

Submission for the 2021 Larry L. Sautter Award

Project Title: UCI Graduate Division Slate Admissions System

Submitter:

Mei-Lien Young, Business System Analyst, Enterprise Student Management Systems, Office of Information Technology,
meilieny@uci.edu

Abstract

The discussion and reflection of who shall lead (IT members or business users) and how to lead the software development to success of the system is the main force that drives the software development methodology evolution (i.e., Waterfall, RAD, AGILE). This paper demonstrates our experience of utilizing the quasi-crowd build software development methodology, using crowdsourcing, to develop the UCI Graduate Division Slate Admissions System. The management and research implications of this practice have three folds. First, collaborative learning through crowd build increases the collective efficacy in achieving the project goals. Second, the crowd build improves the customers' engagement and satisfaction of the Information System development. Third, the crowd build inspires a new partnership between IS/IT professionals and business stakeholders that could contribute to the software development methodology. Moreover, it can be a good approach for the Higher Education Institutions, a place with wisdom, to utilize the crowdsource's knowledge in the enterprise applications development.

Keywords:

Quasi-Crowd Build Software Development Methodology, Crowdsourcing, Collaborative Learning, Higher Education Institutions

Introduction

Crowdsourcing and Crowd Build Software Methodology

Crowdsourcing is a sourcing model in which organizations obtain goods, services, or ideas, from a group (relatively open) of participants. The crowd build software development methodology practices crowdsourcing to utilize the wisdom of a group for a common goal in the software system development (Stol & Fitzgerald, 2014; Dwarakanath, et al., 2015). In comparison with other software development methodologies, the crowd build method highlights the importance of parallel implementation of system design, build, testing, and deployment via crowdsourcing (Figure 1).

The challenges in achieving effective software development through the crowd build method include task management, security, task enablement, the relationship among key players and collaboration, crowd composition and management, and program management (Tajedin & Nevo, 2013; Dwarakanath, et al., 2015).

UCI Graduate Division Slate Admissions System

The first version of the UCI Graduate Division (GD) Slate Admissions System was released for the 2019 Admissions Cycle. It is a centralized integrated system to replace several distributed, decades-old, legacy Admissions Systems used by UCI Graduate Schools and Graduate Division.

The users of the UCI GD Slate Admissions System include approximately 25,000 yearly applicants, 1,500 faculties, student affairs directors, school coordinators, and GD staffs for the works of managing and reviewing 16 schools' and over 100 graduate academic programs' applications. The system scope includes processes, rules, bins, forms, events, widgets, dashboards, scripts, communications, prospects customer relationship management, and report configurations; applicants' applications, test scores, and supplemental document submissions, fee payments, and recommenders' letters of recommendation (LORs); and faculties applications and scholarship reviews, Graduate Schools and Graduate Division decisions making and management, and admitted applicants' intention to register at UCI.

Quasi-Crowd Build Software Development¹ in UCI GD Slate Admissions System

Goals and Challenges. To effectively recruit student talent for the UCI Graduate Schools, and to balance the control over policy implementation of the Graduate Division through a centralized and integrated information system are the goals of the UCI GD Slate Admissions System. To achieve these goals, the project faces several challenges and concerns. First, legacy distributed Admissions Systems, embedded in ways that each Graduate School can develop their recruitment and admissions goals, are to be replaced with a centralized integrated Admissions System, driving a unified top-down direction of the organization in a controlled manner, and suppressing the degree of individual School's autonomy. Second, Information System (IS) resources could be reallocated² among Schools regardless of the school size (big vs. small) and type (art vs. engineering) through the centralized integrated Admissions System implementation. Third, the uniqueness of recruitment and admissions strength of each school is diminished due to the IS's nature of transparency.

Crowdsourcing. Contributors in the UCI GD Slate Admissions System Project includes the Technolutions³ (Slate vendor) Support Team, Slate SMEs team recruited from University of Chicago, UCI Graduate Division project stakeholders (GD Admission, Recruitment, and Fellowship Teams), UCI Schools' Student Affairs Directors and coordinators, faculties members (volunteers who are interested in joining the project or being recruited via the personal network of Schools' Student Affairs Directors and coordinators), and Program Manager, Business System Analyst, and IT developers from the UCI OIT Enterprise Student Management Systems (ESMS) group.

