Conclusion

As shown in Figure 8, PYU CS students enter university with a wide range of computing experience. Issues of time management, priority setting, transparency and professional integrity are behavioral factors that have a huge impact on the success of student capstone projects as well their subsequent IT career. While mentoring individual students is labor intensive, it has the greatest impact on our students as they transition into careers in IT.

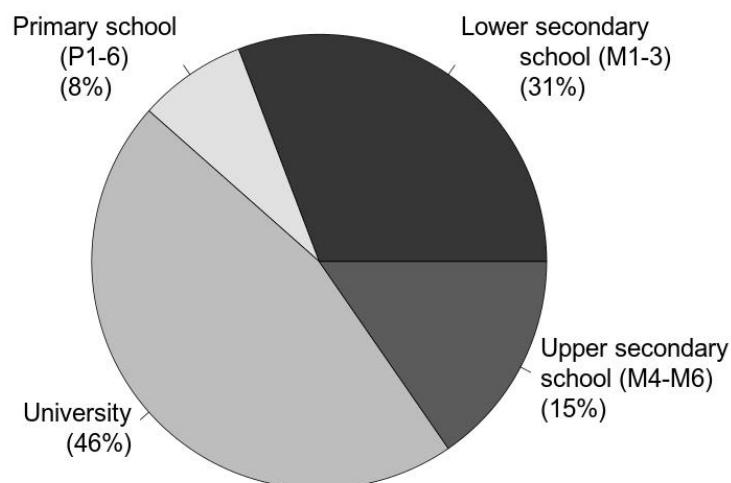


Fig. 8: The educational level at which PYU CS majors began programming computers

ISO29110 was designed to encourage developers to maintain a holistic approach in creating solutions that satisfy the requirements of users. While documented communication between users and developers is essential to satisfy legal, contractual and financial agreements, its key purpose is to ensure that the solution will truly meet the needs of the client, with minimal bureaucratic overhead. However, building solutions this way does not come naturally to all students. The results obtained in the early years of this project were mixed. For some students, the framework acted as a valuable guide to a process designed for on-time delivery of an effective, documented and tested solution. However, the value of good project management was not always recognized especially by those who are not in the regular habit of planning and managing their projects.

To these students, the structure and deadlines of the framework seemed to stifle creativity.

In other CS courses, assignments focus on getting a short segment of code to work. Testing and documentation are added last (if ever). This behavior reinforces a risky, unscalable, and unsustainable form of development. Students trained this way often offer excuses which parallels the common reasons Thai VSE managers give for ignoring ISO29110 registration[16]. (See Table 5) However, as students recognize the value of good planning and project definition, their objections subside as students find ways to overcome them. As instructors address these issues with their students, they are also becoming valuable resources for helping the local IT industry to transition and adopt international project management standards.

Table 5. Ranking of Perceived Barriers to the Implementation of ISO29110 within Thai VSEs and Capstone Project Students (Adapted from Siddoo [16])

Perceived Barrier	Thai VSE	PYU CS Students
• Limitation on time available	1	3
• Lack of experience	2	5
• Heavy documentation load	3	4
• Inappropriate for the current project	4	1
• Too complicated for the team.	5	2

Successful ISO29110 project implementation requires a professional discipline and commitment to communicate and document. Conversely, ISO29110 projects fail when there is no commitment to the documented description of the work or there is no clear plan for the development and testing of the agreed products. The experience with student projects reflect the experience of the IT industry in Thailand. In 2017, SIPA released

statistics that among general software contracts between Thai government and non-registered entities, only 25% of the contracts were completed on time, and 50% within 6 month of the agreed deadline. Another 23% of the projects were canceled. Software projects of ISO29110 registered entities experienced a success rate that was double that of the non-registered entities.

Table 6. Common Complaints of Thai IT Clients and Capstone Mentors (Adapted from Plante [17])

Complaints	Ranking of Thai IT Clients	Ranking of Capstone Mentors
<ul style="list-style-type: none"> • It was very difficult to know the status of specific requests or to get any indication of the overall progress. 	1	3
<ul style="list-style-type: none"> • Very often, there is an incident when a change is put in production. 	2	4
<ul style="list-style-type: none"> • There is a large number of faults detected by the quality assurance department 	3	2
<ul style="list-style-type: none"> • The development process is painful and the documentation produced is not particularly useful. 	4	1

Before the implementation of ISO29110-compatible agile processes, customers reported the complaints given in Table 6. Instructors has seen a similar behavior with their capstone students. The project management and implementation processes of agile ISO29110 adaptations effectively address these very issues in a way that scales well even across a large software development enterprise (17). Although principles of project management were taught in undergraduate courses such as Software Engineering and System Analysis, there is a mind-set and discipline needed for project management to work. As the department continues to revise its curriculum, it is important to find new ways to help students learn to build, test, assess, and document their ideas incrementally in preparation for their capstone project. Term projects in other courses, hackathons and other programming activities provide opportunities for building this experience. Problem solving, programming

contests and research-based instruction can also provide experience in project management and documentation. The undergraduate CS curriculum needs to be augmented with various activities that provide a foretaste of skills and approaches students will need not only for the senior project but also for subsequent entry into their IT careers.

