An Agile approach to specifying DCAPs: Complementing the *Singapore Framework*

Proposal for a session at DC 2009

Document Details

Author:	Paul Walk
Date:	02/07/2009
Version:	1
Related Documents:	

Acknowledgments

UKOLN is funded by the MLA: The Museums, Libraries and Archives Council, the Joint Information Systems Committee (JISC) of the Higher and Further Education Funding Councils, as well as by project funding from the JISC and the European Union. UKOLN also receives support from the University of Bath where it is based.

Background

Introduction

UKOLN has been charged, by the JISC, with the task of encouraging and supporting the development and take-up in the UK of a series of Dublin Core Application profiles. The most prominent of these is the Scholarly Works Application Profile (SWAP) which, while being relatively well-developed, has not yet seen significant adoption. For this reason, we have concentrated our recent efforts on SWAP, as we attempt to examine issues around its adoption or lack thereof.

Usability

While recognising that the range of ‘real-world’ deployment of SWAP (in repository systems for instance) is still quite limited we have, nonetheless, noted a lack of evidence to indicate that SWAP’s *usability* in such systems has been investigated or tested.

We suggest, furthermore, that the usability of an application profile cannot be effectively tested until it has been realised in a system. In the case of SWAP, this means in all practicality that it cannot be really tested until it has been presented to a user via a software user-interface (UI).

Lessons from software engineering

Agile development

While not universally applicable, the *Agile* approach has steadily gained currency in the world of software development. A number of new techniques, practices and tools have been developed in support of this. Agile development tends to favour working solutions over future capabilities and encourages near-continuous engagement with users during the development process, responding to changes in functional requirements as both the developer *and* the user increase their understanding of the problem space.

Test-driven-development

Test driven development (TDD) is an approach to software engineering which emphasises the continuous testing of software *while it is being developed*. Where a more traditional approach would leave testing to the end of the main period of development, more modern *agile* methodologies suggest that this is too late in the development process. Agile, and specifically TDD approaches encourage the developer to consider how the software will be tested *at the beginning* of the development process, and to put in place a test plan before the main development work begins.

This approach relies heavily on clearly articulated (and measurable) functional requirements. Agile development recognises that these can change during the development process and has no problem with changing the test plan to accommodate this flexibility, so long as the plan is implemented.

The net effect of this approach is that software is tested continuously, typically during very short development iterations, which reduces the risk of software ‘drifting’ away from its agreed functional requirements.

Applying Agile methods to AP development

We suggest that these lessons from software development might be advantageously applied to the development of application profiles. Application profiles, by definition, should be aimed at a clearly defined problem-space and it ought to be possible to elicit clear functional requirements from users through a process of close engagement.

Getting to grips with APs

We have observed an issue with users' engagement with application profiles. Application profiles are, essentially, intangible - users cannot interact with them directly. For many users, this presents a very real barrier to engagement. Even if formal documentation such as Description Set Profiles (DSP) is developed during the development iterations, it tales a certain kind of user with a particular interest to engage with these. Many users need to see the sort of system interface which they will ultimately be using in order to contribute feedback on the development of an AP.

Making APs tangible

We have already experimented with two broad approaches to this.

1/. In early stages of requirements gathering, a paper-prototyping approach has shown real promise as an accessible method for eliciting requirements from groups of users. This has the advantage of being potentially very free-form, such that the developer's unconscious influence on users' contributions is reduced.

2/. Once the developer's understanding of the requirements for an AP in development has reached a sufficient level of clarity, it can be prototyped rapidly in software to produce a UI *facade*. This facade can be presented to users' in a more controlled session, increasing mutual understanding of requirements and eliciting more specific requirements.

In the case of the second approach described here, we have utilised stock rapid-prototyping tools to produce Web UIs which the user can engage with. We are learning how to differentiate between feedback which is merely an artefact of the process of creating an interface which is only a facade, and the crucial intelligence resulting from the users' greater appreciation of the problem space arising from this facility allowing them to engage with it in a tangible sense.

We have used these approaches to 'test' the current version of SWAP, recognising that these approaches really should be applied to an earlier stage in development.

Proposal for a session at the DC Conference, 2009

(Working) title

An Agile approach to specifying DCAPs: Complementing the Singapore Framework

Purpose

In this session we would like to present on our recent work with developing application profiles using innovative Agile methodologies borrowed from software development practice. We would like to explain how we intend to situate this ongoing development into the broader context of the Singapore Framework with a view to complementing this community-sponsored approach to application profile development.

With a particular focus on usability and testing, we will also take an overview of the technical and social issues that can limit uptake of a given application profile, and discuss strategies that may enable them to be overcome.

Intended audience

- anyone with an interest in the development of application profiles which are 'fit for purpose' in terms of their being used with software to satisfy real user-requirements
- anyone with an interest in developing processes to facilitate the *functional requirements*, *domain model* and *usage guidelines* stages of the *Singapore Framework*
- anyone with an general desire to see an increase in the 'take-up' of application profiles

We would plan the majority of the session to be accessible to a remote audience, depending on the facilities available at the conference venue. Presentation media could be made available in advance. We would be amenable to the session (or parts of it) being recorded (audio and/or video) for dissemination thereafter, at the discretion of both the conference organisers and the participants.

Format

We propose a two hour session, comprised of the following:

1. A short presentation outlining UKOLN's recent work in:
   - applying rapid prototyping techniques to the early stages of functional specification
   - developing UI facades to test early iterations of AP development
2. A demonstration of some of the software tools which have been used in this approach
3. A presentation proposing how these approaches might be used to enhance the early-mid stages of the Singapore Framework
4. An open opportunity for the community to discuss this, raise questions, voice concerns etc.

We would like to create an environment within which delegates might be able to interact with some of the tools and techniques. Depending on time and circumstances, it might be possible to run an exercise in either prototyping something new, or usability-testing an AP through a pre-prepared UI.