

Using an Agile Approach in a Large, Traditional Organization

Dot Tudor
TCC
Sandbach, Cheshire
dottudor@tcc-net.com

George A. Walter
OCLC Online Computer Library Center, Inc.
Dublin, Ohio
walter@oclc.org

Abstract

Can Agile approaches be used successfully in large organizations, where traditional methods and high levels of governance are the norm?

Although the iterative, agile approaches have been seen to work well in small, flexible organizations, or on smaller projects, they frequently fall foul of the larger organization's need for governance, investment appraisal and control.

Formed in 1967, OCLC develops software for use by libraries and their users, museums, and academic institutions. Researchers, students, faculty, scholars, professional librarians and other information seekers use OCLC services to obtain bibliographic, abstract and full-text information. OCLC aims to be the leading global library cooperative. More than 54,000 libraries in 96 countries and territories around the world use OCLC services to locate, acquire, catalog, lend and preserve library materials.

This paper examines how TCC, a training and consultancy company from Cheshire, England has worked with OCLC, the Online Computer Library Center based in Dublin, Ohio to incorporate the Dynamic Systems Development Method (DSDM) into a development culture that was deeply-rooted in "traditional" software development methods. Examples from multiple projects illustrate how the adoption of DSDM helped OCLC change its culture and achieve success in software development and deployment. OCLC's TLC dashboard was used to track the effectiveness of the development cycle, and to collect metrics from 2003 to the present. We discuss some of the challenges we faced and the six agile steps to success.

1. Introduction

In 1998, The Online Computer Library Center (OCLC), a large, nonprofit company based in Dublin, Ohio, was experiencing problems with its software delivery. We realized time-to-market was becoming increasingly important to our customers and began to really take a hard look at our software development practices.

We identified that projects were typically 18 – 24 months in duration and products were often obsolete even before they were released. Teams were trapped into developing has-been software and then trying to bring it up to speed through maintenance releases. Follow me now on my six steps to Agile success.

The six Agile Steps to success

- Step 1: OCLC Meets DSDM
- Step 2: Does it really work that way?
- Step 3: Project Set-up and the "rolling workshop"
- Step 4: We also need to tailor to our processes
- Step 5: The Metrics of our Journey - OCLC Results
- Step 6: Where are we today, and where do we go now?

Step 1: OCLC Meets DSDM

OCLC became ISO 9000 and TickIT certified in 1998 and has a long history of repeatable software development processes derived through a strong quality Assurance Organization. The “traditional” or waterfall approach to software development was well entrenched, and supported, by a system of policies, procedures, and templates: the overall approach was generally standardized across all project teams working in the Dublin, Ohio Company. After going through the process of documenting our Software Development procedures (over-documenting them as it turned out), solving the document control issues, and keeping all the records necessary to retain our ISO certification, we did not want to implement a method that would conflict with the changes we had made in our culture. During my research into Rapid Application Development (RAD) methods (we did not use the term *Agile* at the time), I came across a publication entitled “The Dynamic Systems Development Method and TickIT”. This publication provided a guide to reconciling the requirements of ISO 9000 with the procedures used for DSDM – but what on earth was DSDM?

Step 2: Does it really work that way?

I established that DSDM (Dynamic Systems Development Method) was an incremental development approach, promising on-time delivery, short increments and it seemed to be strong on user satisfaction. “That’s for us!” I thought. I set off on the trail of this DSDM approach. I identified some training, but it all seemed to be in England. Further research persuaded me that this approach really could help us, and it that it would be worth the trip to a 3-day course in Cheshire, UK for Accredited DSDM training with a local company, TCC.

After all, I could combine it with a spot of golf, and visit my ancestral home in Scotland after the class.

Driving from Manchester Airport, down roads barely wide enough for one car, let alone for two to pass, I arrived at the hotel where the course was running, a typical “olde worlde” English building (**Figure 1**).

The course began. Dot Tudor was the tutor. In addition to her considerable project experience with DSDM, she clearly had a passion for the DSDM approach. “With DSDM, we WILL deliver on time – every time! We WILL stay within budget! We WILL give you what the business needs!” I must have said something like “Would you care to come over to Ohio and prove it?” And so the road to DSDM began for OCLC.



