DHQ: Digital Humanities Quarterly
2017
Volume 11 Number 1

A Hybrid Model for Managing DH Projects

Edin Tabak <edin_dot_tabak_at_postgrad_dot_curtin_dot_edu_dot_au>, University of Zenica

Abstract

A project is a basic unit of digital humanities (DH) scholarship, which suggests that DH as a discipline should pay more attention to project management, and perhaps to develop the project management models, principles, and methods that are more specific to the discipline. However, DH literature deals with this issue merely by listing the basic principles, or offering specific “tips and tricks,” which seriously simplifies management of DH projects. DH projects involve building (or at least using) digital tools, which brings the complex tensions between digital and humanistic aspects of these projects. In order to address such a complexity, there is a need for a model for managing DH projects that will learn from information studies and methods in software development, while still being based on values of the humanistic tradition and methods. This article combines a model of scholarly information practices with some concepts of agile software development into a hybrid model for managing DH projects.

Introduction

Digital Humanities (DH) is scholarship that is commonly practiced through collaborative work, organised in projects. Burdick et al. claim that such team-based approach challenges traditional humanities, which are still based on individual authorship [Burdick et al. 2012]. They point out that although knowledge has always been produced and accessed in fundamentally distributed ways, there is a myth presenting the humanities scholars as solitary genius, working alone on a creative work. In a DH project, however, collaborative knowledge production is more obvious, as a single person cannot possibly carry out all aspects of the project. It is a common practice “for dozens of people to work on a Digital Humanities project, each contributing domain-specific expertise that enables a research question to be conceptualized, answered, and then re-conceptualized and re-answered” [Burdick et al. 2012, 49]. For this reason, one would assume that project management is an important focus of the discipline, and that there are already some project management models, principles, and methods developed, which are more specific to the discipline. 1

However, in spite of such significance of project-based work for DH, it seems that complexity of managing DH projects is still not fully recognised within the discipline. Reed notes that project management in DH is usually presented from a beginner’s perspective, offering merely “basic principles,” “tips and tricks,” or “top-ten lists” [Reed 2014]. Such an approach to project management is likely to render a too simplistic trajectory of any research project, and in particular a DH project since working at the interface between humanities research questions and evolving digital methods makes the trajectories of DH work more difficult to capture than those of traditional scholarship [Brown et al. 2009]. A tip, a trick, or an advice might address a specific issue, but managing a complex interplay between digital and humanistic aspects of DH projects requires consideration of insights and methods from different disciplines. This article will combine the insights from the humanities and information studies with methods in software development in order to develop a hybrid model for managing DH project. The model will be based on an existing model of scholarly information practices, and it will adopt its main elements from agile project management methods. 2

The article will first discuss an argument of some DH scholars that project is a basic unit of DH, suggesting a need for developing a model for managing DH projects. As project-based scholarship of DH differs from traditional scholarship in being experimental, modular and incremental, and involving many dimensions of digital design and technology. The 3 second section will introduce agile project management methods as appropriate for managing such features of DH projects. However, the agile methods, being primarily methods for managing software development, put greater emphasis on digital rather than research aspects of DH projects. A customer orientation of the agile methods conceptualises their main target group as users, while the main target group in DH are the humanities scholars, who are supposed to be not only users but also co-creators in DH projects [Svensson 2009]. In order to make a balance between digital and the humanities sides of DH projects, the third section will propose a model of scholarly information practices, based on *actor-network theory* (ANT), to be combined with the agile methods into a new model. Finally, the new hybrid model for managing DH projects will be presented. The final section will describe the main elements of the model, and discuss the proposed stages in the lifecycle of DH projects.

Project as a Basic Unit of DH

DH is project-oriented scholarship. DH scholars organise their undertakings in terms of projects as collaborative research endeavours, which are, as the verb 'to project' suggests, future-oriented [Brown et al. 2009]. Burdick et al. consider project as the basic unit of digital humanities [Burdick et al. 2012]. They argue that a DH project is an academic work, which like any other project requires design, project management, negotiations, and collaboration. However, a DH project is also projecting in a futurist sense as something that still does not exist, and in which project team members contribute complementary skills and interests in order to conceptualise research questions and to design possible ways to answer these questions. DH projects are thus “projective, involving iterative processes and many dimensions of coordination, experimentation, and production” [Burdick et al. 2012, 124]. 4

Burdick et al. point out that despite many difference between projects of digital and traditional humanities, there are many similarities among them [Burdick et al. 2012]. Both digital and traditional humanities involve practices of analysis, critics, interpretations, editing, historical research, relations between individual and society, etc. DH also deal with qualitative features of human experience such as complexity, ambiguity, media specificity, and subjectivity. Although DH extends beyond textual media, the aim of DH is to be based on values of the humanistic tradition: tendency for analytical rigour and clarity, building of effective arguments, rigourous use of evidence, and communicative expression. 5

However, an issue on which DH differ from traditional humanities is an understanding of contribution of digital media to the humanities in general. DH does not try to neglect print culture, or to merely translate it into digital media, but rather tries to develop “print-plus” or “post-print” models of knowledge [Burdick et al. 2012, 125]. Such a model involves much more than simply adaptation to digital media, that is a cognitive and epistemological modification of the humanities, and the increasing role of team work and collaboration in the humanities research projects. On one side, print offers a literary organisation and research results that are characterised as stable, and a size of document (and consequently argument) that is confined by the physical size of a book or other printed material. On the other side, digital technology enables a rapid move of view from one to another angle. It enables a fluid change through zooming from macro to micro, and vice versa, and mixing of different kinds of data into research results. It enables flittering and versioning of corpuses and multilinear forms of argument. DH projects are considered more as process-based rather than product-based. Burdick et al. point out that when a book goes to print, it stabilises research results, and if we want to make change we will need to wait for the next edition [Burdick et al. 2012]. Digital artefacts can be changed at any moment, and they can have different lives in different platforms at the same time. Such artefacts can be mixed with other media, they can be changed by others during and after a project. Therefore, DH projects should have different design than projects in traditional humanities. 6

DH projects frequently involve building (or at least using) digital tools, which foregrounds the embedded tensions between digital and humanistic in DH discipline itself. Svensson argues that DH is dominated by two basic trends: one focusing on building digital technology tools, and another, focusing on using digital technology as a communication tool in the humanities [Svensson 2010]. McPherson defines these trends as the computing humanities approach, concerned with digitising and preserving the human records, so the focus is on building tools and infrastructure; and blogging humanities approach, focusing on using new modes of peer-to-peer conversation to disseminate the scholarship [McPherson 2009]. She identifies multimodal DH as an alternative approach, which is “about navigating new pathways through scholarly materials that can transform the questions scholarship might ask” [McPherson 2009, 122]. The clash 7 of assumptions between traditional and digital humanities is marked by tensions between digital and humanistic, thus between different ways of thinking and practice of computing and the humanities. Svensson claims that these tensions have been created by different epistemic commitments which create different modes of knowledge. These tensions have a great impact (implicitly or explicitly) on managing DH projects [Svensson 2012].

The definition of management itself adds another tension to the complexity of management of DH projects. Brown et al. note that while management practices are commonly defined as carefully planned practices to achieve a particular aim, many successful DH projects have not been designed to achieve a very specific aim [Brown et al. 2009]. They claim that the success of these projects are not in relation to particular aims but their success is measured in other ways. The reason for this is the multi-faceted nature of much DH research, which tries to balance between traditional humanities content and innovative technological experimentations. Such an experimental nature of DH research makes modularity and incrementalism the main features of DH project management. While traditional humanities tend to disseminate at least partially complete arguments through relatively stable print media, DH projects produce only provisional stability of argument. So just as software products are marked by a release number, “which provide the triumphant moment of celebration while suggesting that more is yet to come, designing projects to incorporate such incrementalism by way of staged releases that mark phases of accomplishment or a number of discrete and in some way publishable deliverables, seems a particularly useful way to structure digital projects” [Brown et al. 2009]. These common features of software development and DH projects (modularity and incrementalism) suggest that management of DH have something to learn from methods in software development and *information system* (IS) design, while still being based on values of the humanistic tradition and methods. 8

As DH projects include building or using digital tools, it seems reasonable the claims that the user studies should be utilised in their management. Warwick notes that it was not a common practice to study users in DH, as there was an assumption that the traditional humanists were not competent about digital technology [Warwick 2012]. A common strategy was to provide users with a digital tool, train them to use it, and hopefully they would adopt it. However in the last two decades, there was a number of studies to investigate information practices of scholars in the humanities. The general findings of these studies suggest that scholars from the humanities have different information behaviour than scholars from natural and physical sciences, and that it is more difficult to design digital tools for the humanities scholars. Kemman and Kleppe show the benefits of involving user research in managing DH projects. They argue that the identification of “the out-of-scope user requirements” can inform us about the digital tool’s compatibility with existing research practices, while “the user requirements that were within-scope led to usable features that were sufficiently generic for the tool to be adopted, also for purposes for which it was not specifically intended” [Kemman and Kleppe 2015, 71]. In an earlier study [Tabak 2008], I have illustrated how awareness of users' non-purposive information practices can contribute to the successful development of an organisational intranet. This article will argue that management of DH projects could benefit from insights of the *information behaviour* (IB) studies. 9

