



PICS System Improvement - Process & Project Plan

Overview

The Purchasing and Inventory Control (PICS) system was created in 2015-16 by a team of Kate Gleason College of Engineering (KGC OE) students to help streamline the paper purchasing order process that was currently in place for the KGC OE Multidisciplinary Senior Design (MSD) teams. While the introduction of the PICS system has been overall beneficial since its inception, the heavy usage of the system within the past couple years has revealed some areas of improvement. It has been estimated that possibly 50% of users time on the PICS system could be saved with improvements to how PICS works and functions. Requested changes include: ability to edit Purchase Requests once submitted, ability to scan incoming package barcodes to link with RIT ID's, moving the budget sheets from separate excel sheets to being directly located on the PICS system, and other changes to aid usability. Additionally, the documentation left behind by the original development team isn't clear and easy to comprehend, leaving our sponsors with no way to understand how the current PICS system is set up and running. The aim of this project is to further develop the PICS system interface to be more readable, understandable, sensible, and usable for MSD students, as well as PICS administrators. This will not only save time for students, faculty, and lab assistants involved in the MSD program, but aid their process optimization efforts by removing the roadblocks and annoying time-sinks they're currently encountering, freeing time to focus on creating the greatest projects and program possible.

Goals and Scope

Upgrading and updating the Purchase Requisition System, fixing System Permissions issues, developing a web-based version of the current Budget System, and allowing non-MSD students to check-out/purchase tools are all within scope of this project. Scanning UPS deliveries and tracking order pickups may or may not be in scope. The team will decide based on what progress is made in the coming months. Our goals are to complete the requested work highlighted above, and provide high-quality technical and non-technical documentation to the sponsors for any further maintenance.

Deliverables

The team is planning on using an iterative process to release every 4 weeks. As such the planned software release schedule is highlighted in the **Scheduling and Estimates** section.

As for non-software releases, the team plans to follow the deliverable schedule outlined by the Software Engineering department, outlined in Appendix A of this document.

Risk Management

Risk	Probability / Impact	Risk Exposure	Action
Breaking / Overwriting the current PICS system while the MSD teams are utilizing it	10% / 1 week	.07 days	Ensuring we keep our developer branch and the current live PICS branch separate, this should be easily avoidable
Requirements changes	15% / 3 weeks	3.15 days	Keeping close and constant contact with our sponsors should keep us updated on any type of requirement shifts
Scope creep	20% / 3 weeks	4.2 days	From the beginning, we have made sure to make it clear with our sponsors what we should be focusing on fixing first and what we could define as stretch goals to ensure we target the most critical areas of the system first.

Estimates Don't Reflect True Complexity Involved	40% / 3 weeks	8.4 days	We'll need to emphasise that the estimates are estimates, and are subject to change as the project continues. By prioritizing tasks with the sponsors, we can ensure that we finish the most-needed ones on time.
--	---------------	----------	---

Scheduling and Estimates

Work Breakdown Structure (WBS)

1.0 Fix System Permissions

- 1.1 Modify Enrollment to Use Active Directory
- 1.2 Allow Archiving of Teams and Projects
- 1.3 Enable Listing Multiple Guides
- 1.4 Enable Listing Multiple Financial Approvers

2.0 Fix the Purchase Requisition System

- 2.1 Add Editing Functionality
 - 2.1.1 Enable Approval of a Previously Denied Purchase Requisition and Vendor Request
 - 2.1.2 Allow Editing of Purchase Requisitions After Submission/Before Approval
- 2.2 Update Emailing Functionality
 - 2.2.1 Email Guide on Cancellation or Modification of Purchase Requisition
 - 2.2.2 Update Email Subject Lines
- 2.3 Update the Purchase Requisition List Screen
 - 2.3.1 Create Snapshot View of Each Purchase Requisition
 - 2.3.2 Enable Approval from the Snapshot View
- 2.4 Update Purchasing Form
 - 2.4.1 Add "length" and "width" Fields
- 2.5 Create Purchase Requisition History Page
- 2.6 Handle Reimbursement Requests

3.0 Build the Budget System

- 3.1 Replicate Current Functionality in a Web Application

- 3.2 Make the Correct Budget Viewable to Students
- 3.3 Allow Export of Budget Data in Excel Spreadsheet Format

4.0 Improve the Inventory System

- 4.1 Allow Non-KGCOE Students Enrolled in MSD to Check Out Tools
- 4.2 Allow for Scanning of UPS Documents with RIT Scanners
- 4.3 Track and Show Order Pickups

Overall Project Schedule

Since we're using an iterative approach to this project, the project schedule is laid out in iterations. Each iteration is 4 weeks in length. Below is a list of the tasks we will attempt to complete for every iteration.

