Is XP agile enough for the development of Web information systems?

Michael S Lane
Glen Litton Van der Vyver
Andy Koronios

Division of Information Systems, Faculty of Business,
University of Southern Queensland,
Toowoomba, Qld, Australia, 4350
e-mail: lanem@usq.edu.au
e-mail: vandervy@usq.edu.au

School of Information and Computer Science,
University of South Australia,
Mawson Lakes Adelaide, SA 5095, Australia
e-mail: Andy.Koronios@unisa.edu.au

Abstract

Agile methodologies are being touted as the next major paradigm shift in development of information systems. However, there is little empirical evidence to determine how well suited agile methodologies such as Extreme Programming (XP) are for fast paced web development. This paper presents the findings of an empirical study that investigates the suitability of XP for web development. Evidence from the findings would suggest that there are many benefits to be gained from using XP. However, there are also many problems in using such an approach because web development is embedded in HTML. However, as the Web becomes more object-oriented with emergence of Web services, agile methodologies such as XP will become increasingly important and more suited to web development.

Keywords

IS Development methodologies, Agile methodologies, Web systems

INTRODUCTION

The modern economies of the world are moving rapidly to a networked knowledge-based global economy where real time, distributed processing capability and access to information in real time is preferred and often considered critical. Organizations increasingly need to develop E-Commerce applications in short time frames to support their business needs such as the delivery of products and services over the Web. Failure to deliver E-Commerce applications to tap the opportunities presented by the market may leave adverse effects on the business and its competitive advantage. Traditional Information Systems (IS) development approaches that focus on processes and documentation, not only struggle to deliver in the time frames demanded by a networked business environment, but also fail to match the needs of customers. On the other hand, web information systems are changing continuously to address the challenges and opportunities posed by the electronic commerce paradigm.

Recently, agile methodologies have gained much attention in practice and have assisted a major paradigm shift in the way information systems are developed. The advocates of agile methodologies argue that these techniques address the shortcomings of traditional information system development methodologies (Beck, 1999; Canos et al., 2003; Fowler, 2000; Highsmith, 2001; Kalermo & Rissanen, 2002). Their main claims are that agile methodologies such as XP deliver working systems within the short time frames that the current business environment demands as well as delivering business value to the customer. It would therefore appear that agile methodologies offer a solution to many of the problems identified by MacCormack (2001)in developing web information systems under pressure. However, there also appears to be some shortcomings with agile methodologies. They may be more suited to particular project types as it relies on the philosophy of cooperation and collaboration, which is more easily achieved in small projects.

In this research paper, we present the findings of an exploratory first phase of a research project that assesses the suitability of agile methodologies for developing E-commerce applications for the fast-paced web based environment . This paper is divided into four sections. Firstly, an overview of agile methodologies is presented followed by a detailed discussion of most widely adopted agile methodology XP. Next, the research question investigated in this exploratory first phase of the research project is outlined. The qualitative method used for data collection, and the data analysis techniques and tools used to analyse the data are described and justified. The following section discusses the findings emerging from data analysis. Finally, the conclusions and the implications for future work in this research project are discussed.

CHALLENGES AND DIRECTIONS OF WEB INFORMATION SYSTEM DEVELOPMENT

The development of information systems for the Web poses many challenges which cannot be resolved by traditional systems development approaches. In a networked business environment , market pressures such as short product life cycles, and shorter lead times, mean that information systems have to be developed in short time frames(MacCormack, 2001). Indeed, there are financial products such as those offered by financial institutions and insurance companies, which have extremely short life cycles Another major challenge is the ability to deliver network connectivity and information interoperability that organizations desire, a challenge bedevilled by the range of disparate and legacy systems with which a web application might need to interface with. A significant trend in this regards has been the gradual realisation of the networked design reliant on thin client architecture.

