ScrumTime 1.0

Definition

ScrumTime is an agile Application Lifecycle Management (ALM) system that centralizes client billing, team management, project definition, client interaction, and project sustainment. This is an open source product that is highly scalable and extensible.

The UI is composed entirely of HTML, javascript, and CSS. The primary javascript library that makes this possible is AngularJS. The data is supplied via RESTful interface with JSON data format.

Release Schedule

1.0 GA (2015)

Blogging

Ernie Paschall

Features

Identify the Project

- Objective
- Define the Actors
- Define Scrum Teams
  - Define Scrum Roles
  - Assign Roles to ScrumTime Users
- Create Team Wiki
  - Allow clients to suggest features that go directly to the parking lot
  - Allow clients to set a voting priority of backlog or parking lot items
- Artifact Library
  - Upload / Manage artifacts

    - Associate artifacts with Features, Ideas, Epics, Stories, Bugs, Wikis, Team Members, Teams, Releases, or Iterations

Plan the Project

- Manage the Backlog (epics, stories, bugs)
    - Manage / Create Epics
    - Manage / Create Stories
        - Manage / Create Tasks
- Identify the Source Control System
    - GitHub
    - Git
    - SVN
    - CVS
    - Source Safe
    - Clearcase
    - Team Foundation Server
- Set Iteration Defaults
    - Length
    - Naming pattern
    - Identify typical workdays and time of day
- Maintain / Generate Iteration Schedule
- Maintain a Parking Lot
    - Features and Ideas that are nice to haves
    - Features and Ideas go through an approval process to be added to the backlog
    - Features and Ideas become Epics or Stories based on the approval process

Execute the Project

- Maintain Current Scrum / Iteration Backlog
    - No Epics allowed
- Maintain the Story Board
- Document the Daily Scrum
- Manage bugs

- Bugs are allowed to go into the current Scrum backlog, the main backlog, or not associated
- Bugs may be associated with stories, tasks, or nothing at all

Sustain the Project

- Manage Support Tickets
- Manage / Generate Client Billing
  - Bill based on flat rate or based on hours completed
  - Create estimates of future work based on the item filters in the backlog
- Create a Client Wiki

Report on the Project

- Burn Down Chart
- Burn Up Chart
- Resource Reporting
  - Team member work load
  - Team work load
- Alerting
  - Epic status change
  - Story status change
  - Bug status change
  - Support Ticket status change

Design

Layers

- MVC UI in Browser - AngularJS, jQuery, HTML, CSS
- Network - JSON / AJAX
- RESTful Server API - Implemented with either Grails or ASP.NET (decision coming Jan 30, 2014)

Future

I would like to find someone interested in doing an Android app, an iPhone app, and/or a Windows phone app. These apps will be relatively simple because they will reuse the RESTful Server API. It will not make sense to begin these efforts until the API is available. I expect that to be June 2014. Please email:epaschall@bytefirm.com if you are interested.