Enablers. Several enablers facilitated the crowd build process for the UCI GD Slate Admissions System. First, the commitment to the UCI GD Slate Admissions System goals is the key to bonding the crowdsourcing together. It heavily relies on the trust between UCI Graduate Division and Graduate Schools, in particular, the Schools' Student Affairs Directors and coordinators who play important roles in bridging the gap between the goals of the administration and research systems. They help in promoting the project to faculties, encourage, and recruit them to join the project development works. With their help, many of the project challenges (listed above) are resolved. Second, the crowdsourcing's intention to grow together as a team via the collaborative learning and mutual helping processes through the Software Development Life Cycle (SDLC). The collaborative learning and mutual helping processes are via formal settings such as requirement gathering sessions, workshops (Slate introduction & practices, training), Joint Application Development (JAD), and informal chatting and discussions. Third, Slate⁴, a friendly web-based tool with low IS/IT entrance obstacles that enable crowd development. For example, the Graduate Schools crowd can customize the application review flow in the Bin Structure (Figure 2), and the charts and SQL scripts (Figure 3 - 5) to fulfill their needs with minimum training or help from the IT team. Its configuration flexibility also helps the project in leveraging the goals of GD top-down management and the individual School's autonomy in recruiting and admitting student talent. Fourth, the recruited University of Chicago Slate SMEs team is an important enabler that the UCI project team not only learns the lessons of University of Chicago Admissions System development (i.e., the decision of centralized or distributed settings and bin configuration) but also lends their knowledge in Slate via a weekly JAD session. The learnings of Slate knowledge are peer shared among the JAD sessions attendees and non-attendees via the workshops, trainings, and informal discussions.

Crow Build. Figure 6 illustrates the collective efforts of the crowd works parallelly through the SDLC. Sub crowds are dedicated to the tasks that meet their highest interests or fit their best strengths. For example, in the design phase, the GD Stakeholder team, together with School Student Affairs Directors, coordinators, faculties' representatives, and BSA work as a group (in several sessions) to design the bin structure, and application travel and review flows in the designed bin structure (Figure 2), the charts and queries that the GD and Graduate Schools desire, and the population rules used to protect the applications being seen by authorized users only (transparency restriction). At the same time, the GD

¹ Typically, crowdsourcing recruitment is relatively open. It can be recruited through the competition. We added "Quasi" to the software development methodology to indicate that in our project, part of the crowdsourcing is from the project assignment, and part of the crowdsourcing is recruited via the personal network or from the volunteers who are interested in the project.

² For example, the big schools have IS resources to customize the legacy distributed Admissions Systems to fit their interests (like End User Computing), but not the small schools. When the legacy distributed Admissions Systems being replaced by the centralized integrated Admissions System, the IS resources could be reallocated, or the flexibility of customization could be restricted/limited.

³ <https://technolutions.com/>

⁴ Slate is a web-based software tool and platform to empower non-technical users to configure the system. The backend SQL Servers are hosted and maintained by Technolutions. All configurations (include SQL customization) and data manipulations are via the Slate UI.

Stakeholder team, BSA, and IT developers work together to design the Slate user roles and permissions, and the provisioning and de-provisioning of those permissions with the integration of the Kuali Security Access Management System (KSAMS).

The Implications

Our practice of crowd build software development methodology using crowdsourcing in UCI Graduate Division Slate Admissions System has three management and research implications.

The Collaborative Learning and Collective Efficacy

This project, with the crowd build software development methodology, offers collaborative learning opportunities through interaction with various dynamic crowdsources. Through it, the crowdsource's collective efficacy in software development increased and this collective efficacy helps the crowd in overcoming the challenges during the SDLC. For example, by observing the Slate SMEs in building the system using Slate (in JAD and workshops), the novices can assimilate and imitate how Slate SMEs configure the modules to develop their Slate skills. As what we observe, after the system go-live, the GD business users can configure and maintain the GD Slate Admissions System with minimum help from the IT team. In addition, the Graduate Schools' Student Affairs Directors and coordinators (faculties, as well) can configure or maintain the application review flow and SQL scripts to fulfill their needs (Figure 4). Another fantastic example is how the GD business team learned to use Jira, a software management system, to manage changes and releases. The business uses it with little to no IT involvement and use it as efficiently and effectively as a well-versed software development team. Business users' reliance on the IT team to maintain the system is significantly reduced.

The Customers' Engagement and Satisfaction

User involvement in the process is an important factor in success of the software development (Bano & Zowghi, 2014). In our practice, the project stakeholders (Graduate Division Admissions, Recruitment, and Fellowship Teams) and Schools users (student affairs directors, coordinators, and faculties) are recruited as project crowdsourcing that is not only participating in the SDLC as the requirement gathering resources but also join in the actual system building activities (Schools also can request the IT support per their needs). They know their needs well and are empowered to use the tool in tailoring their portions of the processes to practice their strategies in recruiting student talent.

A New Partnership from the Crowdsourcing and the Software Development Methodology Evolution

The discussion and reflection of who shall lead (IT members or business users) and how to lead the software development to success of the system is the main force that drives the software development methodology evolution. Our practice shows a collaboration of crowd lead in the parallel implementation of the system design, build, testing, and deployment via crowdsourcing. This alliance between IS/IT professionals and business stakeholders could contribute to the software development methodology. Moreover, this new partnership from the crowd building could be a good approach for Higher Education Institutions, a place with wisdom, to utilize the crowdsource's knowledge in the enterprise applications development.