Figure 1 – DSDM in Cheshire

Let’s Meet the DSDMer - Dot Tudor, TCC –

I’m an avid believer in the iterative approach, and in delivering on-time by prioritizing which features you deliver in relation to business need. Requirements will always change: new ones will emerge; and while we are building our software the world will just “move on” so that what we first envisioned will be just plain wrong! Why not involve the users throughout the development of *their* products. If we were making a suit of clothes for a customer and delivering it 6 months later, we wouldn’t expect our first measurement to still be perfect 6 months later (especially if the project spanned the festive season) so we would have regular “fitting” sessions. DSDM does the same - involving the right users and testing throughout the life-cycle. Yes, I am passionate about this. I’ve spent too many years using a structured, waterfall approach and trying to specify and estimate every requirement up front - some things are just not possible or sensible, however we try to deceive ourselves.

With DSDM, requirements can be agreed at a high level and prioritized in relation to their business importance. DSDM uses the acronym MoSCoW to focus the prioritization of requirements: M(ust have) for those requirements which it would be impossible to manage without; S(hould have) for requirements which, although important, we could work around in some way and manage without for at least a while; C(ould have) for requirements which would add business benefit but are not essential; and W(ont have) for requirements which have been discussed, but it has been agreed to drop for the current deliverable (increment) in order to focus on the more important ones. DSDM also relies on the use of a timeboxed schedule – planning work into specific periods of time (usually between 2 and 6 weeks) in which they either will happen or be dropped. Any timebox must have an element of Must,

Should and Could, to give flexibility to drop something non-essential if time is running out.

Step 3: Project Set-up and the “rolling workshop”

(The First DSDM Training and Consultancy in Dublin Ohio)

I was to delighted be invited to run the first DSDM 3-day class with a project team that needed to bring a product to market in time for the Library world’s biggest conference in January. Some work had been done before my journey to identify the Executive Sponsor, Visionary and Ambassador Users for the project and to identify those in need of training. It was Halloween when I arrived and one of my most vivid recollections is being taken by the project team, after the class, to see a Halloween Corn Maze (we seldom have such things in England). However, if the team were trying to lose me, they completely failed and I was back in action the following week, with a DSDM planning workshop!

The high level requirements for the project had already been established and agreed and the next step was to estimate their size, prioritize them and put them into a plan. We did this by having the Project Manager, Visionary (a key business role occupied by the business lead) and myself in a room for a full two days, running a kind of “rolling workshop”. This involved inviting the team (users and developers) into the room in twos and threes to estimate the work required for each of the features to be built and to commit time in their diaries to doing it. We had a paper-covered wall, marked out into weeks, ready to receive the requirements/products to be delivered. These all had to be entered onto large post-it notes, together with their priority (according to the MoSCoW priorities), size, personnel needed (both developers and users) and dependency on other requirements. We had secured different colors of post-it notes for the different priorities of requirements. It looked like the Amazing Technicolor Dreamcoat when we’d finished, with green "Musts", purple "Shoulds", yellow "Coulds" and shocking pink “Won’t Haves”. **Figure 2** is a picture of a Timebox plan for the ILL Policies Directory Project. You can see the style of our planning – very informal, but very visible!



Figure 2 – Timebox Plan

It was our first such plan in OCLC and we hadn't quite got the stationery department to understand our needs. However, the rather bright colours had the desired effect of making clear the M(usts), S(houlds) and C(oulds), as evidenced by the following incident. Half way through our first planning day, George Walter came into the room. The room was a bit of a corridor, long and narrow, with a door at each end. He came in through one door, glanced nonchalantly across at the plan, from a considerable distance, and casually observed, "Hmmm, rather a lot of M(usts) in there". Without stopping, he continued on out through the other door. We all looked at each other, and then back at the plan. "He's right, you know!" We proceeded to completely rework the plan to allow not just M(usts), but also some S(houlds) and C(oulds) in each timebox!

In planning a DSDM project, it is very important that the work should take place exactly in the timebox in which it is scheduled and that both user and developer time are included. One issue we faced on this first project was that the developers found it difficult to commit specific days or weeks of time because they were also responsible for supporting the current live systems and this “fire-fighting” was a large and unpredictable commitment which had hampered progress in the past. With the agreement of the business lead and the project manager, we rescheduled the support responsibilities into a rotation, so that developers all took two weeks dedicated to support followed by four weeks when they were fully available to the project. That way, a skeleton staff was always solely covering support requirements and the others could concentrate on their timeboxed activities within the plan.

