

The Influence of Short Project Timeframes on Web Development Practices: A Field Study

Michael Lang

Business Information Systems Group, Cairnes School of Business & Economics, NUI Galway, Ireland. Michael.Lang@nuigalway.ie

Abstract: A number of recent surveys of Web development have revealed that typical project timeframes are of the order of 3 months. This paper reports the findings of a field study conducted in Ireland which set out to contribute towards a better understanding of the nature of high-speed Web development practices. Qualitative interview data was gathered from 14 interviewees, purposefully selected from a variety of different organizations and backgrounds. This data was then analyzed using the Grounded Theory method, and ten core dimensions were revealed: (1) the role of collaborative groupware tools; (2) collective code ownership; (3) timeframe driven by business imperatives; (4) enablers of productivity; (5) quality “satisficing”; (6) requirements clarity; (7) process maturity; (8) collectively agreed project schedules; (9) closeness to client; and (10) working software over documentation.

1. Introduction

Project timeframes can dictate both the choice of a systems development method as well as the extent to which its various features may be used. A number of recent studies of Web-based systems development reveal that average delivery times are now about 3 months [1,7], as further confirmed by the author’s own survey of Web development practices in Ireland [9]. A few years ago, this apparently hectic so-called “Web time” development context was alleged by Thomas [16] to give rise to “guerilla programming in a hostile environment using unproven tools, processes, and technology”. More recently, Baskerville & Pries-Heje [3,4] found that short timeframes can lead to practices such as “coding your way out”, and “negotiable quality”.

However, almost 70% of the respondents to the aforementioned survey reported having no or minor problems coping with the accelerated timescales of Web development, it being a “major” issue for just 4%. The motivation for this paper was therefore to follow up on the survey findings with a qualitative field study which looked more closely at the nature of high-speed Web development practices.

2. Research Method

A field study consisting of semi-structured qualitative interviews with 14 Web designers/developers was conducted. A purposive, theoretical sampling approach was taken in selecting interviewees so as to seek out similarities and dissimilarities, looking at both typical and atypical cases [6,12]. The profile of selected interviewees varied according to organisational size, organisational type (e.g. commercial/public sector or not-for-profit/private sector), nature of activities (e.g. Web design, digital multimedia design, “traditional” graphic design, and/or “traditional” software development), application domains, location of end-users (in-house -v- external clients), and professional background of interviewee (e.g. software development, graphic design, or other), as shown in Table 1.

With regard to application domain, it was observed in our previous survey [9] that certain industry sectors (e.g. Financial Services, Computer-based Training) may be different from others as regards their use of development processes, methods and techniques. However, rather than selecting interviewees from a broad variety of different industrial sectors, it was decided to select a number of interviewees whose *clients* are from a broad variety of different sectors (i.e. Web development houses and design agencies) and to ask them how their development processes vary from one client to the next, if at all. A limitation of this approach is that there may be differences between, say, developing systems *for* a bank (outsourced development) as opposed to developing systems *within* a bank (in-house development), particularly where critical systems that require the application of specialised domain knowledge (e.g. advanced security) are not out-sourced. That said, in selecting the interviewees for the field study, it was not feasible to contemplate all possible application domains and specialised considerations thereof, so the adopted strategy was seen as a good, if not optimal, way of eliciting data with regard to the influence of different application domains on process tailoring.

In most of the organisations visited, one personal interview was conducted with the team leader, typically convened during the mid-day break so as not to encroach upon busy work schedules. In one organisation two developers were separately interviewed, and in another the managing director brought five staff members into the meeting room. Where available, secondary data sources were also consulted. Data gathering continued until a point of reasonable “theoretical saturation” was reached. The data was analysed using a hybrid method, mainly based on the procedures of grounded theory [11,14], but also informed by the principles laid down by Miles & Huberman [12].

Table 1. Profile of interviewees

Organisation	Industry <i>(all private sector unless otherwise specified)</i>	No. of emps	No. of developers	Interviewee job title	Interviewee background	Interviewee experience (years)
Bizweb	Web development	60	40	Managing Director + MIS Applications Architect + QA Manager	Business studies / Software development / Industrial engineering	(unknown)
Clearscope	Web design agency	12	7	Senior Designer	Graphic design	7
DigiCrew	Web development	5	4	Internet Software Engineer	Computer games development	10
JobsPortal	On-line recruitment firm (in-house)	50	3	Web Project Manager	Software development	5
KL Design	Graphic design	5	5	Managing Director	Graphic design	12
Martech	Web design agency	10	8	Creative Director	Graphic design	9
OEG	University (public sector / in-house)	1,300	2	Chief Web Technologist	Software development	9
OEG	University (public sector / in-house)	1,300	2	Web Editor	Physics / Web development	10
Broadcorp	Broadcast media (public sector / in-house)	2,000	8	Web Project Manager	Industrial design	6
Redmoon	Visual communications	8	8	Managing Director	Film-making / Journalism	> 10
Strata	Web development	7	5	Creative Director	Industrial design	10
IBUS	Web development	30	10	Managing Director	Software development	10
Webcorp	Web development	25	20	Commercial Director	Business studies	10
W3M	Web portal	2	1	Managing Director	Physics / Software development	15