The significance of projects for DH and their specific features in relation to traditional humanities projects suggest that DH as a discipline should pay more attention to project management, and perhaps to develop the project management models, principles, and methods that are more specific to the discipline. However, Reed points out that managing DH projects is often in practice considered as a “soft” skill and it is positioned in opposition, or even as competition for funding, with “hard” skills such as programming and software development [Reed 2014]. She claims that such a distinction on soft and hard skills is a result of difficulties in evaluation of project roles for which a distinction between implicit and explicit skills could be more adequate. The most typical roles in DH projects are technical expertise (we could identify it as D in DH) and subject matter expertise (H in DH). Both roles present much more explicit modes of knowledge than the implicit skills of project management, which as such often go unexamined and under-appreciated from both D-side and H-side of a DH project. Reed notes that even when project management principles are made explicit to DH practitioners, they are usually presented from a beginner’s perspective, offering merely “basic principles,” “tips and tricks,” or “top-ten lists,” seriously simplifying management of DH projects [Reed 2014]. In order to make management of DH projects explicit, this paper will offer a model for managing DH projects, based on insights from information studies and agile project management methods. 10

Agile Project Management

There is something similar in the trajectory of the introduction of the Agile methods in software development and IS design with the trajectory that led to renaming the field of humanities computing to digital humanities at the beginning of this century. A major driver for both trajectories was a dissatisfaction of practitioners from these fields with the treatment of their main target groups: the humanities scholars in DH, and customers in the case of software development. This similarity could be explained by the focus of both areas on development of computing tools. Svensson argues that such an instrumental orientation led the humanities computing to neglect its main target group - the humanities scholars [Svensson 2009]. Renaming humanities computing to digital humanities has shifted the focus from the computing to the humanities, which became the central noun rather than merely an adjective. Almost at the same time, agile methods have been defined by the *Agile Manifesto* [Beck et al. 2001]. According to Abbas et al., only 2 of 12 principles of the Agile Manifesto were relatively new ideas as the “Agile ideas” have been around since at least 1970’s [Abbas et al. 2008]. Both principles are related to the main target group of software development - the customers. One of these principles proposes that developers should welcome changing requirements for the customer's competitive advantage, while another requires that developers and customers must work together daily throughout the project. This suggests that a major intention of both renaming humanities computing to DH and the Agile Manifesto was to involve users and other stakeholders more explicitly in the project activities. 11

The Agile Manifesto have been proposed as a reaction to the traditional methods in software development, such as the waterfall methodology, which is a sequential design process consisting of several stages: conception, initiation, analysis, design, construction, testing, implementation, and maintenance. Such a sequential process requires that developers can move to the next stage only when a current stage is completed. During 1990s, many practitioners found this linear process frustrating and difficult to implement in the new context of rapid technological change where “customers have become increasingly unable to definitively state their needs up front while, at the same time, expecting more from their software” [Cohen et al. 2004, 3]. In such a context, the practitioners tried to develop methods that will respond to this change. In 2001, the representatives of these methodologies defined their common values and principles in the Agile Manifesto, differentiating themselves from traditional methods. 12

The manifesto proposes four core values for such methodologies. The Agile Methods value individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. This does not mean that there is no value in tools, documentation, contract negotiation, and planning, but that the focus of these methods are on individuals and interactions, working software, customer collaboration, and response to change. There are 12 core principles of Agile Methods: The highest priority is on the satisfaction of the customer achieved through early and continuous delivery of valuable software. The Agile Methods welcome changing requirements, even late in development, as they harness change for the customer's competitive advantage. The working software should be frequently delivered, preferably in the shorter timescale. Customers and developers must work together daily throughout the project. Projects should be built around motivated individuals, who should be empowered. The best communication within a development team is face-to-face conversation. Working software is the primary measure of progress. Agile processes promote sustainable development. Agility is enhanced by technical excellence and good design. Simplicity is essential. For these methods, the best projects emerge from self-organising teams. The team should regularly reflect on how to become more effective, then tune and adjust its behaviour accordingly. 13

The Agile Methods are iterative, modular, and incremental. The product is delivered through multiple software increments. The work is usually organised into a “backlog” of existing requirements. The agile methods are adaptive rather than predictive. They harness changes in requirements, and realise that plans are usually short-lived. They rather focus on customers' scenarios of process, so they rely on use cases and use stories. The most preferred communication is face-to-face meetings, commonly short and frequent. In any case, the active user involvement, feedback and learning are crucial for the agile methods. They all recommend some forms of regular retrospective meetings both within the team and between the team, customers, and other stakeholders. An underlying intention is thus to involve the stakeholders more directly in development. 14

The agile methods are organised in stages or phases, similar to the processes in other software development methodologies, such as planning, management, evaluation, and control of iterative and incremental processes. 15

However, these processes are designed specifically to cope with rapid change, team empowerment, and active stakeholder involvement. The teams are small and self-organised, so that decisions can be made faster. Most of the agile methods use time-boxing, a planning technique, where the schedule is divided into a number of separate time periods so that each period can have its own deliverables, deadline, and budget. In Scrum method, for example, the work occurs in a “sprint” (a time-box between 2 and 6 weeks). The focus is on frequent delivery of products or features of the final products. In the Feature Driven Development, a feature is conceptualised as a client-valued function that can be implemented in two weeks or so. Such a modular and incremental approach is a most common characteristic of the life-cycle of different agile methods. While all agile methods are based on the common values and principles, each of them has its own software development life cycle [Bhalerao et al. 2009]. Table 1 represents some of the agile methods with the proposed iterative stages in the life-cycle of an agile project, and the main features of each method.

Method	Stages	Main Features
XP [Beck 1999]	Exploration, Planning, Iterations to Release, Production, Maintenance.	Customer-oriented; small team size; frequent deliverables; reflective improvements.
Feature Driven Development [Coad et al. 1999]	Develop overall model, Develop feature list, Plan by feature, Design by feature, Develop by feature.	Object development model; system implementation in features; short iteration (max 14 days); feature teams; regular inspections.
Adaptive Software Development [Highsmith 2000]	Speculate, Collaborate, Learn.	Continuous learning; feature-based and iterative; self-organising team; mission-driven development; speculative planning.
Scrum [Schwaber 2004]	Sprint Planning, Sprint (Development), Daily Scrum, Sprint Review, Sprint Retrospective.	Backlog; process management; daily stand-up meeting, retrospective meetings; a month long iterations (sprint).
Flexible Project Management [DeCarlo 2004]	Visionate, Speculate, Innovate, Reevaluate, Disseminate.	Real-time communication; change-tolerant culture; virtual meetings; focus on collaboration and scoping.
Agile Project Management [Highsmith 2004]	Envision, Speculate, Explore, Launch, Close.	Iterative delivery; overlapping development phases; strategic plans and capability analysis.

Table 1. Some of the agile methods

16 While these stages of different agile methods could be easily conceptualised into a general framework with typical stages in any project (such as planning, development, evaluation, and deployment), the terminology of a particular agile method is not a trivial issue. The naming of stages as visioning, speculating, innovating, collaborating, or learning stresses the significance of specific features of agile projects such as adaptive planning, extensive collaboration, constant learning, incremental development, and modularity. These features are also common to DH research projects. Project-based scholarship of DH differs from traditional scholarship in being experimental, team-based, iterative and ongoing, rather than fixed or final, in its outcome, involving “many dimensions of conception, design, coordination, and resource use that build extra layers of complexity onto the traditional approach to humanities research” [Burdick et al. 2012, 130]. Dyba et al. claim that agile project management is particularly appropriate for managing such complexity and uncertainty [Dyba et al. 2014].

17 This suggests that we could easily adapt iterative cycles of an agile method, for example Flexible Project Management framework [DeCarlo 2004], to define the cycles of a typical DH project. The framework starts with the Visionate cycle, which identifies the project objectives, benefits and risks, as well as possible future scenarios, producing a collective vision of the project and its deliverables. The Speculate cycle defines a plan for interim deliverables and milestones, however tentative these may be. The focus of Innovate cycle is on rapid development and real-time feedback loops. The work during this cycle takes place within a series of time boxes, usually one to six weeks. The Reevaluate cycle is a series of "learning" iteration, in which a decision is made about the future course of the project. Since in an agile project management, the goal is not to produce the planned but rather the desired result, what we are “seeing here is a

continuous cycle of planning, deplanning, and replanning” [DeCarlo 2004, 220]. Finally, the Disseminate stage is about publishing the project deliverables, but it includes also maintenance, so it is important to “review how the deliverable will be supported and maintained” [DeCarlo 2004, 398].