Ultimately, the goal of CS and other IT fields of study is to develop skills that deliver effective solutions to customers, solving their problems within constraints of time and budget. This requires a holistic mindset which lies at the heart of student capstone courses. One recent alumni summarized it this way:

“When I was a student, I hated having to upload my materials regularly to GitLab. I wanted my adviser to only see the final version and I hated periodic code reviews because I knew the code was not finished. I just did not want any record or discussion of my unfinished work. Now I manage other programmers and I

finally understand the value of using repositories to share code and documentation as we build solutions together. Now I spend my days trying to get new hires to do the very things

I hated as a student. Learning to build things together as a team does not come easily, but it is necessary.”

References:

1. Joint Task Force on Computing Curricula (2013). Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. (Association for Computing Machinery(ACM), New York, NY, USA).
2. Sherriff M, Heckman S (2018). Capstones and large projects in computing education. *ACM Transactions in Computing Education* 18(2).
3. Murphy C, Sheth S, Morton S (2017). A two-course sequence of real projects for real customers in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, SIGCSE '17. (ACM, New York, NY, USA), p. 417–422.
4. Eraslan S, et al. (2020). Errors and poor practices of software engineering students in using Git in Proceedings of the 4th Conference on Computing Education Practice 2020, CEP 2020. (ACM, New York, NY, USA).
5. Muñoz M, Mejia J, Peña A, Lara G, Laporte CY (2019). Transitioning international software engineering standards to academia: Analyzing the results of the adoption of ISO/IEC 29110 in four Mexican universities. *Computer Standards and Interfaces* 66:103340.
6. Arcidiacono G (2017). Comparative research about high failure rate of IT projects and opportunities to improve. *Project Management World Journal* 6(2).
7. Software Industry Promotion Agency (2018). Implementing ISO/IEC29110 in Thailand (https://www.dga.or.th/upload/download/file_0483b39a080e0cc65114e360f2cbfed0.pdf).
8. Takeuchi M, Kohtake N, Shirasaka S, Koishi Y, Shioya K (2013). Report on an assessment experience based on ISO/IEC 29110. *Journal of Software: Evolution and Process* 26(3):306–312.
9. Tamprasirt A, Sangpa P, Suwannaroj S, Utayanaka T (2013). Thailand initial implementation of the ISO29110-VSE model (https://www.cetic.be/IMG/pdf/Thailand_Initial_Implementation_ISO29110.pdf).
10. Habra N, Simon Alexandre JMD, Laporte CY, Renault A (2008). Initiating software process improvement in very small enterprises: Experience with a light assessment tool. *Information and Software Technology* 50:763–771.
11. Laporte CY, Alexandre S, O’Connor RV (2008). A software engineering lifecycle standard for very small enterprises in Proceedings of the 15th European conference, EuroSPI: Software Process Improvement, CCIS vol 16, ed. R.V. O’Connor ea. (Springer Verlag), pp. 129–141.
12. Chacon S, Straub B (2020). *Pro Git Book*. (Apress, USA), 2nd edition.
13. Lawrance J, Jung S, Wiseman C (2013). Git on the cloud in the classroom in Proceeding of the 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13. (ACM, New York, NY, USA), p. 639–644.
14. Haaranen L, Lehtinen T (2015). Teaching Git on the side: Version control system as a course platform in Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '15. (ACM, New York, NY, USA), p. 87–92.
15. van Baarsen J (2014). *GitLab Cookbook: Over 60 hands-on recipes to efficiently self-host your own Git repository using GitLab*. (Packt Publishing).
16. Veera Siddoo, Noppachai Wongsai (2017). Factors influencing the adoption of ISO/IEC29110 in Thai government projects: A case study. *International Journal of Information Technologies and Systems Approach* 10(1):22–44.
17. Plante F, Laporte CY (2009). Implementation of an agile-ISO29110 software process in a large Canadian financial institution: Mini case study (http://profs.etsmtl.ca/claporte/Publications/Publications/VSE_Case_study_Bank.pdf).