Agile Data Curation - Technical Debt Conceptual Model

February 17, 2015

Technical Debt - Integration into an agile curation conceptual model

The concept of *debt* in software development was introduced in 1992 by Ward Cunningham in the context of the adoption of object-oriented principles in the development of financial software. In his formulation

> Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. Objects make the cost of this transaction tolerable. The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt. (Cunningham 1992, 30)

Cunningham focused on the development flexibility that is gained by accepting some degree of technical debt that is understood to require repayment some time in the future.

With the emergence of agile software development methods based upon the principles and spirit of the *Manifesto for Agile Software Development*:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> *Individuals and interactions* over processes and tools
>
> *Working software* over comprehensive documentation
>
> *Customer collaboration* over contract negotiation
>
> *Responding to change* over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more. (Beck et al. 2001)

The tradeoffs presented in the *Manifesto* provide opportunities to accrue technical debt throughout an agile software development process that has been recognized as a challenge that must be addressed when adopting agile methods. As noted by dos Santos et al.

> Technical debt has been a central theme among researchers and practitioners in the last years as an alternative perspective for software development and maintenance decisions. It offers a real world metaphor which is naturally understandable by mose software stakeholders and serves as tool to evaluate the tradeoffs between proposed enhancements, corrective maintenance and technical/non-functional improvements. (Santos et al. 2013, 124)

As we consider the application of agile development models to the process of data curation, the concept of technical debt may prove useful when attempting to identify the requirements for and progress towards enabling reuse of research data for communities of users and uses beyond those for which data were initially collected or created.

Assumptions

The conceptual model developed here has the following assumptions embedded in it. It is envisioned that violations of these assumptions will tend to complicate the model, but wouldn't necessarily invalidate it.

1. *Data products in use by the original community of users that created/collected those data are assumed to have no technical debt.* This assumption translates into a scenario where the effort required to make the data discoverable, understandable and usable to its creators has already been expended. A corollary to this is that there exists a group of initial users of a data product for whom those data are already serving some need. *It may be appropriate to relax this assumption as it relates to technical debt associated with future reuse of the initial data products of a research program.*
2. *Enabling reuse of a data product by a new community of users and/or a use for which those data products were not created requires additional effort.* While the level of effort required for a specific community or use will vary, some level of effort must be expended.
3. *There is a minimum level of effort that must be expended before any defined reuse may take place.* Prior to the expenditure of this minimum effort reuse by a specified community for a defined purpose cannot take place.
4. *Some additional effort may accelerate or increase the use of a data product within a specified community for a given use.* This effort is above and beyond that required for initiation of use ((3) above), and may reach a point of diminishing returns in terms of use when compared to effort.
5. *Through the application of deliberate design strategies the incremental effort required to enable reuse in subsequent reuse scenarios may be reduced.* This assumption is founded upon the principle of design for reusability and support for open standards whenever possible, with the cumulative impact of investments for use by more uses and communities providing more existing capabilities that may be reused by new users.

Types of *Effort* that are considered

When considering the reusability of data products, several key dimensions (and examples of how those dimensions translate into more specific requirements) related to those products may be translated into *effort* required to enable reuse:

Discoverability Metadata content that, when properly indexed and searched, enables members of a community of users to find the product and consider it for use

Availability of searchable metadata in locations that are accessible to and used by the intended community of users

The existence of search interfaces that the intended community of users understand and are capable of using

Machine readability of metadata and accessibility of search interfaces for integration into other applications and services

Metadata content and search locations and interfaces may support both search (the location of data resources that are known to be relevant to a particular problem) and discovery (the [sometimes serendipitous] discovery of data that are of interest to a particular researcher or applied user).

Understanding Metadata content that allows potential users to understand suitability for purpose in the context of their anticipated use

Other documentation or training materials that enable potential users to gain sufficient understanding of specific data products to enable effective use

Use Data products available in formats commonly used by the intended community and use

Data services available that are consistent with those that are used by the intended community

Data volume/size that is manageable by the intended community

Actors / Responsible Parties

Given that different participants in the data curation process possess specific knowledge and skills it is necessary to identify the roles of different actors in the agile data curation process:

Scientific Data Expert These experts have significant knowledge of the data that are going through the agile curation process. This knowledge may be based upon their role in creating the data or upon their obtained familiarity with the data through primary or secondary use and research.

Data Curation Expert These experts have expertise in the preparation of data products for long-term preservation, discovery, access and use by users beyond the initial researcher or research team that created them. This expertise includes knowledge of relevant documentation standards, data formats that enable effective preservation and reuse, and standards for data discovery and access.

Reuse User The reuse user has expertise in the new research, application or other reuse scenario and is able to effectively participate in the identification of the needs, barriers and potential solutions for achieving reuse of a dataset in a new use scenario/community.

While this is not a comprehensive list of the participants in the process of agile data curation, it is a high-level categorization of the actors between whom responsibilities for enabling data reuse must be divided, without whom it cannot be accomplished.

Illustration of the conceptual model

[Figure 1 about here.]