The emergence of the J2EE and .NET architectures is a direct response to the business demand for Web services based on thin client architectures (Vawter & Roman, 2001). Graham (2001) argues that much of the next generation information systems will be server based and that will pose significant challenges for traditional development approaches. He argues that information systems developed for a thin client architecture will present a great opportunity for start-up companies that have innovative flexibility and the commitment to develop on such architecture. Another challenge for the development of E-commerce applications which interface with the general public is the difficulty involving the end-user at the client end of E-Commerce applications given their relative autonomy and anonymity (Lane & Koronios, 2000).

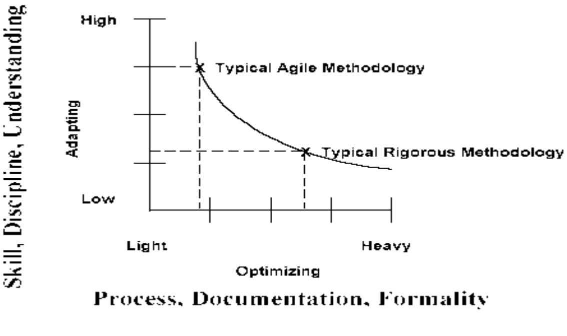
DEFINITION OF AGILE METHODOLOGIES

In providing a precise definition for agile methodologies, it is useful to firstly define the term "system development methodology" in the context of the discipline of information systems (Kalermo & Rissanen, 2002). An information system development methodology can be defined as an organized collection of concepts, beliefs, values and normative principles supported by material resources. The purpose of an information systems development methodology is to help a development group successfully change object systems, that is to perceive, generate, assess, control, and to carry out change actions in them (Lyytinen, 1987b). For this paper we use the following definitions to clearly describe the characteristics of agility methodologies.

> "For a software development organisation: Agility is the ability to adopt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment. An agile methodology is one that readily embraces and supports this degree of adaptability. So it is not simply about the size of the process or the speed of delivery; it is mainly about flexibility" (Kruchten, 2001, p. 27).

While Kruchten (2001) stresses adaptability and flexibility, Cockburn (2001) also emphasizes that agile methodologies are based on the use of light but sufficient rules of project behaviour and the use of human communication oriented rules. Consequently Agile methodologies are flexible and responsive to changing business requirements because of less reliance on formalized processes and documentation (see Figure 1) (Canos et al., 2003).

Figure 1 Comparison of attributes of agile system development methodologies with traditional system development methodologies (source adopted from Highsmith2001)



Agile Alliance and Agile Manifesto

The upsurge in grassroots developer support for agile methodologies has resulted in the formation of an organisation called the Agile Alliance (2003). The Agile Alliance was formed to promote the concepts of agile methodologies and a general lightweight, flexible approach to developing information systems. The Agile Manifesto (2003) outlines the concepts behind the ideas and system thinking of agile methodologies The Agile Manifesto consists of four conceptual ideas which drive the agile methodological approach:

- 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.

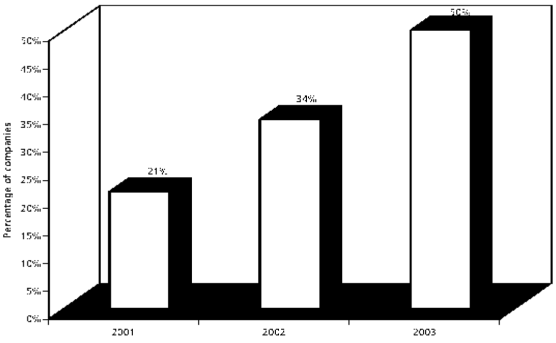
FUNDAMENTAL DIFFERENCES BETWEEN AGILE METHODOLOGIES AND TRADITIONAL METHODOLOGIES

Martin Fowler, in his thought-provoking article 'The New Methodology', outlines many of the distinct differences between traditional process oriented methodologies and agile methodologies (Fowler, 2000). Agile methodologies are lightweight in terms of the effort needed to apply the methodology in contrast to traditional methodologies which are heavyweight in process and documentation. In essence, lightweight methods are adaptive rather than predictive and welcome change in a software project. At the same time, they are people oriented rather than process oriented. Overall, the traditional engineering metaphor does not fit well with building software because design requires so much more creative effort in comparison to engineering a physical construction such as a bridge. Agile methodologies acknowledge that requirements are always changing and accommodate change which make agile methodologies similar to the 'evolutionary delivery model' of software development proposed by Gilb (1988). The iterative nature of agile methodologies means the frequent release of working systems allow developers to have some sort of control over an unpredictable process.