References

- Bano, M. and Zowghi, D. (2014) A Systematic Review on the Relationship between User Involvement and System Success, Information and Software Technology (2014), doi: <http://dx.doi.org/10.1016/j.infsof.2014.06.011>
- Dwarakanath, A. et al. (2015) Crowd Build: A Methodology for Enterprise Software Development Using Crowdsourcing. 2015 IEEE/ACM 2nd International Workshop on CrowdSourcing in Software Engineering, May 2015, pp. 8-14.
- Stol, K. and Fitzgerald, B. (2014) Two's Company, Three's a Crowd: A Case Study of Crowdsourcing Software Development. 36th International Conference on Software Engineering, Hyderabad, India.
- Tajedin, H. and Nevo, D. (2013) Determinants of Success in Crowdsourcing Software Development. Proceedings of the 2013 Annual Conference on Computers and People Research, 30 May 2013, pp. 173-178.

Project Crowdsources

UCI Graduate Division Project Stakeholder

#	Name	Role
1	Court Crowther	Assistant Graduate Dean
2	Ruth Quinnan	Director, Admissions & Academic Affairs
3	Audra Hansen	Associate Director of Graduate Admissions
4	Jason Cruz	Senior Admissions Analyst
5	Megan Story	Admissions Analyst

UCI OIT ESMS Team

#	Name	Role
1	Christopher Shultz	Director, Enterprise Student Management Systems
2	Rachel Tam	Assistant Director, Enterprise Student Management Systems
3	Jackson Chen	ESMS Program Manager
4	Mei-Lien Young	Business System Analyst
5	Jeff Fellows	Application Developer
6	Jon Desiderio	Programmer Analyst
7	Anchun Ouyang	Programmer Analyst

UCI Graduate Schools

#	Name	Role
1	John Romine	Director of Computing; Crowdsources
2	Schools' Student Affairs Directors	Crowdsources
3	Schools' Coordinators	Crowdsources
4	Faculties	Crowdsources

University of Chicago Slate SMEs and Technolutions Support Teams

#	Name	Role
1	University of Chicago Slate SMEs Team	Crowdsources
2	Technolutions Support Team	Crowdsources

Figures

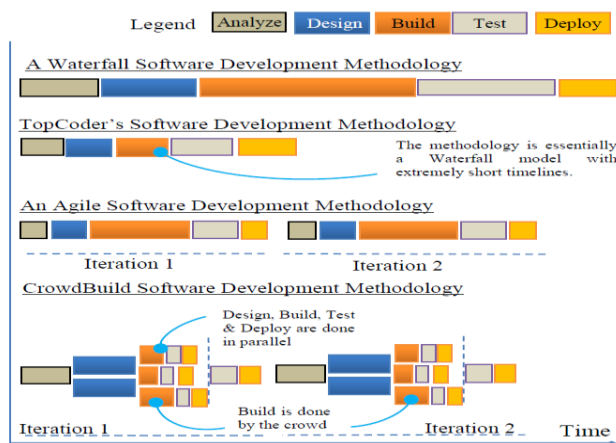


Figure 1 Crowd Build and Other Software Development Methodology (Dwarakanath, et al. 2015)

slate Browse									
Graduate Division									
Home	Awaiting Review	Review	Interview 1	Committee Review	Interview 2	Pre-Decision	Decision		
Slate	Awaiting Submission	Initial Review	Interview 1	Committee Review	Interview 2	Waitlist	Graduate Division Admit		
Search		292	140	0	89				
Queue									
Recent	Awaiting Payment	Faculty Review		Additional Committee Review		School Decision	Deny		
Share	325	488		80					
Classify						Exceptions	Withdrawn		
Help	Awaiting Materials	121				Graduate Division Decision	Incomplete		
Edit									