Iteration 1 - September 25th, 2017 to October 20th, 2017

- 1. 2.1.1 Enable Approval of a Previously Denied Purchase Requisition
- 2. 2.1.2 Allow Editing of Purchase Requisitions After Submission/Before Approval
- 3. 2.2.1 Email Guide on Cancellation or Modification of Purchase Requisition
- 4. 2.2.2 Update Email Subject Lines
- 5. 2.3.1 Create Snapshot View of Each Purchase Requisition
- 6. 2.3.2 Enable Approval from the Snapshot View
- 7. 2.4.1 Add "length" and "width" Fields
- 8. 2.5 Create Purchase Requisition History Page

Iteration 2 - October 23rd, 2017 to November 17th, 2017

- 1. 2.6 Handle Reimbursement Requests
- 2. 1.2 Allow Archiving of Teams and Projects
- 3. 1.3 Enable Listing of Multiple Guides
- 4. 1.4 Enable Listing of Multiple Financial Approvers

Iteration 3 - November 20th, 2017 to December 15th, 2017

- 1. 1.1 Modify Enrollment to Use Active Directory

Iteration 4 - January 16th, 2018 to February 9th, 2018

- 1. 3.1 Replicate Current Functionality in a Web Application

Iteration 5 - February 12th, 2018 to March 9th, 2018

- 1. 3.2 Make the Correct Budget Viewable to Students
- 2. 3.3 Allow Export of Budget Data in Excel Spreadsheet Format
- 3. 4.1 Allow Non-KGCOE Students Enrolled in MSD to Check Out Tools
- 4. 4.3 Track and Show Order Pickups (first half)

Iteration 6 - March 12th, 2018 to April 6th, 2018

1. 4.3 Track and Show Order Pickups

Resource Allocation

Austin Cook: Meeting Scribe

Ben Meyers: Quality Assurance & Website Coordinator

Grant Gadomski: Team Coordinator/Leader & Sponsor Coordinator

Ryan Lachacz: Configuration Coordinator

Estimation Techniques Used

To estimate our timing on this project, we took a handful of sequential steps. First and most important was to break down all requirements into a Work Breakdown Structure (WBS), shown above. By creating this granular view of the project, we were better able to estimate, and going forward will have a much easier time tracking project progress. Next, we followed a technique similar to the Scrum Methodology's Planning Poker.

Planning Poker ensures all team members estimate the effort required at the same time. This helps to prevent ingroup-outgroup bias and ensure an individual's estimate isn't skewed by their peers. From there negotiation occurs until all members agree on a certain effort. We followed a similar approach with hours instead of story points. Since availability was short, all team members were to write down their estimates for each task individually on a document.

Once all team members finished, we met to combine and finalize estimates. We used a technique called Three Point Estimating, which takes the shortest estimate, the longest estimate, and the most-likely estimate, voted on by the team. These are plugged into the following formula:

$$E = (\text{shortest} + \text{longest} + 4 * \text{most likely}) / 6$$

The result is our estimate for a task.

Project Tracking Strategy

This project will be tracked by hours in a couple of ways. Our task estimates are in hours. We will also be tracking the amount of hours worked per person, and through these two measurements we can track our project progress.

As a team we expect to log 120-160 hours total per iteration. This boils down to 7.5 - 10 hours per week, per person. We estimate at least 80 of those hours will be devoted to development time, whereas the rest will be project management, meetings, and other maintenance tasks. At the end of every iteration we will mark the total hours worked, as well as the the total Estimated Task Hours we were able to deliver.

This provides a few indications for why the project would fall behind schedule. If the team's logging fewer than 120 hours per cycle, the issue is total time worked. If the team is logging a healthy amount of work but not delivering enough work, the issue may lie in underestimating tasks. After a few iterations the team will have a velocity, or average amount of Estimated Task Hours it's able to deliver. From there we can plan our next delivery based on our previous iterations. If the team is delivering an average of 110 Estimated Task Hours per sprint, we will plan our next iteration around delivering another 110 Estimated Task Hours worth of tasks.

Estimated Task Hours

The sum of estimated hours for a task or set of tasks.

Example: Task 1 Estimated: 8 hours.

Task 2 Estimated: 4 hours.

Total Estimated Task Hours: 12 hours.

Schedule Adjustment Strategy

No matter how confident we feel and how much we prepare, at least one of our estimates will probably be wrong. Because of this, it's important to have a strategy for how we'll react to a necessary change in schedule. Ultimately we'd like to complete every task listed in the Work Breakdown Structure above. If the schedule starts to slip due to a lack of hours work, the team will have to regroup and commit to increasing its hour count. There may be an iteration where 180 hours or more are worked, due to a lack of time dedicated in a previous iteration.