The very nature of agile methodologies mandates a different kind relationship with a customer than the type of relationship with the customer characteristic of traditional methodologies (Highsmith, 1998). The end user needs to work closely with the developers to ensure that the software developed meets their business needs. Agile methodologies require a skilled team. Developers and customers, that is 'people' are the single most crucial element rather than replaceable components in a process (Cockburn, 1998). People are highly variable and non-linear, contributing strongly either to success or to failure in software projects at different times.

While agile methodologies are still not yet a mainstream practice, evidence from an extensive recent survey of 200 IT managers worldwide by the IT Cutter Journal (see Figure 2) would indicate that agile methodologies will become more prominent in next couple of years (Charette, 2003).

Figure 2 Percentage of companies with more than 50% of projects defined as agile (source adopted from Charette2003, p. 3)

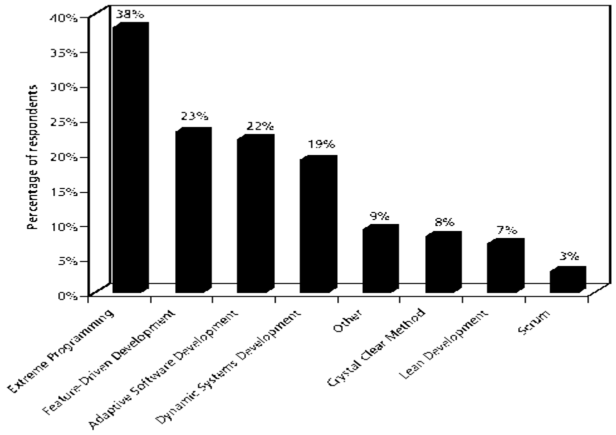


EXTREME PROGRAMMING (XP)

On the face value of anecdotal evidence, XP would appear to be the most widely adopted agile methodology (Williams, 2000, Baskerville, Levine, Pres-Heje, & Ramesh, 2002; Highsmith, 2001; Miller, 2003). XP was first described in the late 1990s as a collection of process/organizational patterns and, since 1999, has been the focus of a number of methodological publications (Amber, 2002). XP has certainly received enough attention in industry and in the literature, even making a mention in a Dilbert cartoon recently. Figure 3 is a comparison of the adoption rate of the various agile methodologies drawn from the analysis of the results of recent survey of

200 IT managers by IT Cutters Journal on the adoption of agile methodologies by practice. Figure 3 emphasises the dominance of XP as an agile methodology used in practice which is why XP was the focus of this study on agile methodologies.

Figure 3 Comparison of the adoption rate of agile methodologies (source adopted from Charette2003, p. 3)

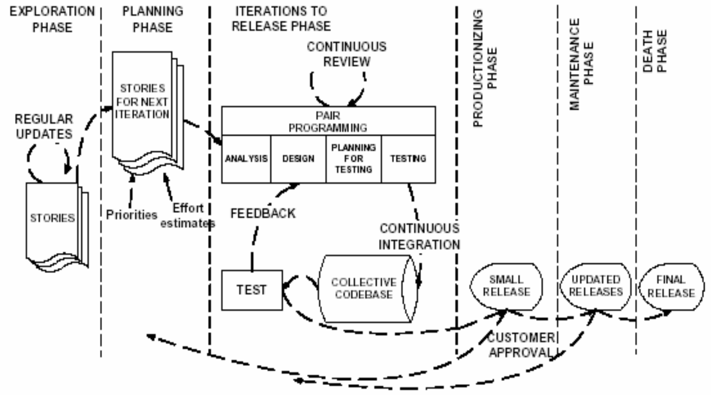


The XP methodology has 12 core practices (Beck, 1999):