Our maxim was, “In a DSDM timebox, commit to only what you really believe you can do – but then, do what you committed to within the timebox.”

The project was a tight fit for the time we had (do you know how many holidays there are between Halloween and January!) but the team had estimated, and they stuck to their estimates and delivered what was needed on time. Not all of the Shoulds and Coulds were finished, but all of the Must Have requirements were and many Shoulds and Coulds too.

Let’s shift back to the OCLC perspective -

The immediate success of the first DSDM combined training / project initiation made it easy continuing with the method. OCLC was organized based on product lines yielding multiple software development groups (silos) all functioning somewhat differently. Dot quickly developed a working relationship with many of the influential Technical and Business Leads so our decision to continue using DSDM was an easy decision. The advantage of having a common method, process, and language to use within our project teams among the various development groups gave us a reason to continue using DSDM even as new and different agile methods were introduced.

There were some culture changes needed and some entrenched ideas to break down. As we looked back we remembered when we started the workshops following the 3-day practitioner class for the NCIP project. Our Business Lead had with her a list of all the requirements for the entire product (a very large new product that we knew would take several 6-8 week timeboxes to complete). We asked her to try and determine the *Must* haves for only the first timebox (6 week effort) and of the 48 requirements she determined the 46 were must haves for the first timebox. Persuasive tactics were needed! Actually, we just dug our heels in and said, “No deal!” This point was a reality check for the particular product group in question: the message that the culture was going to have to change began to be accepted from this point on.

Each time the training was offered, it was scheduled around the initiation of a new project which was considered to be well-suited for the DSDM method. (DSDM has a Suitability Filter for this purpose.) The 3-day DSDM Practitioner class was followed immediately by 2 – 7 days of kickoff, prioritization and timebox planning for the project being targeted. OCLC worked with Dot to train staff members and immediately walk them through the real world application of the method. The combination resulted in increased effectiveness of the training and brought about the culture shift we were seeking.

Step 4: We also need to tailor to our processes

Adjustments to the development life cycle procedures and templates were required in order to facilitate iterative development. In the end, we identified that what each project needed was a timeboxed project plan, an agreed set of requirements, a design review, a test plan, and test records.

The benefit OCLC was attempting to achieve was delivery of the primary functionality for systems in a fraction of the time required by traditional approaches. OCLC believes that DSDM provides an approach characterized by **quick time-to-market** through flexible requirements. That means teams could expect to give the user the necessary functionality very quickly and add enhancements incrementally through subsequent iterations. **Table 1**, below, shows the nine DSDM guiding principles and the results of applying them at OCLC.

There has been difficulty in embracing all 9 guiding principles. The principles of “all changes during development are reversible” and “testing is integrated throughout the lifecycle” have proven to be more of a challenge. The reality is that while unit testing or some level of integration testing might be occurring throughout each timebox, system and acceptance testing is still occurring, against a “frozen” code base in a controlled testing environment, just prior to installation. Also, the level of current software configuration management along with the size and complexity of the systems being impacted have made it difficult at best to actually be successful at reversing each and every specific change during development.

	DSDM Guiding Principles	Principles strongly adopted by OCLC teams
1	Active user involvement is imperative	Active user involvement is critical – project teams are being co-located so that ambassador users are integrated throughout development.
2	DSDM teams must be empowered to make decisions	DSDM teams must be empowered to make decisions – since deliverables are smaller, authority to determine what is and is not included has been delegated to the team members

3	The focus is on frequent delivery of products	Most products now perform all software maintenance and enhancement projects through use of DSDM in 4 to 6 week cycles. The result is a constantly evolving product or service
4	Fitness for business purpose is the essential criterion for acceptance of deliverables.	When the user realized they will get the next enhancement in 4 to 6 weeks they are willing to streamline inclusion to those items that are truly "Must haves".
5	Iterative and incremental development is necessary to converge on an accurate business solution.	OCLC products and services are constantly evolving through the use of 4 to 6 week enhancement cycles.
6	All changes during development are reversible.	See above.
7	Requirements are baselined at a high level.	This has been a radical change for OCLC. What once took 400 pages can now be accomplished through timeboxing at a high level with continual involvement of users working with developers to achieve the desired end.
8	Testing is integrated throughout the lifecycle.	See above.
9	A collaborative and co-operative approach between all stakeholders is essential.	The continuous team communication has developed a high degree of trust and enabled practitioners to create better quality in a fraction of the time.