Figure 2 Application Review Bin Structure for All Schools

UCI Graduate Division Slate Admissions System

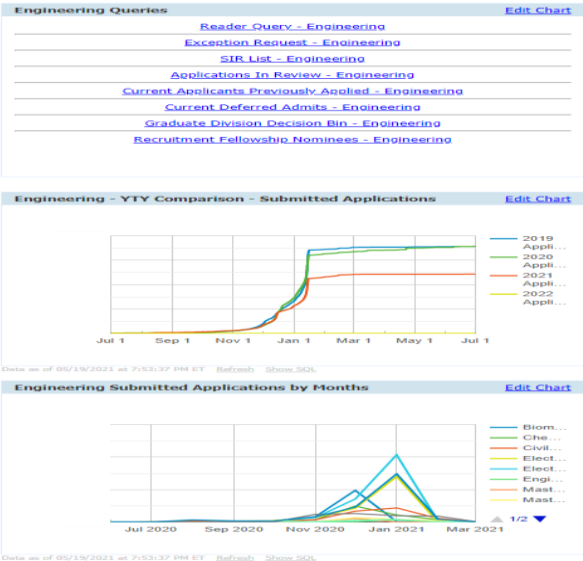


Figure 3 Widget for Engineering School

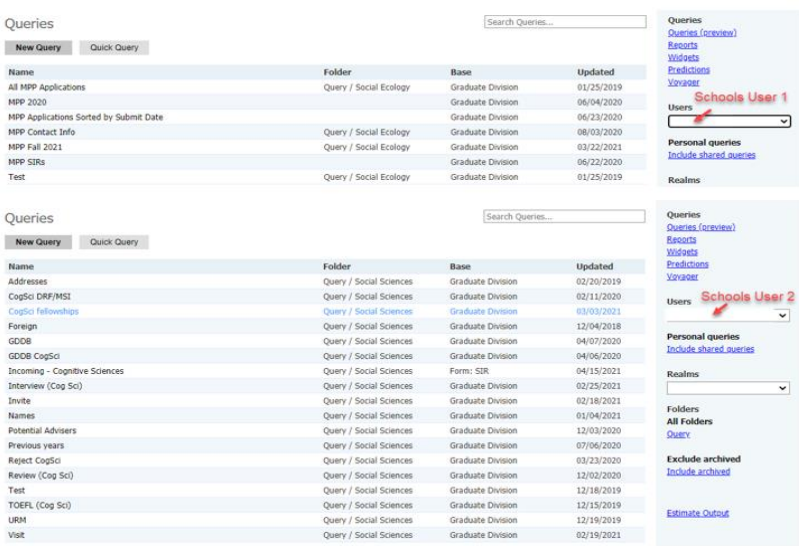


Figure 4 Examples of the Queries Written by Different Schools' Users

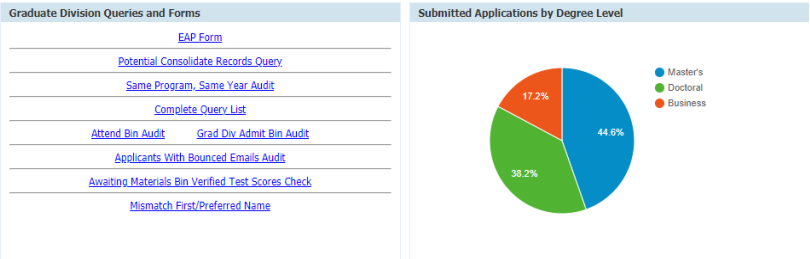


Figure 5 Widget for Graduate Division

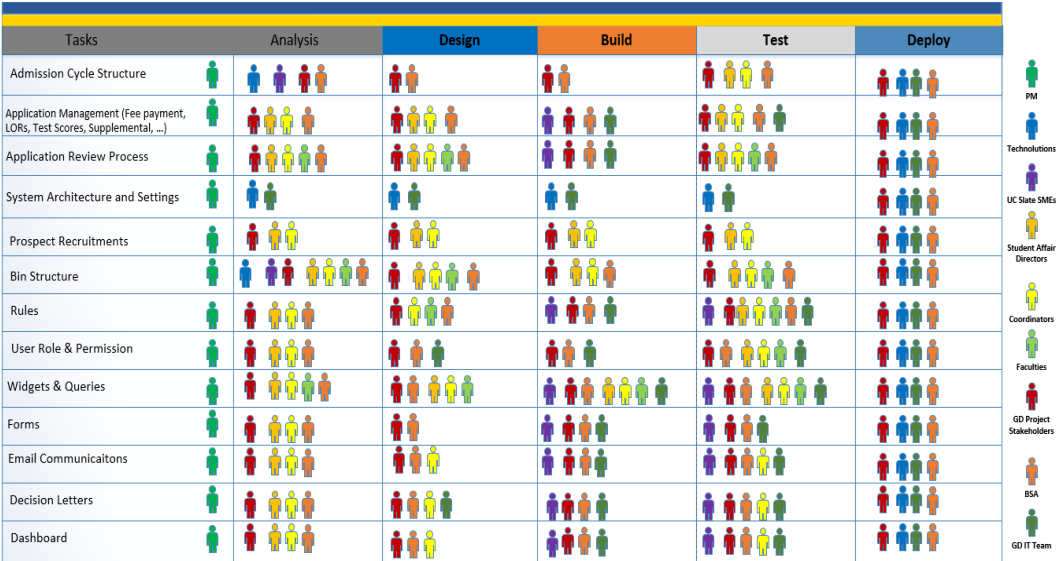


Figure 6 Crow Build UCI GD Slate Admissions System (SDLC)