Fermilab

Lessons Learned

ESH&Q Operational Readiness Clearance Notable Project

Version 0.8

12/01/2016

DocDB # 5727

PREPARED BY:

Robert D. Kennedy

Table of Contents

1. Overview

This document contains the Lessons Learned from the ESH&Q ORC Notable Project. In our project management process, we stress inclusiveness and completeness in Lessons Learned documentation rather than vetting each observation or suggestion. Statements expressed here do not necessarily reflect the opinions of the Project Managers, Project Sponsor, nor Computing Management, nor do they necessarily reflect a consensus view of the project team. The intent is collect all potentially useful input from the project team and stakeholders with little filtering to help guide future phases of this project as well as other projects.

2. Lessons Learned Contributors

Lessons Learned sessions were held in the 10/25/2016 Core Team Meeting. Input was also accepted by email if received by 8:00 AM Central on 11/7/2016.

Core Team Meeting – Lessons Learned Attendees

- Rob Kennedy
- Keenan Newton (teleconference)
- Eric McHugh
- Kimberly Myles
- Mandy Rominsky
- Bob Sieloff

Email Contributors to Lessons Learned, if not already listed above:

- (none)

3. What We Did Well

1. The project was a success. The ORC Notable was met: a consistent across-the-lab ORC policy was defined and implemented in FESHM. Consistent TSW and ORC processes were defined, and implemented in a user-acceptable ORC/TSW tool.
2. The project Core Team included a broad mix of experienced functional and IT experts who worked together effectively even as the project hit road bumps while executing.
3. The project worked closely with several field testers to broaden our experience outside of the CTR test center with this relatively new approval workflow tool.
4. Project developers worked very hard to deliver not only this project as planned, but also an unexpected higher priority project, EPRA, that was initiated shortly after this project began executing. Fortunately, work on each project could be re-used on the other, so there was not a doubling of effort to accomplish these projects. Nevertheless, the development team had to put in long hours in order to meet the challenging and effectively immutable deadlines of both projects simultaneously.
5. Using a developer time budget approach proved effective in facilitating the project Core Team's creating realistic feature release plans.

4. What We Could Have Done Better

6. SDLC-based Development Process: We suffered multiple delays in prototype testing due to tool breakdown. The tool worked on a Wednesday in a demonstration to show that we were ready for organized testing, but was broken in that broad testing session on Friday. This occurred more than once, and was in each case due to a developer working on new features after the Wednesday demonstration, breaking the tool. There were no separate tool instances to support simultaneous development and testing.
    a. In the future, the SharePoint Development team should follow a well-defined development and release process with multiple tool instances, such as (at least) Development, Testing, and Production. Two instances are not enough. Enterprise Applications, for instance, maintains more instances for their larger systems such as Training, Integration, and so on.
    b. More instances may be desirable if parallel development of workflow and user interface features are desirable, to allow each sub-system to evolve without "stepping on each other's toes". Then the Testing instance can serve as an integration testing platform. This is well-documented SDLC methodology.
    c. There should be a documentation change process for the
7. Testing sessions in early to mid-project involved significantly modifying the design as the tool behavior was observed in situ. The project had agreed-to design artefacts describing the static tool design, but they were not sufficient to specify the tool behavior and communications timing. We lacked a design specification of the tool's dynamic behavior expressed at a high level that all team members could discuss, agree to, and test against. Approval flowcharts was requested but not delivered by the BA assigned to the project until much later than expected due to personal leave and lack of familiarity with business process design. Eventually the PM drafted approval flowcharts and state diagrams to reduce the churn we had observed in each major test and release cycle. However, this occurred AFTER the initial release of the tool, a release which was not widely used as it was not quite trusted yet to support the TSW and ORC processes.
    a. Project Management Process: The project had what it thought was a sufficient design to drive implementation. This design however proved to be focused on static aspects of the tool and not the dynamic aspects that are often captured in flowcharts and state diagrams. The project ended up working on both the dynamic design and the tool implementation iteratively, which was much less efficient than developing a well-specified draft design first, then working on a tool implementation. The PM process should have been more waterfall-like for the dynamic design and iterative for the user interface. Some iteration and/or refinement is still expected in the design, but we could have eliminated a fair amount of rework by the developers had we developed a more mature approval process design earlier in the project.
    b. Process Design Specification: The project had static design documentation, but lacked approval flowcharts and tool state diagrams until relatively late in the project timeline. There was no common specification of behavior, understandable to implementers and users, that could be used for guide implementation and then specify the desired state for unit and integration testing. We had to demonstrate the tool to users who then provided feedback based on a mix of what they remembered requesting weeks earlier and what

they preferred now that they saw the tool in action. This led to some churn and rework, and was an inefficient use of project team members' time.

   c. Resource/Skillset Availability: When the tardiness of the requested dynamic design specification was recognized, the PM requested an alternate resource from the Computing Sector BA team to assist in delivering the process design specifications. However, no other resources were available who were familiar with business process design or requirements gathering of this sort. Relying on past experience in software systems design, the PM had to develop the needed specifications. There should be a clear understanding of what is expected of a project BA role for future projects to help PM's gather the necessary skills and resources to execute a project. Also, the PM should have escalated this sooner, as soon as the initial design delivery dates were missed, irrespective of why those delivery dates were missed.