Table 1 – DSDM Principles and OCLC Results

Let's look at some more project examples:

The Cooperative Online Resource Catalogue (CORC) was a very large project and completed prior to the implementation of full DSDM. However, we had learned about some of the DSDM techniques and CORC was considered successful because of a shift to iterative development and the use of prototypes. The project development time was still fairly long, but the entire team felt the implementation was much quicker and more successful through the use of rapid application development methods.

The NCIP project, the first to use DSDM practices, was again very successful. Positive comments during the post project review indicate the use of prototypes and manageable timeboxes (6 week) facilitated a very successful outcome. A lesson learned was that developers

are often reluctant to change an estimate once it has been posted in a timebox. This could lead to quality issues since they might take a quick fix approach in order to meet the date without focusing on quality of the deliverable. **Figure 3** shows the NCIP timebox plan including the resources, schedule, and requirements that will be satisfied in the timebox.

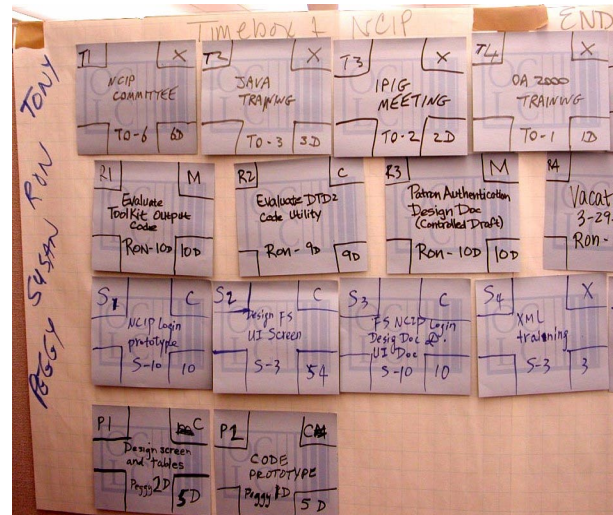


Figure 3 – NCIP Timebox Project Plan

The Digital Archive project wasn't an obvious good fit for DSDM, due to the size and complexity of the product; however, the DSDM methodology allowed the team to better track deliverables and meet the expected installation date by applying the iterative development approach.

One benefit identified on most of our projects is that of the visual aspect of DSDM "schedules". Any member of the team can walk into the room and take a quick look at the timebox sheet and know what they are expected to complete for the immediate timeframe. Additionally, frequent meetings, small team size, and user involvement during development result in better communication and hence better success at meeting requirements.

The primary requirements for documentation have shifted as a result of adopting the DSDM approach. No longer do we need voluminous requirements documents; a picture like the one here can be stored in the document management system and routed for approval.

Additionally, the number of required documents has been streamlined considerably which results in less overhead and more focus on deliverables for the product installation.

We have seen numerous variations on the methodology. As can be seen in Figure 2, timebox tasks are sometimes arranged across swim-lanes assigned to team members. This makes the very useable schedule a better fit for this particular team.

Some teams have been less successful in making the shift from traditional approaches to DSDM. However, even those teams that typically still develop in a waterfall approach have adopted some of the DSDM tools and techniques. The visual timebox sheets are one example of a widely-adopted approach to managing schedules. Often, teams still develop the software in a quasi-waterfall approach, but they will use the timebox sheets to facilitate status meetings and team cohesiveness.

Step 5: The Metrics of our Journey – OCLC Results

Measuring the effectiveness of implementing agile software development processes at OCLC has been very important over the last 3 years. Through the successful implementation of DSDM practices, we have begun to produce more reliable data which is being captured through post project reviews. This data is then tracked and monitored for improvement opportunities through the extensive use of dashboards. **Figure 4** is the dashboard used to track the effectiveness of the software development life cycle. This dashboard tracks the results for project schedule, deliverables, costs, and process effectiveness.

Step 6: Where are we today, and where do we go now?