1. The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Changing requirements are welcomed, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Software is delivered quickly and frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Projects are built around motivated individuals. Give them the right development environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 4 illustrates how the 12 core practices of XP fit into the exploration, planning, iterations to release, productionising, implementation and death phases of the system development life cycle (SDLC) (Abrahamsson, Salo, Ronkainen & Warsta 2002). The terms used for the various phases of SDLC in the XP approach, although somewhat different to those used in the traditional descriptions of the SDLC, still emphasize the standard analysis, design, build, implementation, maintenance phases in an ongoing cycle until the ultimate death of a system. However, in the XP approach there is an emphasis on priority of customer/client stories which represent the business requirements for each system release. The XP approach involves rapid iterations of small releases of system functionality/features, with developers writing the tests for system functionality/features before actually writing the specific code. The involvement of customer from the inception of a project through to the Customer/client acceptance testing before production release of code ensures strong buy in by the Customer/client.

Figure 4 Different activities of XP associated with the different phases of the system development life cycle (source: adopted from Abrahamsson et al., 2002, p. 19)



RESEARCH QUESTION AND METHOD

The general research question that was explored in the first phase of this study was:

How suitable are agile methodologies such as XP for fast paced web development?

Our motivation and justification for conducting exploratory study, is that while agile methodologies have gained the attention of practitioners, academics are only just starting seriously to research the worth and impact of agile methodologies. This claim is evidenced by the lack of empirically grounded studies on agile methodologies and recent formation of Network for Agile Methodologies Experience (NAME) (Canos et al., 2003). A qualitative approach was deemed appropriate to guide the focus of the first phase of this exploratory study (Miles & Huberman, 1994; Yin, 1994). A qualitative approach allowed the researchers to explore the research question in an interpretative manner inline with the research objectives. It was necessary to capture rich domain knowledge from practitioners experienced in using an agile methodology such as XP, in order to identify the key issues in relation to the general research question that reflect the reality of current practice.

A convergent interview was conducted on an online XP discussion forum to solicit the opinions of practitioners. The convergent interviewing technique allows the researcher to identify and confirm the key issues when there is little previous theory available (Dick, 1990). In the next phase of this study we will conduct more structured interviews with respondents that have agreed to be involved in the next phase of the study. The novel approach of conducting the convergent interview using an online discussion forum added an extra dimension to the technique. That is, such approach allowed the respondents to interact and comment on key issues raised by other respondents in relation to the general research question posted on the online discussion forum. Of the practitioners who responded to the convergent interview conducted online, some are acknowledged world-known experts in using agile methodologies such as XP.

This study focused on XP as an agile methodology in the investigation of the suitability of agile methodologies in fast paced web development because it is the most widely adopted agile methodology in practice (Baskerville et al., 2002; Highsmith, 2001; Miller, 2003; Williams, 2000). This claim is supported and evidenced by the number and size of online discussion forums on the Internet devoted to sharing knowledge amongst practitioners regarding XP agile methodology. The extreme programming online discussion forum (http://groups.yahoo.com/group/extremeprogramming/) alone has nearly 4,000 members. The general research question was posted on this XP online discussion forum.

The responses to the general research question by the XP discussion group members were collated and stored in a text file. These were then analysed using the qualitative data analysis software package NVivo 1.3 (QSR

International, 2003). This software package allowed for the exploration of raw data and identification and coding of the common themes as well as the identification of relationships between themes in a rigorous manner. NVivo 1.3's modelling tool was used to identify and validate the findings of this study. While there are some limitations in the approach used, it is felt that the richness of the data collected far outweighed the methodological shortcomings of such an approach. Extensive use has been made of direct quotes to highlight key issues raised by respondents in the study. The coding for direct quotations is as T = Text file which contained the collated responses to the research question and numbers = line numbers of the sourced quotation in the text document stored in NVivo.

FINDINGS

A number of key issues were identified in relation to the general research question: *How suitable are agile methodologies such as XP for fast paced web development?* These findings were grouped into positive and negative issues and a discussion of these issues follows.

Positive issues identified in relation to the general research question

The positive issues identified in relation to the practitioner responses to general research question are presented in Table 1 2 & 3. A detailed discussion of these positive issues supported by some relevant comments from the respondents follows each table.