We can identify these stages in many DH projects. A DH project starts with a vision that identifies the project objectives, benefits and risks. DH projects are iterative, modular, and incremental, in the same way as agile projects. For this reason, it is important to speculate in order to define a plan for interim deliverables and milestones of a DH project. The naming of the next cycle as “innovate” rather than “develop” cycle is also appropriate for a DH project as a scholarly project. The process of reevaluation, defined as a series of learning iteration, is even more crucial for a DH project. Both innovation and evaluation cycles require constant learning and real-time feedback loops. While both in agile and DH projects, the goal is not to produce the planned but rather the desired result, the outcomes of a DH research project are more unpredictable. The continuous cycles of planning, de-planning, and re-planning are thus even more frequent in DH projects. Finally, the name of the dissemination stage could not be more appropriate for the final stage of a DH project. Moreover, this stage in the Flexible Project Management framework refers not only to publishing of the project deliverables, but it also include maintenance, which addresses an important issue in DH practices that “tend to focus heavily on project planning rather than project management” [Reed 2014]. 18

However, there are several issues, which prevent some aspects of agile project management to be easily translated into DH research projects. The agile methods assume that there is a customer on the other side waiting for a product to be used, while in a DH project, there is a researcher constantly creating research questions rather than a list of features to be implemented [Bezerra et al. 2014]. To be used in a DH project, agile methods should be adapted to such an exploratory innovative research environment, in which each single iteration leads to new insights, and thus to changing not only users' requirements but also perspectives and aims of the project [van Zundert 2012]. Way et al. argue that the customer orientation of most agile methodologies requires the most adaptation [Way et al. 2009]. They claim that the most significant factors that contrast an academic research project that involves software development (which is a typical DH project) with an industry software development project are variability of schedule and unpredictability of progress in academic settings. In addition, the customer orientation of the agile methods conceptualises their main target group as users, while the main target group in DH are the humanities scholars, who were supposed to be not only users but also co-creators of DH tools [Svensson 2009]. This suggests that an adoption of an agile project management to DH projects should also address some limitations of user-centred approach of the agile methods. 19

The issue of the variability of schedule in academic settings could be easily addressed by simply modifying some concept from agile project management to be compatible with a research environment. For instance, Bezerra et al. propose *ARDev* (Agile Research Development), a methodology to support research management, based on Scrum principles [Bezerra et al. 2014]. The methodology provides a valuable modification of the Scrum roles and work products in a context of research project management, which, to some extent, addresses the issues of variability of schedule in academic settings. The four roles in ARDev are Research Starter, ARDev Master, Research Team, and Stakeholders, corresponding to the main roles in Scrum: Project Owner, Scrum Master, Scrum Team, and Stakeholders. Work products are defined as results of the research activities, and include: technical report of the research project, list of research tasks per iteration, research block list, research iteration planning meeting, research iteration review, research iteration retrospective, and regular meeting. Another methodology that adapted Scrum to research management also addresses the issue of the academic schedule. Hicks and Foster propose the SCORE (Scrum for Research) methodology, in which the Scrum meeting structure has been adopted to academic settings, resulting in improved research group identity and productivity [Hicks and Foster 2010]. The model, described in the last section of this paper, will propose similar modifications of roles and work products in DH projects. 20

The second issue in adopting agile project management to DH projects is unpredictability of progress in DH research projects. Way et al. point our that research projects involve a constant redefinition of requirements as new research findings often lead to new definitions of the original goals [Way et al. 2009]. Such unpredictability of the progress of a project makes the research process inherently unsuitable for an agile project management methodology. However, they claim that despite this unpredictability, the core values and principles of agile methods can be adapted and applied to research projects, and propose the “Agile Research Penultimatum,” an attempt to adopt these values and principles to a 21 research settings. The most notable is the adoption of the agile principle stating that the simplicity is essential. They suggest that while simplicity is still essential in a research project, a recognition that complex problems often require complex solution is even more essential. The false simplicity should not be a shortcut for avoiding perplexity, which is often a departure point for any research, in particular for the humanities research. Therefore, a model for managing DH projects should harness both change and perplexity. This could be achieved by making retrospective evaluation as a constant process within each cycle of DH project instead of treating it as a separate cycle, which will be discussed in the final section of the paper.

However, a greater issue for adopting agile project management to a DH project is a suggested higher degree of involvement of the stakeholders as co–creators in DH project than in a typical software development project. The humanities scholars as the main stakeholders of DH projects are considered not only as users but also as co–creators [Svensson 2012]. Although the aim of agile methods is to involve the stakeholders in development of the product, the customers are rather conceptualised as users than co–creators of the product. The main techniques for involving the customers in development include modelling user requirements, use cases, and user stories. While such a user–centred approach may offer a more effective framework for a typical DH project than traditional project management methods, it still brings technological instrumentality, which is more appropriate for humanities computing than DH projects. Gasson argues that the most user–centred approaches in IS design, including the agile methods, focus on “closing down” a technology–centred and goal–directed IS problem, rather than on “opening up” the social and organisational context to examination [Gasson 2003]. She claims that even in an agile approach, which is a worthwhile attempt to deal with misunderstandings between technical designers and users, the design problems usually remain unexamined after they are determined early in the project. Although an agile approach provides techniques that may generate new system requirements and goals during the life of the project, “it does not question the essential form and social role of the technical system, as even this approach focuses on an individual ‘user’ of technology” [Gasson 2003, 39]. For this reason, she proposes a human-centred approach, and offers a model with two cycles of the design process in order to to deal “separately but interactively” with system inquiry and implementation. During the first cycle — the cycle of inquiry – the stakeholders develop a shared understanding, as a result of the negotiation and development of a common worldview, which allows them to move to the second cycle–the implementation cycle–driven by the goals defined in the first cycle. The model emphases such a dialectical process, which should achieve a balance between the common worldview of stakeholders and the closure of predetermined, technical problems. 22

Balancing between the digital (technical problems) and the humanities (stakeholders) is a significant issue in DH discipline, and perhaps a major reason for renaming humanities computing into digital humanities. This renaming aimed to invite traditional humanities scholars to embrace digital technology, as DH was defined as “a part of the humanities not so much in the sense of being a distinct and separate discipline, but in the sense of interrelating deeply and multifariously with the humanities disciplines” [Svensson 2012]. It was not unusual for the humanities scholars to consider linking the humanities and computing as a paradox, trying to combine two incompatible worlds, which may feel “like chopping up a Picasso into a million pieces and feeding those pieces one by one into a machine that promises to put it all back together, cleaner and prettier than it looked before” [Barron 2010]. Gasson's model is based on a similar assumption of incompatibility between social and technological. She proposes that two incommensurable worlds of socio-cultural work and technology-interaction cannot be analysed with common methods, but we “should focus on separating problem investigation from solution design, using approaches and methods that permit us to operate in different modes of inquiry in each world” [Gasson 2003, 42]. 23

However, separating solution design (technological innovation) from problem investigation (research questions) would only further separate D–side from H–side of DH, making them two incommensurable worlds. Instead, this paper takes an Actor–Network Theory (ANT) approach that argues for methodological symmetry, advocating the use of the same type of analysis for all elements of network whether they are natural, technological, or social. A single explanatory framework should be used for interpretation of actors, both humans and non–humans [Callon 1986a]. By eliminating the human/non–human binaries, ANT enables technological artefacts to act, move and speak [Kim and Kaplan 2005, 169]. An actor is not just a point object or a placeholder [Latour 2005], but it is an association of heterogeneous elements, so that each actor is also a simplified network [Law 1992]. The concept of actor–network should be seen as a term that is 24 intentionally oxymoronic, embedding a tension between the centred actor and the decentred network–it “is thus a way of performing both an elision and a difference between what Anglophones distinguish by calling agency and structure” [Law 1999, 5]. While the hyphenated term made it difficult to see, the intention of ANT is not “to occupy a position in the agency/structure debate, not even to overcome this contradiction” [Latour 1999a, 16]. Instead, ANT's strategy is to ignore it, and focus on relations between the entities rather than their essences.

ANT approach offers an alternative to the user–centred approach in information studies by extending agency to non–humans, but also by focusing on relations between entities rather than on entities themselves. A constant shifting and positioning between two main tenets of the user–centred approach in IB field–centrality of the user and the essential role of context in information practices–has become a differentiation point for contemporary IB theories, but also the main difficulty in tracing information practices [Tabak 2014]. On the other hand, ANT ignores both human/technology and cognitive/social divide: the information practices are conceptualised as simultaneous and continuous circulation of exchanging properties between heterogenous entities. Such a conceptualisation offers a relational approach to information practices, in which information and users, individual and collective, humans and non-humans, cosmos and politics, constantly exchange properties. Information Cosmopolitics model [Tabak 2015] has been proposed to describe information practices as the circulation of hybrid entities continuously exchanging properties in the process of composing a collective, which is constantly redefined by the circulation. 25

A model for managing DH projects, proposed in the final section of this paper, will combine Information Cosmopolitics model with the iterative stages of Flexible Project Management [DeCarlo 2004], an agile project management framework. The model will provide the means for dealing with the issues that prevent some aspects of agile project management to be easily translated into DH research projects. In order to address issue of variability of schedule in academic environments, the model will modify definition of roles and work products of agile methods to be specific to DH research projects. The second issue in adopting agile methods to DH projects–unpredictability of progress–will be addressed by using evaluation as a constant process within the all iterative cycles of the model. This will enable a progressive and incremental cycle of creative uncertainties and their provisional closures, presented through the modular deliverables. Finally, in order to avoid technological instrumentalism and to balance the D–side and H–side of DH projects, Information Cosmopolitics, an ANT approach to information practices, will be employed to describe the processes in DH projects as a continuous circulation in which digital technology and the humanities constantly exchange properties. 26

Information Cosmopolitics

Information Cosmopolitics [Tabak 2015] is a model of information practices, based on actor-network theory [Latour 2005], and findings from a field study investigating information sharing practices in academic communities. The concept of cosmopolitics was borrowed from Isabelle Stengers [Stengers 2005] to describe a continuous circulation in which information and users, individual and collective, humans and non-humans, cosmos and politics, constantly exchange properties. A single cycle of the circulation consists of several consequential moments, necessary to perform the circulation (Figure 1). Each cycle starts with the perplexity brought by in-scription of collective propositions into information objects, which are then translated to individual propositions through the process of de-scription. The next moment of contextualisation provides an interface between individual and collective propositions. Finally, standardisation enables transformation of local de-scriptions into collectively accepted propositions, providing a closure to the initial perplexity. However, the closure is always provisional as each proposition is another invitation to perplexity. 27