Working longer hours isn't always the best long-term solution though, and

the team members have other demanding classes outside of senior project. Working too long of hours over a long period of time could lead to burnout and apathy within the team. If the team's working a substantial number of hours but falling behind schedule, a reassessment of the product's schedule with the sponsors and coach will be needed. In this case the first step would be a re-estimation of the remaining tasks, given what we've learned about the system. We would then provide this re-estimation to the sponsors and coach, and if our estimated development time exceeds the end of April 2017 we will ask the sponsors about the priority of remaining tasks, and request to cut the low priority ones that fall outside of our schedule.

Measurements and Metrics

Time/Schedule Metrics

One of the most important roles of metrics is tracking a team's progress towards completing a project, with respect to time and a schedule. We will be using the Estimated Hours of Work Delivered Each Cycle to help team to track progress and plan future iterations. This is congruent to the concept of velocity in Agile. After a few iterations we will look at what we delivered each iteration, the hour estimates we placed on the tasks delivered, and develop an average that we can plan future iterations on.

If the team does begin to slip schedule-wise, we have a couple metrics to help understand why we're off schedule, and how to adjust. The Total Hours Logged per Cycle will be a good indicator of whether the team needs to adjust it's estimates, or simply put more work into this project. We expect to log 100 - 160 hours total per cycle as a team, which breaks down to 6.25 - 10 hours per person per week. If we're not delivering work on schedule, and hours per cycle is below our goal, as a team we know the solution is simply devoting more time to this project. If we are hitting this metric and still not staying on schedule, it's an indication that we need to adjust our estimates.

As for the accuracy of our estimates, that can also greatly affect the project schedule. Which is why the last schedule metric is the Average Estimation Deviation. With every task delivered we will compare the estimated time for said task to the actual amount of time spent, and get the average for all tasks. With this the team can detect trends, whether it's a tendency to overestimate or underestimate. From there we can decide to adjust our future estimations.

Project Management Metrics

Besides simply building the product requested by the sponsors, as a team we also must remain cognizant of other tasks surrounding this project. The team is using a technique known as the Four-Up chart to track project Progress, Plans, Risks, and Needs. This simple tool provides a surprising amount of visibility into the status and progress of the project. To ensure that we're actively contributing to and maintaining this

documentation, we will track the average number of bullet-points on each Four-Up chart. A healthy number of bullets means the entire team's engaged and has a good understanding of the project from a high level.

Ultimately this system is being built for certain people, represented by our sponsors. Therefore frequent communication is a must to ensure that the team's building the exact product that the sponsors need, and keep the sponsors clued in on the project's progress. The Number of Emails between Team and Sponsors will be tracked, to ensure that a consistent flow of communication is occurring between the two parties.

Technical Process

Given that we are going into this project with an already created system in place, we have opted to continue on with the project using an iterative / incremental methodology. The PICS system has been up and running for a couple years now, with changes primarily only needed in how accessible and usable the system should be for its administrators and users. With these details in mind, using an iterative methodology would be most beneficial in helping to continuously roll out and subsequently test features as they are implemented. The Iterative design methodology is cyclical in nature, focusing on bi- to tri-weekly releases which demonstrate the new features / changes included with each release.

Following the designated procedure of the iterative process, we have six main stages throughout our process lifecycle, two of which are only completed at the inception and completion of the project. The **Initial Planning** stage is used to get an idea for the project scope as well as a loose understanding of the requirements. The **Requirements** stage is used as a time to further pin down exactly what requirements we intend to work on during this project release. What follows next is the **Analysis, Design, and Implementation** portion. This time in the development cycle allows us to actually start looking at the internals of the PICS system, as well as think about how we want to implement a feature on our requirements backlog. After the system analysis has been conducted and the design has been considered, we finally take the time to actually implement our change into the system. The subsequent **Testing** stage allows us thoroughly test our recent feature implementations to weed out unintended behavior. In addition, this time also allows us to show the release to our sponsors for them to use, in order to give us feedback. The **Evaluation** portion permits us to elicit the feedback gained from our sponsors during their time testing our release. This evaluation time is also used independently by our development team to discuss how our release proceeded this time around, and if any changes in process or structure need to be made. The final stage, **Deployment**, is incurred after the Analysis, Design, and Implementation portion is completed, and is only reached when the team is definitively deploying their current release as their final product.