Table 1 XP encourages a disciplined approach to web development

XP enforces high quality design in web applications
A lot of problems with the development of web applications could be alleviated with XP approach

First, one positive theme that emerged from the data was the notion that XP actually encourages a much more disciplined approach to web development. The following comment by a practitioner illustrates the contrast between the disciplined approach of XP and the ad hoc grass roots approach that is common in web development.

> "*At least with XP, there's a small, clearly-stated set of practices to be disciplined about, and all of them are pretty obviously related to quality and speed.*" (T312:314).

Interestingly, the practitioner noted that a potential conflict of interest exists between the need to deliver web applications in short time frames and the disciplined approach that XP brings to the web development process. This potential conflict of interest is evident in the following comment by the same practitioner which emphasizes how the developer may be pressured to relaxing a discipline approach such XP to meet unrealistic expectations of management and customers:

> "*Using XP requires some discipline. It also requires managers and customers to support that discipline, instead of undermining it with pressure, fits of anxiety or cajoling to "skip testing, just this once". But doing anything right requires some discipline, and more pressure seems to require more discipline in just about any endeavour.*"

Another practitioner captured the thinking behind an undisciplined approach to developing web applications in the following comment:

> "*If you mean "we have to get this done in a week" as "fast-paced", then Agile or XP might be too much overhead for a slammin-fast project. And so the best methodology for one of these projects would be...? "Cowboy Coder"-ing ?*" (T211:213).

The disciplined approach of XP actually enforces high quality design in web development through a philosophy of simple design and continuous testing as evidenced by the following comment:

> "*I have found that XP helps to maintain very high levels of quality through unit tests, simple design and test driven design. The high quality gives you a lot of confidence in your website which is an advantage since a lot of websites are flaky.*"(T174:177)

XP's emphasis on small releases in short iterations facilitates the fast paced development of web applications by allowing business decisions in a web site to be implemented quickly, gaining quick feedback from the customer regarding any changes or enhancements (see Table 2).

Table 2 Small releases of working code in short iterations suits web development

Short iterations associated with XP approach allow real feedback from the customer
Web development environment supports short releases in small iterations

Supports quick implementation of business decisions
Dynamic web sites require version control
Version control on early releases of new features

A number of comments by practitioners emphasize advantages of XP's ability to support frequent small releases in fast paced web development

> *"Speaking broadly, I think XP's advantages for fast-paced web projects are even stronger than for other types of project: (1) Having all the running copies of the program under your direct control makes frequent releases much easier. (2) Because your competitors are just one bookmark away, it's much harder to insulate yourself from competitive pressure. That makes the ability to adapt quickly much more important. It also gives you a bigger advantage if you can keep ahead of your competitors.(3) Because new releases are adopted immediately, feedback comes quickly. This makes steering the project easier.(4) Since bugs can be very, very public, XP's ability to drastically reduce bug counts and bug severity are a big win. (5) In fast-paced environments, priorities often change rapidly. Releasing frequently means that when a project suddenly gets shelved, relatively little work sits around unreleased."* (T97:114).

However, web information systems are becoming increasingly complex and dynamic. In order to manage frequent change while at the same time maintaining quality it is important to have some sort of configuration/version control in place. There are a number of open source version control tools available to manage version control process in XP. The importance of version control and testing in dynamic generated web information systems was evident in the following practitioner comments:

> "*And because an XP project is "always in maintenance", it uniquely targets situations where end-users are using the release version at the same time as the team is adding features and removing issues" (T25:26)."As soon as you have dynamic behaviour, and frequent changes to that behaviour, you need at least to get everything under version control and institute some basic testing (T294:296). I worked on a few large web dynamic web sites for about a year -- content management, shopping-cart-style ordering, user registration, mass email based on searches, that kind of thing. Features tended to get added to the live site every 1-3 weeks. We always had some version of each site that we could quickly push to the live site if we needed to (e.g. to fix a bug or rush a feature)(T285:290).*"

Since the philosophy of agile methods such as XP is based on frequent small releases of working software, it is critical that the appropriate testing environment is in place to support such an approach. **Table 3** summarizes the positive issues that emerged in relation to continuous testing.