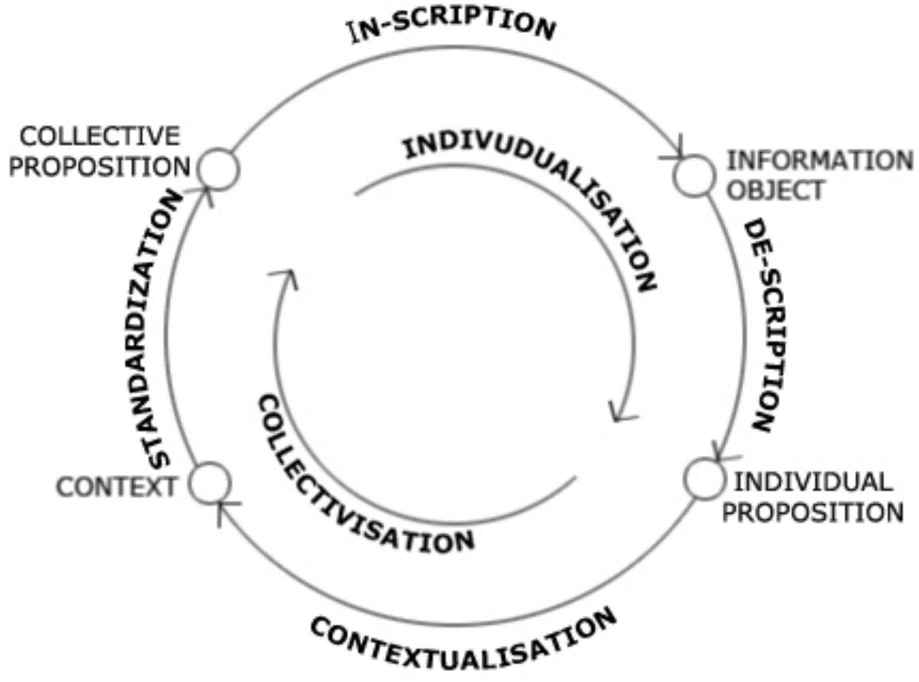

Figure 1. Information Cosmopolitics

In-scription

28 The circulation of information cosmopolitics starts with users' perplexity while encountering new information objects, which can be any object (human or nonhuman) that can carry information. Information objects are frequently placed in institutions such as libraries, archives, museums, and distributed through different media such as print, videos, and the Internet. But we often deal with them in our everyday life. For example, a speed bump on a street is an information object with an inscribed message from a police department which asks drivers to slow down [Latour 1987]. In-scription is a process when an entity "becomes materialized into a sign, an archive, a document, a piece of paper, a trace" [Latour 1999b, 306]. An information object is thus anything that leaves a trace, which could be retraced, translated, de-scribed. Such a trace often leaves an actor in a state of perplexity: Where has the trace come from? How can it be retraced? Can I translate it to my language? Is it relevant to me? How can I de-scribe this in-scription? This kind of perplexity is the starting point of information cosmopolitics. While information objects are in-scribed with implicit propositions for a generic individual, a specific individual has to translate the implicit propositions into explicitly relevant information in order to navigate the situation. Therefore, for a generic body to become a fully individualised actor dealing with a specific situation, a proposition in-scribed into an information object has to be de-scribed.

De-scription

29 Information is always transformation, a de-scription of information object. In order for inscription to be informing, actors need to translate information objects in their own terms. Actors approve or reject propositions in-scribed into information objects. They individualise collective propositions according to their own methodology, making them relevant or not. This individualisation transforms a user from a generic body to an actor. This is why the existence of a collective, an institution, or a cosmos is in the hand of local users [Latour 1987] – in the hand of local representations, as there is no reality without representation [Latour 2004b, 127]. In order to illustrate that individuals are neither in control, nor out of control, Latour uses the web term plug-in as a conceptual tool. He intentionally uses a web metaphor to illustrate that individuals can download a plug-in from an institution, which will provide them with a temporary competence for a specific situation [Latour 2005]. By downloading some cultural clichés, individuals are provided with competencies to create an opinion of a book, to have a political stance, or to know to which group they pertain. Someone is an individual

only if she or he has been individualised by downloading these “individualizers” from the outside. However, the outside is not a global context which determines the inside, but it is another local place to which an actor can be plugged-in. Plug-ins do not have the power to determine, they can simply make someone do something, and making do is not the same as causing, dominating, or limiting. De-scription is thus a process of "trans-forming" rather than “in-forming” an actor [Latour and Hermant 1998]. Rather than seeing actors as limited to their own description, their social or cognitive representation of the external world, information cosmopolitics approaches de-scription of an in-scription as rebuilding the cosmos. Through this transformation, an actor attributes to another actor “identity, interests, a role to play, a course of action to follow, and projects to carry out” [Callon 1986b, 24]. Such an approach enables us to see information processes as a result of complex negotiations, since “transfers of information never occur except through subtle and multiple transformations” [Latour 1999b, 298]. Actors change the role from intermediaries, simply transporting information, to mediators, that is “actors endowed with the capacity to translate what they transport, to redefine it, redeploy it, and also to betray it” [Latour 1993, 81]. They change from intermediaries that are shaped by heavy layers of context to mediators that shape the context, and “information changes from brick to clay, moved and shaped in unique ways by each perceiver” [Dervin 1983, 169]. However, meaning needs to be legitimised by others, so de-scriptions have to be put in a “big picture” to become comparable with the existing standards. Individual propositions have to be contextualised to become relevant for others.

Contextualisation

Contextualisation provides a common language for heterogeneous entities. Nothing links humans, non-humans, different domains of interests, different times and spaces, like a context. Linked by a common context, each of these entities depends on each other. They become combinable and commensurable. However, actors are not placed into this context, but attached to it. The context and its significance is created by actors. Individual interpretations are put in “big picture” in attempt to become significant for others. This frequently creates new information needs or changes priority of existing information needs, effecting circulation in consequent cycles. As the configuration of plug-ins is slightly changed by this prioritising, the individual identity is slightly changing too; thus the information is not de-scribed by the exactly the same individual, nor it is contextualised into exactly the same context, in every subsequent cycle. Therefore, the moment of contextualisation could be a departure point for all kinds of “serendipity” [Pettigrew 1999], and as such it is perhaps the most creative moment in the circulation providing the maximum possible combinations of propositions. Contextualisation allows “the maximisation of disputability” [Latour 2004a] so that a cosmos is not prematurely simplified into an agreement. Information cosmopolitics is not an attempt to put miraculously everyone in agreement [Stengers 2005]. It is neither about equality of actors, but it is about making them all “to be present in the mode that makes the decision as difficult as possible, that preludes any shortcut or simplification, any differentiation a priori between that which counts and that which does not” [Stengers 2005, 1003]. Contextualisation thus provides a space for a due process before decision of inclusion and exclusion. It provides actors to be commensurable by attaching them to the same context, before some information is excluded and other standardised. 30

Standardisation

Standardisation is a process through which information is articulated into a collective proposition. Provisional closure to the initial perplexity is provided by assigning an essence to the information through negotiations. What was an object of perplexity becomes the object of an agreement. Some information are standardised, some are excluded during these processes that produce the inside and the outside of the circulation through which a collective – a cosmos – is provided with a shape, size, and collective identity, represented through collective propositions. Once the propositions are standardised, the actors “no longer question their legitimate presence at the heart of collective life” [Latour 2004b, 109]. This move from context to standards is a move from a generic to a specific collective, a move from flexibility to certainty. While contextualisation provides diversity that enables commensurability, standardisation enables a hierarchy that provides unity. However, collective propositions can only gain stability in a constant circulation. In fact, they are recreated and translated in every single cycle of information cosmopolitics. While a dramatic change is always possible with a sudden introduction of an external in-scription, most changes are subtle but regular. Each cycle of the circulation is a new trial of credibility of a collective proposition. It is important to note that such circulation of negotiation does not 31 produce statements but propositions. To paraphrase Latour, there is a great difference between talking about information practices "if one uses propositions (which are articulate or inarticulate) instead of statements (which are true or false)" [Latour 2004b, 206]. The difference could not be sharper as statements refer to accuracy, and propositions are the result of articulation. While a quest for accuracy ignores negotiations, and as such paralyses circulation of information cosmopolitics, articulation is almost a synonym for cosmopolitics in which heterogeneous entities exchange properties in a constant negotiation. As such, collective propositions provide a local, thus manageable, totality, which in turn provides a closure for initial perplexity.

Invitation to Perplexity

However, such a closure is always provisional as through a new inscription new perplexity is invited. Figure 1 of the model does not illustrate the full integration of information into collective proposition, but rather only one cycle of this slow transformation. Each new piece of information goes through this process. If it is excluded before it is articulated into a proposition, it can still be in-scribed later when a new configuration of circulation is more favourable. Any excluded information is a new possibility in a new cycle, and every accepted proposition is regularly re-inscribed in a new information object in order to be as durable as possible. What was insignificant information in one cycle can be crucial information in another one. Information is not itself significant or insignificant, but it gains significant position in the circulation. It is translated in every moment of the circulation of information cosmopolitics. However, these transformations are not abrupt modifications but rather subtle changes, limited by the boundaries of the network. Actors could de-scribe an in-scription only within the limits of the in-scription and availability of plug-ins used for de-scription. Contextualisation is limited by already established standards in the network, and durability of propositions is dependent on the immutability and mobility of in-scriptions. As subtle as they are, these transformations still slightly modify the configuration of associations within the circulation. As a result, a new cycle brings a need for new plug-ins, which slightly changes individual identities, new priorities are created that slightly change context, and standards are slightly changed creating a need for new in-scriptions. All this creates new information needs and new perplexity, which feed the constant circulation of information cosmopolitics. 32