8. Lexicon and Use-Cases: Role names in the TSW/ORC approval processes varied a lot in the early phase of the project due to different viewpoints and functional language. Different role names are now ingrained in the implementation which could cause confusion for maintainers. We needed a team member role to help users define requirements in a way that everyone can understand, and explain what is going on in the design and the tool. Early focus should be on developing a common lexicon, and testing that lexicon through use-case development to ensure a common language is used in the design and implementation of the tool. Use-case development not only tests the lexicon, but also assists in developing the dynamic process design specification.
9. Naïve User Feedback: The project engaged a naïve user for feedback in User Acceptance Testing. While there was much to filter from that user's feedback, there was also a lot of value in it too. We could have engaged more naïve users in a separate session to get more feedback and be more efficient in introducing the constraints on the tool to help guide users to more effective feedback.
10. Project Initiation: How/when did this effort become a project? I was asked to participate in a meeting on an effort by ESH&Q and then was surprised to find that I had a role and responsibilities in a project to build a tool in SharePoint.
    a. While the project had a kickoff Core Team meeting on 3/22/2016 which explained the project organization and process, more could have been said about how the project idea was developed by ESH&Q and Computing management well after the ORC Notable Outcome effort by ESH&Q began.
11. Demand/Requests Management: There were many new requirements that our developer tried to accommodate before a formal process was put into place to review/approve updates for a particular release under a particular change request. We are still receiving major enhancement requests. For example, RDCoordinator Brian Rebel asked to have multiple Division selections (one- division-only is the current state). This would take significant effort and refactoring of the solution.
    a. How do we formally manage these requests after the project closes out? The project has suggested tracking requests in ServiceNow and setting aside a certain number of developer hours per quarter per tool and then prioritizing requests to fit the developer budget. However, it is up to the service provider/line management to decide how to proceed.
    b. Should we have a TSW ORC Steering committee similar to the Content Management Group's Steering Committee and bring all enhancement requests to this group for review/ approval before approval for a future release?

5. Outstanding Risks and Opportunities

12. <u>Risk</u>: SharePoint-based approval workflow tools are sensitive to users using out-of-date web browsers, email browsers, and/or mobile devices.
    a. We do not and cannot control which browsers are used outside of the Fermilab organization, nor can we completely control which browsers are used inside the Fermilab organization.
    b. Mitigation: Document clearly what browsers are supported. Inform users if they are using an unsupported browser so that they at least know this up front.
    c. Mitigation: We should consider how to harden the tool on the server side for use of unsupported browsers.
    d. Mitigation: As with other services, as time permits, test common mobile devices in order to document whether the tool works well for some roles on such devices. We can do ourselves a favor setting user expectations appropriately. Note that there is a demand for tablets to be more usable by approvers for fast, convenient approvals on a device that is large enough to review attached notes.
13. <u>Risk</u>: The current approach to assigning Service Providers (SharePoint Development Services) as the initial responder to service tickets is based on a model developed for large Enterprise Applications rather than small approval process automation tools. This approach will not scale well for a small development team creating and supporting potentially dozens of such approval process automation tools.
    a. This requires some discussion of how the balance of new tool projects and support for existing tools will be managed by this team. At some point, the team effort will be entirely consumed by fixing or evolving the existing tools with little or no time left to develop new tools.
    b. This requires considering how Computing delivers "small" services specific to Fermilab business units that are not themselves fully on-boarded for IT Service Management in ServiceNow. Once a business-specific tool is reasonably mature, most requests/incidents will be related to the policy and process supported by the tool and not the operations of the tool itself.
14. <u>Risk</u>: The Service Provider has not yet defined how they will manage the requests received from users. This could lead to stagnation of the tool and possibly reduced use of the tool if alternatives are found.
    a. The project has suggested a scheme to budget developer hours quarterly to each such approval tool, some more than others, and prioritize a quarterly release schedule with the appropriate customer. However, this topic is the responsibility of the Service Provider line management.
15. <u>Risk</u>: TSW/ORC lacks a procedure to adapt to Fermilab organizational changes.
    a. This is planned for post-project, with a procedure and tables to simplify adaptation to org changes. However, approvals in flight may have to be adjusted manually or re-entered.
16. <u>Opportunity</u>: To better support future engagements like this one, we recommend developing a template to guide those seeking to design, develop, and deploy an approval process tool like this. The template would include example design documents that combine the experience of this project and the EPRA project. This may help guide customers to begin work on an approval process specification before engaging the development team to create an initial tool. This will help jumpstart future projects by

providing real examples from delivered production tools. Having common formats for the specifications will also assist in the long-term maintenance of the processes by customers and of the tools by service providers.