Appendix A: SE Department Schedule

Fall	Team Activities and Deliverables
Week 1	<ul style="list-style-type: none">• Attend Senior Project Expectations and Meet Your Faculty Coach session.• Hold project kick-off meeting with the project sponsor this week or next week.<ul style="list-style-type: none">◦ Get enough detail concerning the project to write the project synopsis.• Decide on a team name.• Begin tracking team time/effort.• <u>Deliverables</u><ul style="list-style-type: none">◦ Submit request to SE System Administrator to create team account providing: team name, login-ids of each team member and faculty coach.
Week 2	<ul style="list-style-type: none">• Hold project kick-off meeting, if not held in Week 1.• Start regular weekly interactions with project sponsor.• <u>Deliverables</u><ul style="list-style-type: none">◦ Start project website <u>on an se.rit.edu machine</u>.◦ Inform faculty coach and project sponsor of the URL for the project website.◦ As you understand your project better, you can also request a virtual machine for your development work, or to setup a deployment environment.◦ Publish, on project website, first tracking report of time/effort worked.

Week 3	<ul style="list-style-type: none"> ● Get approval of project synopsis. ● Discuss overall project plan with sponsor. ● Discuss development process options with the sponsor. ● Begin detailed project discussions. ● <u>Deliverables</u> <ul style="list-style-type: none"> ○ Submit project synopsis of no more than 250 words to sponsor and faculty coach for approval. ○ Publish, on project website, team information including approved synopsis. ○ Complete project information survey form. ○ Submit first draft of development process and project plan for review by faculty coach and sponsor.
Week 4	<ul style="list-style-type: none"> ● Continue discussion of development process and project plan, if needed. ● Specify development methodology. ● Specify product/process metrics to be tracked. ● <u>Deliverables</u> <ul style="list-style-type: none"> ○ Document development methodology on project website. ○ Document product/process metrics on project website. ○ Domain model on project website.
Week 5	
Week 6	

Week 7	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Mid-term peer evaluation
Week 8	<ul style="list-style-type: none"> • Mid-term project review with sponsor and coach
Week 9	<ul style="list-style-type: none"> • Motivated by the mid-term project review and peer evaluations, discuss with faculty coach adjustments individual team members and the team will make. • Discuss sponsor's attendance at the interim presentation.
Week 10	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Interim presentation scheduled
Week 11	
Week 12	
Week 13	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Draft interim presentation
Week 14	
	Thanksgiving break
Week 15	<ul style="list-style-type: none"> • Give interim presentation this week or next week • Attend required interim presentations

Week 16	<ul style="list-style-type: none"> • Give interim presentation this week, if not done last week • Attend required interim presentations • Discuss sponsor's possible attendance at interim reflection meeting • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Interim presentation evaluations for presentations attended ○ Individual end-of-term peer evaluations ○ Interim team self-assessment ○ Project website and repository on an se.rit.edu machine up-to-date with all project artifacts, tracking reports, etc. ○ Course evaluation
Finals	<ul style="list-style-type: none"> • Attend interim reflection meeting • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Summary of interim reflection meeting
Spring	Team Activities and Deliverables
Week 1	<ul style="list-style-type: none"> • State-of-the-project discussion with sponsor and coach • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Project plan updated for second term
Week 2	
Week 3	
Week 4	

Week 5	
Week 6	
Week 7	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ◦ Mid-term peer evaluation
Week 8	<ul style="list-style-type: none"> • Mid-term project review with sponsor and coach
	Spring break
Week 9	<ul style="list-style-type: none"> • Attend session on making a poster and writing the technical report • Motivated by the mid-term project review and peer evaluations, discuss with faculty coach adjustments individual team members and the team will make • Coordinate with sponsor for attendance at final presentation
Week 10	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ◦ Final presentation scheduled ◦ Project poster concept
Week 11	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ◦ Preliminary project poster
Week 12	<ul style="list-style-type: none"> • <u>Deliverables</u> <ul style="list-style-type: none"> ◦ Project poster delivered to Senior Project Coordinator

Week 13	<ul style="list-style-type: none"> • Poster presentation during Software Engineering Project Day • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Draft final presentation ○ Technical report outline
Week 14	<ul style="list-style-type: none"> • Give final presentation this week, or next week • Attend required presentations
Week 15	<ul style="list-style-type: none"> • Give final presentation this week, if not done last week • Attend required presentations • Discuss sponsor's possible attendance at final reflection meeting • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Presentation evaluations for presentations attended ○ Draft technical report ○ Individual peer evaluations ○ Team final self-assessment ○ Summary of Project Reflection meeting ○ Project website and repository on an se.rit.edu machine up-to-date with all project artifacts, tracking reports, etc. ○ Course evaluations
Finals	<ul style="list-style-type: none"> • Attend final reflection meeting • <u>Deliverables</u> <ul style="list-style-type: none"> ○ Final project artifacts ○ Final technical report ○ Summary of final reflection meeting ○ Senior survey