The context sensitivity of the definition of a "use metric" for a particular data product or collection of products suggests that the development of a strategy for developing a composite use metric that integrates multiple (appropriate) use metrics would be a necessary part of the depiction of the composite life of a particular dataset.

References Cited

Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, et al. 2001. "Manifesto for Agile Software Development." Web Page. http://agilemanifesto.org/.

Cunningham, Ward. 1992. "The WyCash Portfolio Management System." Journal Article. *SIGPLAN OOPS Mess.* 4 (2): 29–30. doi:10.1145/157710.157715. http://delivery.acm.org/10.1145/

160000/157715/p29-cunningham.pdf?ip=129.24.141.226&id=157715&acc=ACTIVE%20SERVICE&key=B63ACEF81C6334F5%2E1447575D8884B3D4%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=627568194&CFTOKEN=96951395&__acm__=1424207594_46cadf2bbba757a439f5d049bb8529b2.

Santos, PauloSérgioMedeiros dos, Amanda Varella, CristineRibeiro Dantas, and DanielBeltrão Borges. 2013. "Visualizing and Managing Technical Debt in Agile Development: an Experience Report." Book Section. In *Agile Processes in Software Engineering and Extreme Programming*, edited by Hubert Baumeister and Barbara Weber, 149:121–134. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38314-4_9. http://dx.doi.org/10.1007/978-3-642-38314-4_9.

List of Figures

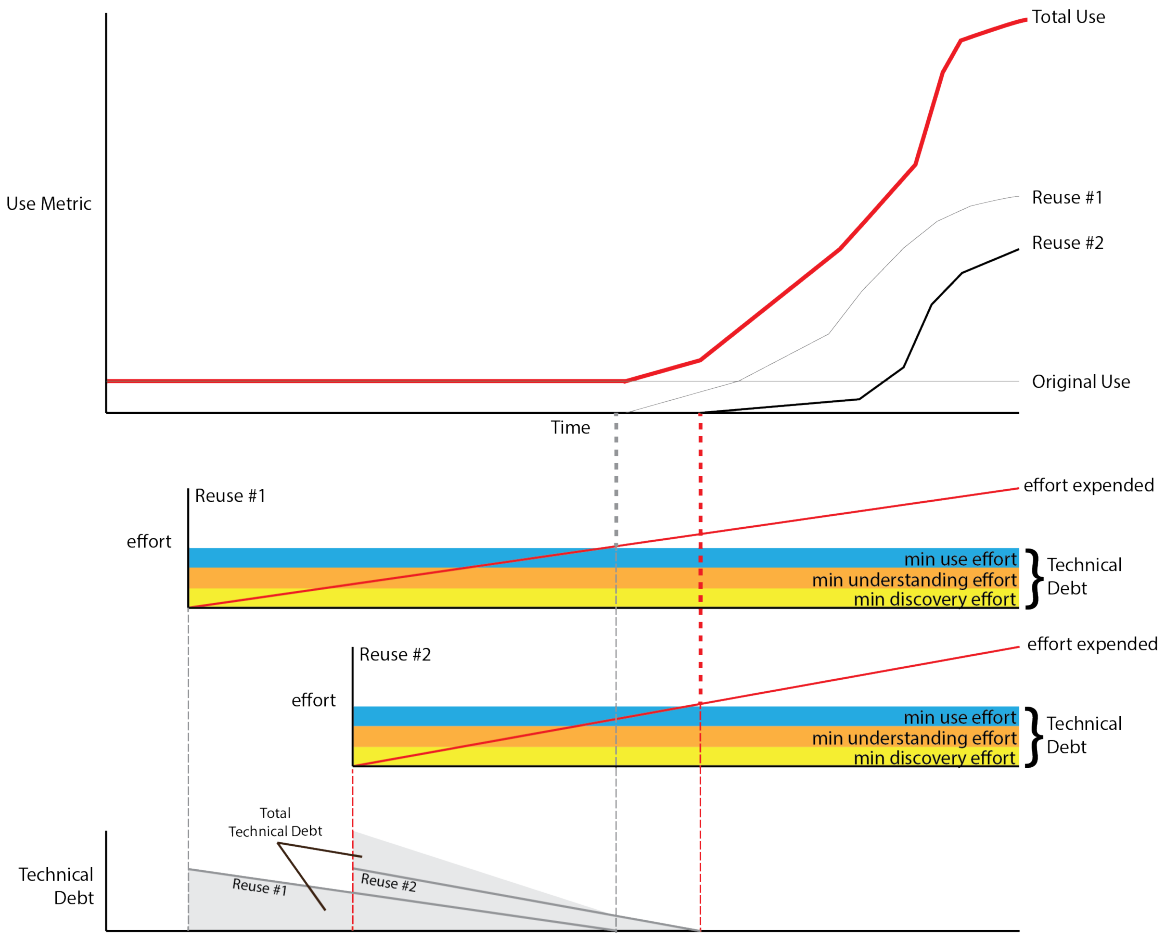

Figure 1: Technical Debt Illustration - the *use metric* defined in the plot(s) will vary by use scenario, but may include: number of downloads, volume downloaded, number of individual users, number of service requests, number of dataset citations, number of active API keys, etc.