Diversity and Unity

By shifting the focus from individual mind and social context to this circulation, Information Cosmopolitics provides an alternative to the existing IB models. Knowledge is not seen as cognitive or social, but as distributed. The model suggests that individual and social are provisional effects of individualisation and collectivisation, and as such they cannot be used to explain information practices. Both individual and collective are merely moments of information cosmopolitics, connected by scripts that circulate between them. In one moment, the collective is scripted by the individual, only to script the individual next moment. These sequential events of exchanging properties between individual and collective is what keeps the circulation of information cosmopolitics, and what enables both diversity of individual associations and their unification into a common world — a cosmos, constantly redefined by the circulation. Following these events may illuminate how users translate information to be relevant to their interests, how they contextualise their individual propositions to evaluate them in order to be accepted by a collective, how the collective propositions are re-inscribed into information objects, and how the new digital inscription creates a new perplexity that triggers a new cycle of information practices. 33

A Hybrid Model for Managing DH Projects

The proposed hybrid model for managing DH projects is presented as a constant circulation of information cosmopolitics through the progressive stages of a DH project, borrowed from the Flexible Project Management framework. This section first describes the main elements of the model, adopted from agile management methods, which include roles of actors involved in the lifecycle of a DH project, the types of outcomes of these processes, and communication procedures. Then, the lifecycle of the proposed model is discussed. 34

The Main Elements of the Model

Project roles

An agile project is organised around self-managing, cross-functional teams, working on a functional increment, a feature, or a product in a short iteration time-box. In Scrum, major roles in projects are defined as Product Owner, Scrum Master, and team members. A Product Owner is an intermediary between customers and the team, who define and prioritise the backlog of tasks, help elaborate those tasks with the team, and accept the completed tasks. The product backlog is the list of all the tasks that need to be done in the project. The tasks are ranked and organised in smaller groups of tasks to be accomplished within a time-box, which is in Scrum defined as a sprint. The Product Owner is responsible for both product backlog and sprint backlog. Another role in Scrum is a Scrum Master, who is responsible for providing the team members with all the resources necessary to accomplish their tasks, to facilitate the team interactions and meetings, and to ensure regular progress. Finally, the team members are specialists working on the tasks. 35

In order to develop an agile methodology to support research management, Bezerra et al. introduce the ARDev model, in which they modify the Scrum roles into Research Starter, ARDev Master, and Research Team [Bezerra et al. 2014]. Research Starter, corresponding to the Product Owner in Scrum, is responsible for directing and monitoring research progress. While the Product Owner defines sprint backlog and prioritises user stories in a product backlog, Research Starter defines research items (research stories) into list of the research tasks per iteration and prioritises them in a research block list. ARDev Master manages the research process, keeps the team focused on the current task, and remove obstacle of research. Research Team is responsible for delivering research artefacts at the end of each research iteration. 36

While such a modification of work roles in a research project is more appropriate for DH projects than the proposed Scrum work roles, DH projects are more complex than a traditional humanities research project. Burdick et al. point out that DH projects involve multiple strata of personnel from a principal investigator to project advisors, IT staff, the humanities scholars, and students. As such, they all “have technical, administrative, and intellectual aspects to their production” [Burdick et al. 2012, 132]. The hybrid model proposes that any DH project must include three basic roles: a role that corresponds to the digital aspects of DH, another one related to the humanities aspects of DH, and a role that combine both digital and humanistic aspects of a DH project. For the purpose of the model, these core roles in DH projects are defined as: 37

- D-role,
- H-role, and
- DH-role.

The roles are based on the core competencies in processes and methods required by DH projects [Burdick et al. 2012]. The D-role deals with the technical aspects of DH projects, which include specific competencies for doing DH work such as web development, infrastructure, server environment, interface design, choices about tools, platforms, software, and hardware. The H-role provides the content, the most visible intellectual aspect of DH projects. This role enables the fundamental humanistic values and perspectives to take a central part of a DH project. Without the D-role, a DH project might not fully benefit from digital technology; without the H-role, there is a danger to neglect the main target group of DH project — the humanities scholars. The major responsibility of the DH-role is to provide the balance between digital (technical) and humanistic (intellectual) aspects of DH projects. While H-role provides the content for a project, Burdick et al. point out that DH projects “present arguments and knowledge experiments in many different ways, often contributing to the creation of new knowledge through complex interactions, visualizations, data and data structures, and even code” [Burdick et al. 2012, 133]. They propose that core intellectual competencies for a DH projects are cross-cultural communication, generative imagination, and iterative and lateral thinking. The DH-role should be able to align these competencies with both digital and humanistic aspects of DH projects. 38

The focus of different DH projects can vary from designing a simple digital repository to a complex modes of distance reading and cultural analytics. Different projects will require different competencies. The projects also come in different sizes. They can be as large as international and multidisciplinary projects involving a number of institutions, but they can be as small as a project involving a single researcher, working on a specific issue. However, they all need to involve the three core roles of DH projects. A project member can play more than a single role, and in the smallest projects, a single 39 member will be required to play all three roles. The name of these roles is not important as long as they refer to digital, humanistic, and DH aspects of DH projects. Similarly, while it is possible to use the terms Research Starter or Product Owner, depending of the project focus, this model will rather use the term Principal Investigator (PI) as it includes other than merely intellectual (research) or merely technological (digital) aspects of a DH project. This also implies that a DH-role is the most appropriate to take the role of PI because the PI role deals not only with administrative but also with technical and intellectual aspects of the project. The responsibility of the PI in a DH project is “to organize the project team, establish timelines for deliverables, and assess the project at each stage of development” [Burdick et al. 2012, 124].

Communication procedures

In order to organise the project team and assess the project progress, the PI needs to establish appropriate communication procedures. This model proposes the following ceremonies and tools for communication between the actors in DH projects, adapted from Scrum: 40

- Project backlog,
- Milestone backlog,
- Regular meeting,
- Milestone planning meeting, and
- Milestone retrospective meeting.

In Scrum, the role of the Product Owner is to collate the stakeholders' inputs into a “backlog” of requirements. The backlog consists of all the project tasks, ranked in priority order. A sub-set of the product backlog is taken by the team to be completed within a short duration milestone, called a sprint. This sub-set of the product backlog that the team commits to complete within a sprint is called the Sprint backlog. This model will use the terms project backlog and milestone backlog to refer to similar project management tools because it will not use a sprint for time-boxing nor a DH project is not necessary a product-oriented project. The project backlog consists of all the project milestones, while the milestone backlog refers to the tasks to which the team is committed while working on a milestone. 41

In order to facilitate team communication, Scrum employs a series of meetings, known as ceremonies, including sprint planning, daily scrum, sprint review, and sprint retrospective. Sprint planning refers to a comprehensive meeting that defines what needs to be done by the end of the sprint. The daily scrum is a short meeting of no more than 15 minutes, in which each member of the team provides a very short answers to the questions: What I did yesterday? What I plan to do today? What is in my way (or what is blocking my progress)? The Sprint review provides a feedback on a demo of the working software that the team has built, which helps the team validate that they are on the right track. The sprint retrospective enable the team to reflect upon the previous sprint in order to learn some lessons and then adjusts their work or the environment so that they improve their work. This model uses the regular meetings instead of daily scrum meetings in order to accommodate to different work schedules of team members in a DH project, while the milestone planning and milestone retrospective correspond to the sprint planning and sprint retrospective. The milestones are defined in the project plan, and they are related to the project outcomes. 42

Project outcomes

The model defines four types of project outcomes: 43

- vision,
- plan,
- prototype, and
- deliverable.

They corresponds to four phases in the proposed lifecycle of a project (visionate, speculate, innovate, and disseminate), which are discussed in the next section. A vision type of project outcome is a departure point for a DH project. The main outcome of this stage is the project vision document, which defines aims, objectives, the main stakeholders, and 44 expected benefits of the project. The vision document may also (although roughly) estimate timing, human resources and budget. The main outcome of the speculate-stage is the project plan, which defines the project milestones, estimates required resources to accomplish each milestone, identifies project risks, and present the contingency plan. In this stage, the project plan does not include the tasks necessary to accomplish a milestone, which will be added during milestone planning meetings within the innovate-stage. The main outcome of the innovate-stage is a prototype. The term prototype is used here rather in an engineering sense that refers to a first instance of a product to be tested (and refined if necessary) just before the final product is launched. In this model, a prototype refers to an instance of a project deliverable that needs to be tested (and refined if necessary) before it is disseminated. This might be a fully functional prototype of a digital tool or a draft of an academic manuscript that needs some editing. After testing and evaluation, a prototype is translated into a deliverable, which is a major outcome of the disseminate stage.

Project Lifecycle

An adopted cycle of Information Cosmopolitics model and stages of the Flexible Project Management framework are combined into a hybrid model for managing DH projects. The model proposes that the lifecycle of a DH project consists of a constant circulation of information cosmopolitics, in which an iteration starts with a perplexity and ends with a provisional closure, in-scribed into a work outcome. A progressive flow of a DH project is adopted from the Flexible Project Management framework, which identifies five steps in an agile project: Visionate, Speculate, Innovate, Reevaluate, and Disseminate. As the processes of evaluation are already incorporated within the circulation of information cosmopolitics, the proposed model defines the lifecycle of a DH project to be a progressive flow in four stages: 45

- Visionate,
- Speculate,
- Innovate, and
- Disseminate.