3. Findings and Discussion

Our analysis of the interview data revealed ten core dimensions of high-speed Web development practices, as explained in the following sections.

3.1. Role of groupware tools as enablers and drivers of collaborative work

To support their project management process, Martech use an in-house workflow/job management tool where members of the team regularly update each other on each others' status. The use of similar groupware tools to support team collaboration was also mentioned by Webcorp, Bizweb, DigiCrew and IBUS, not just for project co-ordination but additionally for aspects such as code sharing, bug tracking, and documentation of guidelines. A common pattern which emerged from interviews was that successful experimentation with open source or trial software often antecedes and expedites process definition, rather than the other way round. Otherwise put, the search for a simple, useful, shareable tool to address an ongoing problem that has attained chronic magnitude (e.g. requirements change control) can lead to efficient workable processes being built up around that tool. This is very much in line with Suchman's notion of "situated action" whereby "people use their circumstances to achieve intelligent action" [15].

3.2. Collective code ownership and ease-of-maintenance

At Bizweb, the organisation with the largest number of developers (40) of those visited, inefficiencies arising from the collective ownership of code, – or more accurately, the lack thereof, – has driven them to standardise their working methods and devise a mechanism whereby programmers can access and edit each others' code:

"We've found in the past where somebody might end up wasting 5 or 6 days trying to work out and re-do somebody else's code. A lot of that was happening, it was ridiculous. Because people are going to get sick or take annual leave, so the continuity factor just didn't exist. It was you alone as regards your code, and that led to phone calls on holidays, which again leads to staff morale going down. So the [newly introduced] development procedures do work really well because it is trying to get everybody on the same track. And it forces you to examine your coding practices as well, and learn lessons about how things can be done better." (MIS Applications Architect, Bizweb)

This idea of collective code ownership, which notably is one of the tenets of Extreme Programming (XP) and other "agile" development methods, and the concomitant issue of ease-of-maintenance was also commented upon by the creative director at Martech:

"*Simplicity* would be something that we would value a lot. Is the programming solution as simple as it can be? Are we inviting trouble down the line with this? Can another programmer pick this up and understand how it's written and why it's organised in this way? Equally, we would place importance even on the way designs are constructed. For example, a Photoshop document can be very complex with hundreds of layers, so it

should be constructed in a way that makes it easy to re-use and modify, or for another designer to pick it up and key into.” (Creative Director, Martech)

Another mechanism commonly used with the aim of achieving more cohesion and collective ownership within design teams is the use of regular morning briefings, a practice mentioned by quite a few of the interviewees which again is similar to the agile methods idea of daily stand-up meetings.

3.3. Timeframe driven by business imperatives: developer-push and client-pull

It became apparent that, in many cases, the imperative to deliver systems quickly is as much if not more driven by the desire of Web design agencies to maximise throughput and revenue as by any sense of genuine urgency on behalf of clients. As such, most companies are working within the parameters of their own self-defined comfort zones. Notable exceptions are Broadcorp and JobsPortal, where pressing organisational deadlines are the norm and there is a very real and heightened sense of immediacy. In the majority of the other companies interviewed however, it seems to be more the case that clients tend to have fixed budget allocations, which in turn indirectly impose time constraints. As one developer explained:

“The price is determined by how many days you spend on it, so if it costs, whatever, well then you work out how many days you can spend on that job, and that’s all you can spend on it. Even if it’s not fully done 100% correctly, it still has to go out ... The developers and the designers will tell you that they want to finish the job perfectly, whereas the sales person and the project team will say ‘you can’t spend any more time on that job’.”

This excerpt raises the problematic issue of negotiating compromises between time, quality, and cost/resources; this potential trade-off and how it is managed is discussed later.