Table 3 Continuous testing approach of XP

Continuous testing of XP eliminate a lot of bugs in Web applications
There are some HTML testing tools to support the continuous testing of XP

The following comment by a practitioner noted the benefits of continuous testing such as reduction in bugs had been realized once he convinced the programmers of the worth of continuous testing in fast paced web development.

> *"I succeeded in introducing programmer tests in that company largely because I am very stubborn and dislike staying up late tracking down strange bugs. Secondary credit goes to co-workers who were more interested in getting things done than in dictating how I should work, and who were willing to learn about programmer testing when they saw its positive effect." (T304:308)*

One of the problems with web development is that the applications are embedded in HTML, hence there is a need for testing tools suited to the HTML environment. There are, however, a few HTML testing tools as noted by the following practitioner.

> *"The best Pseudo-OOP (or, POOP) approach I have found so far is Fusebox, which makes it POSSIBLE to employ encapsulation. Fusebox provides a "TestHarness" for Unit Testing, and some tools for collecting user stories in a site-structured way. I think some Agile techniques will be applicable. (Just getting started myself.) (T41:43) There is also CFObjects, which is not widely adopted, but which does do objects in ColdFusion." (T45:46)*

Negative Issues identified in relation to general research question

A significant number of negative issues were also identified from the practitioners' responses. Tables 4, 5 and 6 presents a summary of key negative issues that emerged from the responses of practitioners in relation to the general research question. A discussion of these negative issues follows each table.

Firstly, it would appear that XP is somewhat in conflict with the ad hoc approach of web developers in general (see Table 4). Indeed one of the practitioners cautioned about choosing your development team wisely if you want to use an agile methodology such as XP. Many web developers are often reluctant to adopt a disciplined approach such as XP.

Table 4 XP is disciplined approach which is in conflict with ad hoc approach to web development

Fast paced web development environment pressures developer to deliver something regardless of the quality
Web developers are reluctant to adopt a disciplined approach to web development such as XP

The following comment by a practitioner highlights the difficulty in getting web developers to adopt an agile method such as XP:

> *"I would also caution you to carefully choose who works on your XP web team. The web developers I have worked with are notoriously independent and inexperienced in working in groups. The group of web developers I introduced to XP, found XP to be too meddlesome with the way THEY wanted to write code. " (T182:186).*

It was noted that an ad hoc approach to web development was really only suitable for simplistic static web information systems (T293:294). However, the fast paced nature of web development places a lot of pressure on developers to deliver something regardless of the quality:

> *"Any fast-paced environment is going to put pressure on the developers to (1) do something now, or just (2) look like you're doing something now." (T304:305)*

In using an agile method such as XP for fast paced web development, another significant challenge faced is that web development is embedded in HTML and HTML by its very nature is a presentation language (see Table 5). Therefore you need to have XP development tools that accommodate the nature of the HTML development environment.

Table 5 Web development is embedded in HTML

HTML development environment is risk averse
HTML is a poor testing environment
Extremely difficult to automate the testing for client side functionality
HTML does not enforce OO principles
OO language is desirable for XP

Some issues that emerged were that the HTML development environment is inherently risky, given that it is so easy change HTML code and it is often tempting to go live with a new application or enhancement without sufficient testing. The following comment by a practitioner emphasizes the significant risk associated with web development given the very open and public nature of the Web platform:

> *"HTML & its support systems are very easy to refactor, and terrible to refactor as well. If the customer changes their mind, based on actually seeing the working site, changing it without up-front tests adds incredible risk." (T13:15)*

The risk is further extenuated by the fact that HTML is a poor testing environment, a situation which would be exacerbated by the lack of HTML testing tools support for agile methods. It was noted that it is also extremely difficult to test client side functionality of a web information system because of the unpredictability of scripting languages like Java script, VB script and DHTML across different platforms. The following comment by a practitioner highlights the difficulty of applying XP testing in a web development environment.