The process of evaluation is considered as a constant activity rather than a stage in the project lifecycle. The individual evaluation refers to de-scription process in information cosmopolitics cycle, while a collective evaluation is related to the standardisation process. The four progressive stages are defined by progressive work outcomes, conducted in four progressive work contexts: visionate-context, speculate-context, innovate-context, and disseminate-context. The visionate-stage results in the project vision; the speculate-stage in the project plan; the innovate-stage results in a prototype; while a deliverable is a work outcome of disseminate-stage. Figure 2 illustrates the lifecycle of a DH project, according to the hybrid model. 46

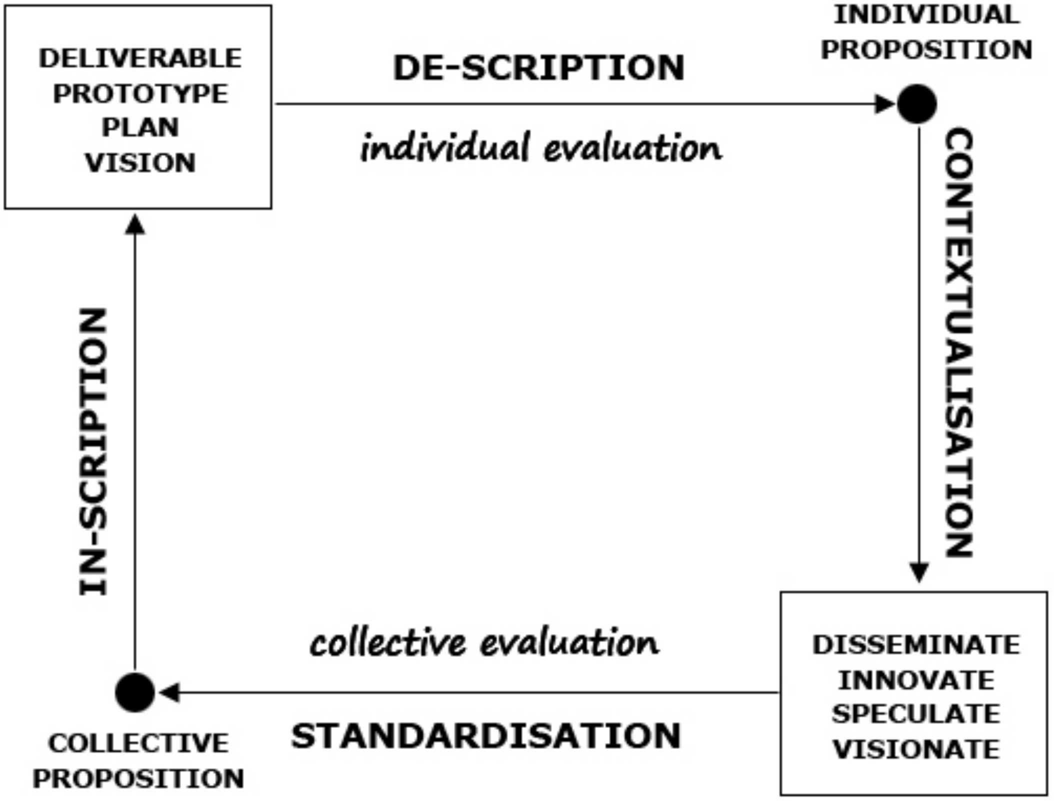

Figure 2. Hybrid model for managing DH projects

Each cycle starts with encountering a work outcome by a team member. First, the team members de-scribe an existing work outcome (vision, plan, prototype, or deliverable), according to their own interests and relevance criteria, which results in their individual propositions. In order to make an individual proposition relevant to other actors in the project, a team member has to place this proposition in a work context, corresponding to the four work stages of the project. Work contexts include the work on visioning, speculating, innovating, and disseminating. Once the individual proposition is in a particular context, it becomes commensurable with other propositions within the project. As such, the individual proposition is a subject to standardisation, that is a collective evaluation. Some propositions are accepted and some are rejected during this process. The accepted propositions are transformed into collective propositions, and they are finally in-scribed into a work outcome, normally into a work outcome corresponding to the context to which the proposition has been attached in the current cycle. 47

In a linear progressive flow, this circulation through a context and its corresponding work outcome is repeated as long as a milestone is not reached. A milestone refers to in-scription of the final collective proposition within a context, which allows the project to move to the next context. It means that when the final collective proposition within the visionate-context is in-scribed into the project vision document, a milestone is reached, which enables the project to move to the speculate-context. Reaching the milestone within the speculate-context will move the project to innovate-context, while a milestone within innovate-context will enable dissemination of a deliverable. However, the progress of a project is not necessary a linear process. For example, collective evaluation of an individual proposition attached to the innovate-context can result in reconsidering the project plan, or even the project vision, which may require a complex negotiation within the project. The more a proposition deviates from the linear flow of the project, the more difficult is to in-scribe it into a project outcome. And a deviation from a linear progressive flow of a DH project can occur at any moment, as we will see in the following discussion of stages in the project lifecycle. 48

A project idea

The very first iteration of the project lifecycle can start with an idea occurring merely in an individual mind. The 49 relevance of this idea is evaluated by the individual, and then attached to any context (visionate, speculate, innovate, or disseminate). In this stage, anything can be a trigger for developing a project, from a general aim to make a knowledge contribution to a wish to deliver an outcome for personal benefits. However, in order to translate an idea into a project vision, the idea has to be collectively evaluated, even if it means virtually in an individual mind. At this stage, this "virtual" collective can be anything from an academic specialty to the general public. Through the process of standardisation, an individual compares the idea with standards of this collective to evaluate its possible value for others before the idea is in-scribed into a rough draft of a project vision. This first instance of a project vision, which may be as short as a sentence, is a starting point for a number of iterations through visionate-context, leading to a formal vision document.

A project vision

The initial idea for a DH project commonly comes from an individual who can take a DH-role in a project, or someone who may play the role of PI in the project. However, this idea may come from an H-role, proposing a traditional humanistic topic, or a D-role, offering a simple computer code for text manipulation without a clear vision of possible benefits for the humanities. In any case, for a research project to become a DH project, it is crucial to involve both digital and humanistic aspects of a DH project from its very inception. The idea in-scribed into the first instance of a project vision is presented to at least the three main roles in a DH project (D-role, H-role, and DH-role) through a set of iterations. Each iteration starts with de-scription of the initial in-scription of the project vision. An individual proposition is then placed in the visionate-context, to be collectively evaluated and translated into a collective proposition. Finally, this collective proposition is in-scribed into a vision document. The formal vision document will define at least its most important elements: the aim of the project, its objectives, the expected deliverables and benefits, the main stakeholders, technical and human requirements, and estimated timing and budgets. Definition of each element requires a minimum of three iterations, each iteration addressing one of the three main roles in a DH project. 50

For instance, an idea can come from a D-role, proposing a computer code manipulating Google maps. In the first iteration, a DH-role may de-scribe this idea into a possible delivery of the project in the form of "deep maps." The vision of this deliverable may be de-scribed by a H-role in the next iteration into an objective of the project to map spreading a cultural concept through time and space. In the third iteration, a D-role may de-scribe deep mapping of a cultural concept as a possibility to use specific media in the project, which will be in-scribed into a new objective, deliverable, or into new estimation of requirements and budget in the project vision document. 51

There is an infinite number of possible outcomes of such iterations, which is not possible to anticipate by a model. It is up to a PI and other project stakeholders to make a decision when it is time to scope the project and to draft the formal project vision document. When such a decision is made, the PI starts a new set of iterations to define the scope of project. This process also requires the minimum three iterations for each main role, so that both digital and humanistic aspects of a DH project are considered. The final outcome of this process — the project vision document — is the first milestone in the project. As a milestone is defined as a final collective proposition within a context, the final agreement about the project vision document within this stage enables the project to move to the speculate-stage. 52

Speculating a project plan

The main outcome of the speculate stage is the project plan. While the name of the main outcome of this stage suggests an order in this process, the naming of the process as speculation suggests flexibility and adaptability of the process. Failure to plan and schedule an agile project can send the project into chaos, while over-planning can make team members prisoners of the plan, disabling the project "from initiating or effectively responding to change" [DeCarlo 2004, 296]. The aim of the speculate stage in this model is to identify the project milestones rather than planning the detailed tasks. While in traditional approaches, planning is seen as a distinct project stage with a large up-front effort, the speculate stage in this hybrid model is defined as "a moderate up-front effort followed by lower level but constant updates" [Chin 2004, 107] in a continuous process of planning throughout the entire project. The detailed planning will occurs at the very beginning of the work on each milestone during the innovate-stage. 53

The speculate stage is comprised of several sets of iterations that define the main elements of the project plan. Each 54 iteration involves de-scription of the element by individual team members. They attach their individual propositions to the speculate-context, which enables their comparison with the standards defined in the vision document. An individual proposition is then translated into a collective proposition through the process of standardisation, or it is rejected as a result of this collective evaluation. When an element of the project plan is defined, the speculation moves to another set of iterations, defining another element. The project plan should define at minimum the following elements:

- the project milestones,
- the milestone owners,
- required resources,
- the project risks, and
- a contingency plan.

55 The first set of iterations identifies the milestones of the projects. The next step is to assign ownership to the milestones. In other sets of iterations, PI and milestone owners estimate the human, technical, and financial recourses required to accomplish a milestone, identify the project risks, and make a contingency plan. The contingency plan should focus on high-level anticipation of possible project pathways since contingency planning in agile approach looks on alternative paths of the project, while in the traditional approach it looks "for the shortest path around an obstacle" [Chin 2004, 129]. At the end of this stage, the project plan is completed, and the project moves to the innovate-stage.

Innovating

56 The innovate-stage involves the work on each milestone of the project. During the speculate-stage, the milestones are prioritised in the project backlog, and each milestone is assigned to a milestone owner — with the resources necessary to accomplish it. However, this is all done on high level speculation, without detailed planning. During the innovate-stage, each milestone is defined in detail. The work on a milestone starts with a milestone planning meeting, which identifies the tasks necessary to accomplish the milestone. The tasks are in-scribed into the milestone backlog, which will be continuously prioritised by the milestone owner. The next step is the actual work on a task, involving intense collaboration and feedback. Each iteration is based on "trial and error" practices that eventually result in a form of a prototype. Such experimental approach provides the innovation process with possibility to generate early values or fast failures. This prevents the project to be stacked at a dead end. Accepting a failure as something other than defeat "is part of the life cycle of innovation" [Burdick et al. 2012, 22]. The final step in this process is a milestone retrospective meeting, which confirms that a prototype is ready for dissemination.