Although teams have tailored the DSDM methodology to work better for their needs and some have only adopted tools and techniques, overall the DSDM approach has been a very positive move for OCLC. Whereas most projects used to take 18 – 24 months to deliver a product, they now typically take 2 - 3 months. Many teams do quarterly releases which include incremental enhancements using DSDM along with maintenance support.

Unique Challenges

In addition to the stories above, these are a few of the many challenges we have faced:

- There are many layers of customers and stakeholders. Decision-making can be slow and access to real customers is difficult (sometimes even undesirable as certain product information is commercially confidential). However, it always proved beneficial having the intended end-user of the product working with developers to ensure that the features were being built in the right way and we made considerable effort to do this, co-opting a librarian from one of the customer libraries for some projects and using customer focus groups to gather feedback and ideas for others.
- Some of OCLC’s working practices, established over many years, did not suit the DSDM mixed-team approach. A much more “us and them” approach to development had been the pattern and people have needed to become accustomed to working together as a hybrid team, taking responsibilities for different aspects of the development process than they were used to. With time and training, we have managed to establish the benefits of the mixed team, the successful implementation of which has considerably improved user/developer working relationships.
- The requirement for compliance with strict ISO9000 quality standards has had to be accommodated, whilst meeting challenging deadlines for delivery of high-profile products to the marketplace. We have addressed this by not allowing the most extreme agility of process but have allowed considerably more flexibility than previously.
- Buy in – winning hearts and minds. Not everyone initially welcomed the change with enthusiasm. Some users expressed the concern that the MoSCoW prioritization would mean that they would get less functionality. However, experience has shown that the users working alongside the developers are able to exercise control over what is done and what is omitted, and the on-time delivery has proven to be a very valuable aspect of the approach. One thing that has allowed the change to occur is the growing confidence of users that if they do allow a S(hould) or C(ould) to be dropped, it will be included in a subsequent iteration very quickly.

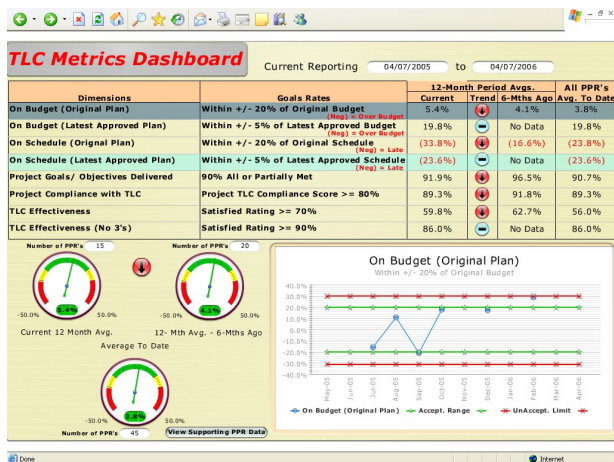


Figure 4 – The Life Cycle (TLC) Metrics Dashboard

- Metrics – how successful have we been if we omit all of the “S(houlds)” and “C(oulds)”? Initially, it was difficult to decide how to view the omission of S and C requirements. Was the project totally successful if only Must haves were delivered? This aspect has to be negotiated with users as the only way of ensuring that they get the right quality and on-time delivery of their most important features.
- Quality – how do we reconcile the approach with organizational expectations? This aspect was made easier as it was an initiative of the strong Quality function within OCLC to introduce DSDM. The senior Quality staff had a clear idea of where compromise could be made, but also of what practices would be acceptable. They also had the will to give the new approach a chance, which paid enormous dividends in the end.

Conclusion

So, in conclusion, while we have not yet managed to achieve the ideal agile development process, OCLC has moved a long way on the *Road to Agility* and is still moving in the right direction. This has been a very big WIN for us.

REFERENCES

- The DSDM Consortium* www.dsdm.org
- The DSDM Framework* www.dsdm.org
- The DSDM Student Workbook*
IJ and DJ Tudor Galatea 2002
 ISBN 0-9543071-0-0
- Rapid Application Development*
J Martin Maxwell/Macmillan1991
 ISBN 0-020376775-8
- Team Roles at Work*
R M Belbin, Butterworth-Heinemann, 1993
 ISBN 0-7506-0925-7
- The Dynamic Systems Development Method & TickIT
 ISBN: 0 580 27081 5