> *"The main limitation I can think of is that XP's focus on testability does not match well with the web environment in two ways: (1) One of the major criteria for a web site is "looks right". This is hard to automate. (2) Client-side functionality (javascript, dHTML, assorted components) are very popular with some site designers. This is fiendishly difficult to test automatically, especially across a wide variety of platforms." (T114:120)*

Indeed, many web developers have little time or inclination to test code:

> *"The HTML culture barely comprehends testing. So anything sent to the end of the lifecycle just might get skipped. The site looks okay, so publish it." (T15:16).*

A more fundamental problem that arises when applying XP to the web development environment stems from the fact that HTML does not enforce OO principles. The XP approach with continuous testing and small releases really does lend itself to the use of an OO language:

> *"Can't argue with that. It seems that OOP is essential to XP, and HTML isn't OO (T39). An Object Oriented language is seen to be a requirement, but I don't believe this is entirely the case as you can unit test, refactor, and pair program just as effectively in a non OO environment if you have the experience with that environment." (T378:382).*

Another major issue is that web information systems vary greatly in scope and complexity**Error! Reference source not found.**). A simplistic, static web information system does not require a substantive methodological approach like XP (See Table 6).

Table 6 Web development varies greatly in scope and complexity

Simplistic static web sites do not need any substantive methodological approach
Some web sites did not require any significant testing
Web site needs to support significant business logic to warrant XP approach
Gold plating of web sites is an overkill for many websites that only require basic functionality
With web projects there are a significant number of activities that are not programmed based
Aesthetics design is difficult to automate
Other agile methodologies such as Scrum maybe more suited to Web development

The following comments by practitioners highlight that simplistic static web information systems do not require a disciplined approach like XP.

> *Actually, I'd think there would be some web projects where it would be best to just slap it together and ship it with no rigorous automated testing: (T221:223) Say they want a static web site: Use a good authoring tool, and check for broken links. There; you're done. What could break?" (225:226)*

However, web information systems that are dynamic, complex and support significant business logic do benefit from using a method such as XP. The following comment by a practitioner emphasizes these points.

> *"Personally, I won't touch a web project unless it has a significant amount of business logic. It would be a waste of my time and their money: "Contract it out to someone who knows Front Page." I say.)" (T:229:231).*

At same it also worth noting that during the dot.com era many web developers were guilty of over-engineering web sites (gold plating) when all that many web information systems required was the appropriate functionality to support the business objectives of the organisation (Lane & Koronios, 2000). The following comment by a practitioner emphasizes this tendency by web developers to over engineer web solutions and provide unnecessary gold plating:

> *"The Dot Com Boom fed HTML a huge gallery of support systems, all of dubious quality. Your customer does not want Jade, for example, they want a working site. But if your contract sold them Jade, by golly they're gonna get Jade, even if it gives your project nothing and slows everyone down" (T16:19).*

Some other issues that emerged from the practitioner responses included design activities related to graphical interfaces and the aesthetics of web information systems. These types of problems do not constitute a direct criticism of the applicability of methods such as XP but they do indicate the complexity and diversity of the elements involved in the development of a web information system.

CONCLUSIONS AND IMPLICATIONS FOR FUTURE RESEARCH

Agile methodologies such as XP address many of the shortcomings of traditional information system development methodologies in delivering working systems within time. However, there is little empirical evidence of the suitability of XP for fast paced web development. From our findings it would appear that agile methodologies such as XP address some of the difficulties of developing systems for Web. XP actually enforces a much more disciplined development approach that accommodates the evolutionary nature of web applications and the need to develop web applications in short windows of opportunity. However, the temptation to deliver web applications without applying the rigorous testing of XP is great given the pressure from rapidly changing markets where the window of opportunity maybe short. A key issue relates to the degree to which web development projects are OO-based. We suspect that this issue will largely be overcome as web development moves away from the creation of static web pages to a more dynamic, integrated form.

It is clear that developing mission critical applications for the web is a complex, multidimensional activity. It is not a process that can easily be encapsulated within a formal set of rules. While XP would address problems related to speed and quality of delivery, it could impinge negatively on the more idiosyncratic, creative elements of web development. It is however probably true to say that any formal methodology would have a similar effect.