57 A milestone planning meeting is initiated by a milestone owner, who will give an overview of the milestone, and its relations to the other project milestones and to the project as a whole. The first step is to identify the tasks required to complete the milestone. Each task will be assigned with the task owner, required resources, and time to accomplish the task. However, this is not done by estimating resource allocation and duration along a single sequence of activities. Unlike in traditional project management, an agile approach to planning is based on commitment. Chin points out that planning is tricky in technical and creative environments, such as those of DH projects [Chin 2004]. The actors in such environments need room to experiment with different ideas, so asking them for commitment rather than allocating the tasks with top-down estimate of resources and timing can create a win-win situation. In such situations, the team members will see a planning process as something that encourages creativity and collaboration. As commitments depend on each other, "the whole team needs to work together and become engaged for this process to work" [Chin 2004, 104]. When milestone owners are committed to their milestone, and task owners are committed to their task, the work on specific tasks can start.

58 The work on a specific task is the most basic element of the innovate stage. The task is de-scribed by a task owner, who develops a feature of a prototype within the innovate-context. This is then evaluated and tested according to the standards defined in the milestone meeting and presented in the milestone backlog. If the feature does not meet those standards, the next iteration is performed. If it meets the standards, the feature is accepted by the milestone owner, who updates the task as completed in the milestone backlog. In a case when something goes wrong with the execution of this particular task, the milestone owner and the task owner consider whether there is a need for a different path in

completing the task, or whether there is a need to modify a standard in the milestone backlog.

59 This can lead to downgrading the standards of the project, but also to an “opportunistic translation,” defined as “a form of information behaviour that occurs when people use information, information systems or parts of information systems for other, often unexpected, purposes than their original purpose” [Tabak 2010, 47]. An opportunistic translation can also occur in a normal flow of an innovate iteration when it results in a feature of the milestone that completely changes the perspective on a proposed prototype, a deliverable, or the project as whole. Such an opportunistic translation should be attached back into the speculate-context or the visionate-context as it has a potential to change the project plan and even the project vision.

60 The final step of the work on a milestone is a milestone retrospective meeting, which is conducted when all tasks from the milestone backlog are completed, and when the milestone owner decides that a prototype is ready for dissemination. The milestone retrospective meeting is used not only to confirm the milestone accomplishments, but also for generating creative ideas for further work on the project. The first part of the meeting is dedicated to confirmation that all tasks have been completed and that a prototype is ready for dissemination. The significance of the completed milestone is discussed in relations to the other project milestones and to the project as a whole. This discussion also addresses possible archiving needs. Brown et al. argue that archiving in DH project is not only a requirement of many funding bodies, but it is also “a marker of some kind of doneness” [Brown et al. 2009], a provisional closure for initial perplexity. The next step is to discuss the lessons learned during the process. Finally, the new ideas, generated during the work on the milestone, are discussed. The particular attention is paid to possible opportunistic translations, identified during the work on this milestone. If necessary, a visionate-iteration and/or a speculate-iteration will be performed, and the project plan and vision will be updated by PI. While traditional project management focuses on conformance to the original plan, an agile project management “benchmarks against future possibilities based on the latest results and events” [DeCarlo 2004, 367].

Disseminating deliverables

61 The final stage of the project lifecycle is dissemination of the project deliverables. For a prototype to become a deliverable, it needs to be tested, refined, and evaluated. Finally, the procedures for maintaining deliverables should be drafted. A prototype is first tested against technical requirements and refined according to the test. It is then evaluated by each core role in a DH project. This process is thus composed of the minimum three iterations, in which at least a D-role, a H-role, and a DH-role de-scribes the prototype, places it to the disseminate-context in order to be evaluated by the dissemination standards. If it conforms to the standard, it is in-scribed into a deliverable; otherwise, it is moved back to the innovate-stage. It is possible that, even in this stage, a proposition for modification of the project plan or the project vision is generated. While it is not likely to modify the project vision at this stage, such a proposition should be seriously considered. It may be used for defining further studies in the final project report, or as an idea for a new project. However, when a prototype is translated into a deliverable ready for dissemination, the most important decision to be made is to identify the future life of the deliverable.

62 A fluidity of digital media and text in DH projects stands in contrast to stabilising practices of printing media, which “enables iterative work to hitherto unprecedented degrees, introducing the software term 'version' into units of scholarly production” [Burdick et al. 2012, 21]. This makes maintenance of deliverables in DH projects significantly more challenging than in traditional projects in the humanities. Deliverables in DH projects are often expected to grow into incremental versions, as they are “both multiple and modular” [Brown et al. 2009]. Such a modular design facilitates lateral project development, which enables “building new and more robust projects on the shoulders of previously completed work” [Slumkoski 2012]. Moreover, the future use of deliverables is likely to produce different opportunistic translations, which may provide the most creative ideas for improvement of a deliverable, or even for new projects. The maintenance procedures should consider all features of deliverables in DH projects. They are modular and incremental, they can be used concurrently and laterally in other projects, and they may be opportunistically translated by their users. Therefore, a maintenance plan, which is the final step of dissemination stage, in a DH project should answer not only questions regarding preservation strategies (e.g. Where will the project outcomes be hosted and maintained? How will the deliverables be sustained?), but also questions regarding generative features of DH projects, such as: What are the

incremental possibilities of a project deliverable? How can it be used concurrently and laterally in other projects? Can a completed deliverable be used as a departure point for another project? How will use of a deliverable be monitored? What will be a procedure to capture and utilise an opportunistic translation of a deliverable?

Closing a project

The final step of the project is to close the project. When all deliverables have been disseminated, PI will prepare the final report of the project. The form of this report will depend of the context in which project has been conducted, but it is most likely to include the following sections: an executive summary, a description of the project trajectory in relation to the project objectives, a presentation of deliverables, a discussion on implications of the project, and a maintenance/sustainability plan. An executive summary will summarise the aim, results, and implication of the project, and identify the main stakeholders. The next section will provide the trajectory of the project in relation to each objective of the project, and it will note any change to the original objectives. A presentation of deliverables will provide description of each deliverable, and discuss their impact, while discussion of implications of the project will present the significance of the project, and its practical and theoretical implications. The project maintenance/sustainability plan will define the project preservation strategy and identify possible generative and incremental features that can be used in other projects, or to be a trigger for a new project. The report will be evaluated on the project closing meeting. If there is any proposition for modifications, the report will be modified during the meeting. In a case of a proposition that requires significant modifications of the final report, a new meeting will be arranged. Otherwise, at the end of the project closing meeting, PI will close the project. 63

Conclusions

Managing complex interplay between digital and humanistic aspects of DH projects requires consideration of insights and methods from different disciplines. This article combined the insights from the humanities and information studies with methods in software development in order to develop a hybrid model for managing DH project. The hybrid model is based on an existing model of scholarly information practices, and adopts its main elements from agile project management methods. According to the model, the main roles involved in the lifecycle of a DH project are D-role, H-role, and DH-role. D-role corresponds to the digital aspects of DH, H-role is related to the humanities aspects of DH, and DH-role combines both digital and humanistic aspects of a DH project. A project member can play more than a single role, and in the smallest projects, a single member will be required to play all three roles. The model employs regular meetings, milestone planning meetings, and milestone retrospective meetings as its main communication ceremonies. The four types of the project outcomes are: vision, plan, prototype, and deliverable. They corresponds to four stages in the proposed lifecycle of a project (visionate, speculate, innovate, and disseminate). The lifecycle of DH projects consists of a constant circulation of information cosmopolitics, in which each iteration transforms an initial perplexity into a provisional closure. However, such a closure is always provisional as through a new inscription new perplexity is invited, which feeds the circulation of information cosmopolitics. 64

The proposed model is flexible. The name of work roles, work outcomes, communication ceremonies, or the lifecycle stages are easy to be modified to adopt to different situations. However, it is crucial to keep the features and requirements of the model that address the complexity of DH projects. For instance, it is important to involve the work roles related to both digital and humanistic aspects of a DH project. The communication procedures have to generate feedback and include some retrospective ceremonies. The aim of these procedures is constant learning. The work outcomes in DH projects are modular, incremental, and lateral. The stages in the lifecycle of the project should correspond to these features of the work outcomes. The evaluation is not a separate stage in the lifecycle of a DH project, but it is rather a part of constant circulation of the project iterations. It includes both individual and collective evaluation in order to provide a DH project with both diversity of individual associations and their unification into a collective agreement. The model should thus enable a progressive and incremental lifecycle of creative uncertainties and their provisional closures, presented as a continuous circulation in which the digital and the humanities constantly exchange properties. 65

Acknowledgement

The research leading to this publication has received funding from the European Union Seventh Framework Programme (FP7) under grant agreement n° PCIG14-GA-2013-630732.

Works Cited

Abbas et al. 2008 Abbas, N., Gravell, A. M. and Wills, G. B. “Historical roots of Agile methods: where did ‘Agile thinking’ come from?” *XP2008: Proceedings of 9th International Conference on Agile Processes in Software Engineering and Extreme Programming*, Limerick, Ireland, June 2008.

