The PMACS Software Development Process

Project Management Using “Scrum”
The PMACS software developers and design teams use an Agile method of project management called “Scrum.” Scrum identifies team roles, meetings, and a general method of operating with clients. Details are provided below.

The Roles

The Product Owner (PO)
The PO is responsible for the relationship with clients. It is s/he who meets with clients (s/he may bring a developer at her/his own discretion) to gather requirements, manage expectations, set priorities, and agree on deliverables and deadlines. S/he represents the client to the developers in the articulation of client needs and priorities as expressed in the prioritized backlog of user stories (see below). S/he also represents the developers to the clients in terms of providing a realistic representation of what work the client can expect to be done and in what timeframe.

The PO does not manage the developers, s/he manages the work the developers do. This includes determining in what order work will be done, but not the amount of work that will be done in a given sprint. The developers complete the user stories as they appear in priority order in the sprint backlog. The Product Owner commits not to give new requirements to the developers during the sprint unless there is an unavoidable emergency; in this case, the PO removes a work of equal point value from the sprint so that the aggregate amount of work in a sprint remains the same. Requirements are allowed to change (and change is encouraged) but optimally this should only occur outside the sprint. The PO should also keep the top of the backlog in priority order so that if the development team completes what it has committed to in the current sprint, the Scrum Master (see below) can pull in more stories confident that they reflect the clients’ priorities.

At PMACS, the PO is also generally the last person before the client to test the work done by the development team. Changes need to be expressed in a new story.

Also see below.

The Development Team
The Development Team refers to the software developers who implement the requirements agreed upon by the client and PO. The Development Team is also responsible for estimating the work by assigning ‘points’ and for describing how the work will be implemented through the assignment of tasks. Points represent the level of complexity of the work as agreed upon by the Development Team; the PO can then communicate this to clients, and clients can budget their IT effort accordingly.

The Development Team is responsible for the quality of the work it produces. Development Team members do not release work for PO acceptance until it has been tested to the point at which it could be considered client-ready. This does not absolve the PO and clients of responsibility for doing their own testing, too. The Development Team is responsible for how much work it can complete in a sprint. The Development Team also creates the internal documentation for a project; clients may use this documentation to write their own user guides.

All members of a given Development Team work on all the projects of that Team’s clients.

The ScrumMaster (SM)
At PMACS, the SM is a developer or designer. The role of a SM is sometimes described as being that of a ‘servant leader’ of the Development Team. The SM does not have authority over the Development Team members but s/he does have authority over the process. S/he monitors the progress of the work completed during the sprint, can require certain practices, and can insist on process changes to facilitate the workflow.
The SM also makes sure that the Development Team does not overcommit itself and that the PO does not try to overextend the Development Team. But s/he is also responsible for ensuring technical integrity. The SM is also responsible for helping the PO clarify requirements and maintain the backlog so that the Development Team can implement requirements successfully.

**Requirements Gathering**
The Product Owner (PO) is responsible for collaborating with clients to establish project requirements. These requirements are written in the form of User Stories with Conditions of Acceptance (CoAs).

Preferred user story format:

“A [ROLE] should be able to [ACTION] in order to [BUSINESS OBJECTIVE].”

**Conditions of acceptance**
CoAs should follow either the “Given... when... then...” or “Verify that...” format.

For example:

“Verify that the administrator can sort the results by date.”

These user stories make up the team’s product backlog. Once stories and CoAs are written, they can only be altered by the PO.

**Backlog Maintenance**
The PO is responsible for maintaining the backlog. During release planning, the effort required for each new story is estimated in “story points” by the Development Team members as a group. Story points represent an abstract estimation of the effort, complexity, and time required to complete a story.

**Prioritization**
The client is responsible for communicating priorities (within their set of projects) to their PO. The PO is then ultimately responsible for the prioritization of all stories across clients. Individual team backlogs are placed in priority order by the PO, along with target dates as needed.

**Sprint Planning and Execution**
Work is organized into two-week sprints. A team’s velocity is the average number of points completed per sprint, and it is used to roughly determine how many stories can be pulled into the next sprint.

During a given sprint, work is picked up by Development Team members in priority order. Whenever possible, a story may be shared by multiple Development Team members, each taking on different tasks within the same story.

The SM is responsible for identifying and helping facilitate the removal of any obstacles that may be reducing the team’s efficiency.

The PO may alter stories and CoAs during a sprint, but it is not encouraged. Any changes made by the PO during a sprint must be communicated to the Development Team members. If another member of the Development Team requires further clarification on a given story, that Development Team member should consult with the PO to alter the story and/or CoAs appropriately.

**Unplanned Work/Interruptions**
Unplanned work can include urgent updates, defects, or bug fixes which the PO (in consultation with the client) has determined cannot be put off until the next sprint. Once added to the current sprint by the PO, the Development Team must decide upon the number of story points. The PO must then identify a story or set of stories with the same number of points to pull out of the sprint so the Development Team is not overloaded.
“Spikes”

Spikes are defined as tasks or stories which require research before they can be estimated. At PMACS, spikes can be either tasks added to a specific story or stand-alone ‘research’ projects (or, even more mundanely, documentation projects). They are time-boxed by the PO and the developer may or may not start or finish implementing the feature within that time. Once the time limit is reached, the developer needs to confer with the PO on the status of the spike and their findings. The PO can then decide to either:

- consider the spike closed (at which point points are assigned by the developer for recordkeeping purposes)
- add more time to this spike for further research (removing other stories from the current sprint as appropriate)
- table further effort until a future sprint.

Client/Team Interaction

The relationship between PMACS and its clients is collaborative. The PO is the clients’ main contact, the ‘hinge’ between the client and the developers; as such, work requested by a client goes through the PO. This includes new projects, new features on existing projects, or alterations to newly-implemented features. The PO may ask Development Team member(s) to join her/him in consultations with clients, but their inclusion is at the PO’s discretion; the SM may have recommendations, however, with regard to how these consultations are affecting the efficiency of the Development Team, and the PO should take these recommendations into consideration.

The PO is ultimately responsible for all communication with the client. It may be delegated to other team members for better efficiency, but the PO must be kept in the loop and be prepared to intervene when needed.

The POs facilitate efficient, effective software development. The POs rely on their clients’ ability to provide a clear understanding of the ultimate goals of the project, the audience served, and the functionality needed. While the application may have many users and multiple types of user, it is important that the PO is working with those who can make decisions about features, including their relative importance. For a project to move forward, a PO depends on these decision-makers to be in attendance at meetings, to provide timely and decisive responses to questions posed by the PO/team, and to test the project thoroughly within the requested timeframe, using realistic data and scenarios before giving feedback to the PO.

Decision-making about particular projects lies with the client. Each PO helps each client understand that client’s own project backlog, but defers to the client in the prioritization of its projects and individual items. (As noted above, it lies with the PO to prioritize across clients.). The PO and her/his clients also work on longer-term planning. In this, the PO needs the client to be able to contribute fully in the planning and prioritization, especially when hard deadlines or external constraints are involved.

The development process is often iterative. After initial requirements are gathered, a feature is built and then shown to the PO for review. The PO’s role in this review process is to represent the client. Alternatively, the PO may choose to take it directly to the client and gather feedback to take back to the Development Team in the form of a new story (if refinements are deemed necessary).

The PO’s role is also to keep the “big picture” in view at all times. For example, if a project appears to be getting bogged down in “feature/scope creep”, the PO may suggest to the client that lower-priority features or revisions be put off to a future version, or that end users are consulted when clients are not sure how best to represent them.

In short, then, clear, timely communication between client and PO and PO and developers/designers is vital for good software development.