We conclude that the process of web development is adaptable to the fundamental concepts of XP but some give on both sides will be necessary. In the next few years, further research into the nature of the web development process and a resultant adapted XP model to meet the unique demands of this type of development, would certainly be worthwhile.

REFERENCES:

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods*. Retrieved 12th August, 2002, from http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf

Agile Alliance. (2003). Retrieved 10th March, 2003, from http://www.agilealliance.com

Agile Manifesto. (2003). Retrieved 28th January, 2003, from http://www.agilemanifesto.org/

Amber, S. (2002). *Answering the "Where is the Proof That Agile Methods Work" Question*. Retrieved 13th May, 2003, from http://www.agilemodeling.com/essays/proof.htm

Baskerville, R., Levine, L., Pres-Heje, J., & Ramesh, B. (2002). *Balancing Quality and Agility in Internet Speed Software Development.* Paper presented at the 23rd International Conference on Information Systems.

Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Reading Mass.: Addison-Wesley.

Canos, J., Holocombe, M., Janes, A., Marchesi, M., Moonen, L., Reisoli, M., et al. (2003). *Research Roadmap on Agile Methodologies*. Retrieved 12th May, 2003, from http://name.case.unibz.it/NameResearchRoadmap-v2/NAMERoadmapVf.pdf

Charette, R. (2003). The Decision Is In: Agile Versus Heavy Methodologies. *IT Cutters Journal -Executive Update, 2*(19).

Cockburn, A. (1998). Non-Linear, first Order Components in Software Development. *2003*(1st February 2003).

Cockburn, A. (2001). *Agile Software Development*. Boston, MA: Addison-Wesley.

Dick, B. (1990). *Convergent Interviewing*. Brisbane: Interchange.

Fowler, M. (2000). *The New Methodology*. Retrieved Ist January, 2003, from http://www.martinfowler.com/articles/newMethodology.html

Gilb, T. (1988). *Principles of Software Engineering Management*. Reading, Massachusetts: Addison-Wesley.

Graham, P. (2001). *The Other Road Ahead*. Retrieved 12th April, 2003, from http://www.paulgraham.com/road.html

Highsmith, J. (1998). *Adaptive Software Development: An Evolutionary Approach to Managing Complex Systems*: Dorset House Publishing.

Highsmith, J. (2001). *Agile methodologies: Problems, Principles and Practices.* Paper presented at the XP 2001.

Kalermo, J. M., & Rissanen, J. K. (2002). *Agile software development in theory and practice.* Unpublished Masters, Jyvaskyla, Jyvaskyla, Finland.

Kruchten, P. (2001). Agility with RUP. *IT Cutter Journal, 14*(12), 27-33.

Lane, M. S., & Koronios, A. (2000). *A balanced approach to the capture of user requirements in business-to-consumer web information systems*`, Queensland University of Technology.

Lyytinen, K. A. (Ed.). (1987b). *Taxonomic Perspective of Information Systems Development: Theoretical Constructs and Recommendations*. Chichester: John Wiley &Sons.

MacCormack, A. (2001). How Internet Companies Build Software. *MIT Sloan Management Review, Winter 2001*, 75-84.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An expanded source book*. Thousand Oaks: Sage.

Miller, R. (2003). *Demystifying Extreme Programming*. Retrieved 13th May, 2003, from http://www-106.ibm.com/developerworks/java/library/j-xpcol.html

QSR International. (2003). *NVivo*. Retrieved 10th June, 2003, from http://qsr.com.au

Vawter, C., & Roman, E. (2001). *J2EE vs. Microsoft .NET*. Retrieved 15th January, 2003, from http://www2.theserverside.com/resources/articles/J2EE-vs=DOTNET/article

Williams, L. (2000). *The Collaborative Software Process. Dissertation,*. Unpublished PhD, University of Utah.

Yin, R. K. (1994). *Case Study Research - Design and Methods*. Thousand Oaks, California: Sage.

COPYRIGHT