Barron 2010 Barron, P. 2010. “Putting the ‘Humanities’ in ‘Digital Humanities’”. Available at: http://www.insidehighered.com/views/2010/11/04/barron [Accessed December 16, 2015].

Beck 1999 Beck, K. *XP Explained*. Addison-Wesley Professional, Boston, MA (1999).

Beck et al. 2001 Beck, K., et al. 2001. *The Agile Manifesto.* Available at: ttp://www.agilemanifesto.org [Accessed January 16, 2016].

Bezerra et al. 2014 Bezerra, D. R., Dias-Neto, A. C. and da Silva Barreto, R. “ARDev: a methodology based on scrum principles to support research management on software technologies,” *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, Riverton, NJ, November 2014.

Bhalerao et al. 2009 Bhalerao, S., Puntambekar, D. and Ingle, M. “Generalizing Agile Software Development Life Cycle,” *International Journal on Computer Science and Engineering*, 1(3): 222-26.

Brown et al. 2009 Brown. S., Clements, P., Grundy, I., Ruecker, S., Antoniuk, J., and Balazs, S. “Published Yet Never Done: The Tension Between Projection and Completion in Digital Humanities Research”, *Digital Humanities Quarterly*, 3(2). Available at: http://www.digitalhumanities.org/dhq/vol/3/2/000040/000040.html.

Burdick et al. 2012 Burdick, A., Drucker, J., Lunenfeld, P., Presner, T., and Schnapp, J. *Digital_Humanities*. MIT Press, Cambridge, MA (2012).

Callon 1986a Callon, M. “Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of Saint Brieuc Bay.” In I. J. Law (ed), *Power, Action and Belief: A New Sociology of Knowledge?*, London (1986a), pp. 196-232.

Callon 1986b Callon, M. “The Sociology of an Actor-Network: The Case of the Electric Vechile.” In I. J. Law and A. Rip (eds), *Mapping the Dynamics of Science and Technology: Sociology of Science in the Real World*, London (198ba), pp. 19-34.

Chin 2004 Chin, G. *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. American Management Association, New York (2004).

Coad et al. 1999 Coad, P., Luca, J. D. and Lefebvre, E. *Java Modeling Color with Uml: Enterprise Components and Process with Cdrom*. Prentice Hall, Englewood Cliffs, NJ (1999).

Cohen et al. 2004 Cohen, D., Lindvall, M. and Costa, P. “An introduction to agile methods.” In M. V. Zelkowitz (ed), *Advances in Computers, Advances in Software Engineering*, Amsterdam (2004).

DeCarlo 2004 DeCarlo, D. *Extreme Project Management: Using Leadership, Principles, and Tools to Deliver Value in the Face of Volatility*. Jossey-Bass, San Francisco (2004).

Dervin 1983 Dervin, B. “Information as user construct: the relevance of perceived information needs to synthesis and interpretation.” In S. A. Ward and L. J. Reed (eds), *Knowledge Structure and Use: Implications for Synthesis and Interpretation*, Philadelphia (1983), pp. 153-83.

Dyba et al. 2014 Dyba, T., Dingsoyr, T. and Moe, N. B. “Agile Project Management.” In G. Ruhe & C. Wohlin (eds), *Software Project Management in a Changing World*, Berlin (2014), pp. 277-300.

Gasson 2003 Gasson, S. “Human-centered vs. user-centered approaches to information system design,” *Journal of Information Technology Theory and Application*, 5(2): 29-46.

Hicks and Foster 2010 Hicks, M., & Foster, J. S. “Score: Agile research group management,” *Communications of the ACM*, 53(10): 30-31.

Highsmith 2000 Highsmith, J. *Adaptive software development: A collaborative approach to managing complex systems*. Dorset House, New York (2000).

Highsmith 2004 Highsmith, J. *Agile Project Management*. Addison-Wesley, Boston, MA (2004).

Kemman and Kleppe 2015 Kemman, M. and Kleppe, M. “User Required? On the Value of User Research in the Digital Humanities.” In J. Odijk (ed), *Selected Papers from the CLARIN 2014 Conference*, Linköping, Sweden (2015), pp. 63–74.

Kim and Kaplan 2005 Kim, R. M. and Kaplan, S. M. “Co-Evolution in Information Systems Engagement: exploration, ambiguity and the emergence of order.” In P. J. Ågerfalk, L. Bannon and B. Fitzgerald (eds), *Proceedings of the 3rd International Conference on Action in Language, Organisations and Information Systems*, Limerick, Ireland (2005), pp. 166-80.

Latour 1987 Latour, B. *Science in Action: How to Follow Scientists and Engineers Through Society*. Harvard University Press, Cambridge, MA (1987).

Latour 1993 Latour, B. *We Have Never Been Modern*. Harvard University Press, Cambridge, MA (1993).

Latour 1999a Latour, B. “On Recalling ANT.” In J. Law & J. Hassard (eds), *Actor Network Theory and After*, Malden, MA (1999a), pp. 14-25.

Latour 1999b Latour, B. *Pandora's Hope: Essays on the Reality of Science Studies*. Harvard University Press, Cambridge, MA (1999b).

Latour 2004a Latour, B. *Score: How to Talk About the Body? The Normative Dimension of Science Studies*, Body & Society, 10(2-3): 205-29.

Latour 2004b Latour, B. *Politics of Nature: How to Bring the Sciences Into Democracy*. Harvard University Press, Cambridge, MA (2004b).

Latour 2005 Latour, B. *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford University Press, Oxford (2005).

Latour and Hermant 1998 Latour, B. and Hermant, E. 1998. “Paris: Invisible City.” Available at: http://www.bruno-latour.fr/virtual/EN/index.html [Accessed January 14, 2016].

Law 1992 Law, J. “Notes on the Theory of the Actor Network: Ordering, Strategy and Heterogeneity,” *Systemic Practice and Action Research*, 5(4): 379-93.

Law 1999 Law, J. “After ANT: Topology, Naming and Complexity.” In J. Law & J. Hassard (eds), *Actor Network Theory and After*, Malden, MA (1999), pp. 1-14.

McPherson 2009 McPherson, T. “Introduction: Media Studies and the Digital Humanities”, *Cinema Journal*, 48(2): 119-23.

Nerur and Balijepally 2007 Nerur, S. and Balijepally, V. “Theoretical Reflections on Agile Development Methodologies,” *Communications of the ACM*, 50(3): 79-83.

Pettigrew 1999 Pettigrew, K. E. “Waiting for chiropody: Contextual results from an ethnographic study of the information behaviour among attendees at community clinics,” *Information Processing & Management*, 35(6): 801-17.

Reed 2014 Reed. A. “Managing an Established Digital Humanities Project: Principles and Practices from the Twentieth Year of the William Blake Archive,” *Digital Humanities Quarterly*, 8(1). Available at: http://www.digitalhumanities.org/dhq/vol/8/1/000174/000174.html.

Schwaber 2004 Schwaber, K. *Agile Project Management with Scrum*. Microsoft Press, Redmond, WA (2004).

Slumkoski 2012 Slumkoski, C. “Modular Design, Lateral Project Development, and the Sharing of Work: Lessons from the Edward Winslow Family Papers and the Atlantic Canada Virtual Archives”, *Digital Studies*, 3(1).

Stare 2013 Stare, A. “Agile project management – a future approach to the management of projects?,” *Dynamic Relationships Management Journal*, 2(1).

Stengers 2005 Stengers, I. “The Cosmopolitical Proposal.” In B. Latour & P. Weibel (eds), *Making things public*, Cambridge, MA (1999), pp. 994-1003.

Svensson 2009 Svensson, P. “Humanities Computing as Digital Humanities,” *Digital Humanities Quarterly*, 3(3). Available at: http://www.digitalhumanities.org/dhq/vol/3/3/000065/000065.html

Svensson 2010 Svensson, P. “The Landscape of Digital Humanities,” *Digital Humanities Quarterly*, 4(1). Available at: http://www.digitalhumanities.org/dhq/vol/4/1/000080/000080.html

Svensson 2012 Svensson, P. “ Envisioning the Digital Humanities,” *Digital Humanities Quarterly*, 6(1). Available at: http://www.digitalhumanities.org/dhq/vol/6/1/000112/000112.html

Tabak 2008 Tabak, E. *Non-purposive information behaviour in organisational intranets*. Institute for Information Management, Canberra (2008).

Tabak 2010 Tabak, E. “Opportunistic Translation in Development and Management of an Organisational Intranet.” In A. M. Rawani, & H. Kattani (eds), *Proceedings of 2010 International Conference on Innovation, Management and Service*, Liverpool, UK (2010), pp. 47–52.

Tabak 2014 Tabak, E. “Jumping between context and users: A difficulty in tracing information practices,” *Journal of the Association for Information Science and Technology*, 65(11): 2223–32.

Tabak 2015 Tabak, E. *Information Cosmopolitics: An Actor-Network Theory Approach to Information Practices*. Elsevier-Chandos, Oxford (2015).

van Zundert 2012 van Zundert, J. “If You Build It, Will We Come? Large Scale Digital Infrastructures as a Dead End for Digital Humanities,” *Historical Social Research / Historische Sozialforschung*, 37(3): 165–86.

Warwick 2012 Warwick, C. “Studying users in digital humanities.” In C. Warwick, M. Terras, & J. Nyhan (eds), *Digital Humanities in Practice*, London (2012), pp. 1–21.

Way et al. 2009 Way, T., Chandrasekhar, S. and Murthy, A. “The Agile Research Penultimatum.” In H. Arabnia & H. Reza (eds), *Software Engineering Research and Practice*, Las Vegas (2009), pp. 530-36.

This work is licensed under a Creative Commons Attribution-NoDerivatives 4.0 International License.