3.4. Enablers of rapid development and enhanced productivity

To state an obvious point, short delivery cycles in Web-based systems design have become the norm because they are *possible*. Otherwise put, the factors which enable rapid development also serve to raise expectations and therefore drive demand, be that from clients or from project managers. These enabling factors stem from a variety of sources. Firstly, Web-based systems can be rapidly deployed because the Web is an immediate delivery medium that, unlike traditional IS and off-the-shelf software applications, is not impeded by production, distribution and installation delays. Secondly, there have been dramatic gains in recent years in developer productivity. This is facilitated in the first instance by the availability of high-speed rapid application development tools for Web development e.g. ColdFusion. Thirdly, it has become common practice to make extensive re-use of libraries of pre-fabricated components and applets; templates and wizards for automatic code-generation; plug-and-play interfaces for database connectivity; and customisation of ready-made open source solutions. This has been refined to a

point where most development time is now invested into the ongoing evolution of an “out-of-the-box” “productised” solution, such as advanced content management functionality. Code production has moved from crude cut-and-paste re-use to instant automatic generation, meaning that most of the standard back-end functionality required for any given project can be up and running within a day or two. As an example of the scale of productivity improvements which have been achieved by the use of rapid development tools, automatic code generation, and systematic re-use, DigiCrew can now do in a week what would have taken two months to do just three years ago.

Fourthly, the idea of “picking the right tool for the job”, meaning the one that can get it done as best and as efficiently as possible, was a recurrent pattern:

“If the tool is good at a certain thing you will normally rely on the tool an awful lot more for that particular thing, and you will normally re-arrange the way you do things so that you will use that tool fully before moving onto another aspect.” (Chief Web Technologist, OEG)

Fifthly, just as software components can be re-used, the re-use of graphic design elements also speeds up development. Previous research has suggested that graphic designers, being of a “creative” disposition, are not inclined to re-use previous work [10]. The findings of this field study indicate the contrary. This discrepancy can be explained by clarifying what the concept of “re-use” means to a graphic designer. Whereas in software development, a piece of code might be entirely re-used as-is from a previous project, in graphic design a previously used component would constitute a useful “starting point”, but that component would always be uniquely re-worked to some extent. From their background tuition in art, graphic designers are trained to seek and synthesise elements and styles from various sources. As such, the re-use of concepts is actually a normal part of their work:

“I’ve built up a database of sites that I like. When I come across something, I’ll take a screen grab and store it as a JPEG, so when I’m looking for an idea for a new site, I can go through maybe 200 images that are from previous sites, something that might have the same colour scheme or be the same sector or whatever, and I could use that then as a starting point for a new design.” (Managing Director, KL Design)

Lastly, a factor which improves developer productivity is know-how and expertise. Baskerville & Pries-Heje [3] identified “dependence on good people” as one of the elements of their “e-Methodology” for rapid Web-based systems development. However, as has been recently debated with regard both to open source software development and agile methods, practices which are reliant upon such a rare commodity as naturally talented programmers are not sustainable. The companies interviewed in this field study were mostly industry leaders who have received numerous professional awards. Successful companies arguably have the advantage of being able to attract better staff, but that success in the first instance is predicated upon the quality of existing staff. While these award-winning companies shared the characteristic that they were all led by highly-motivated and talented individuals, they mostly also share a common concern with the management

of design knowledge. Important types of knowledge mentioned were: application domain knowledge, knowledge about development tools/environments and technical standards, knowledge about design methods and techniques, knowledge of core design principles, and a repertoire of time-efficient work-arounds. Most award-winning companies have mechanisms in place to facilitate and encourage the management of such knowledge (e.g. intranet bulletin boards, “Wiki’s” and “blogs”), with rewards and bonuses accruing to employees who use slack time to acquire and exchange useful knowledge. A number of them also schedule regular time slots for research activity, setting aside normal development work.

3.5. Impact of time pressure on quality: extensive re-use, “pragmatic satisficing”

The planning / requirements definition phase is the most time-intensive part of the Web-based systems design process. These aspects are of critical importance, for as Brooks [5] puts it, “no other part of the work so cripples the resulting system if done wrong”. For large scale systems with many different classes of users, it is preferable to perform user-centred design and to conduct a thorough needs analysis if time permits. However, in some cases, it does not. The time-cost-quality trade-off is a well-known phenomenon in software development. In their study of high-speed software development, Baskerville et al [2] found that “when time drives development, product quality along with performance and cost, assumes second priority”. They labelled this concept “negotiable quality”, by which is meant that “customers and users seem to expect low quality” because of time pressure [3].

In spite of project delivery cycles being of the order of 6 weeks, very little evidence of “negotiated quality” was found in this field study. One possible explanation is that, in the wake of the post-Y2K “dot.com” industry shake-up, the marketplace has become more competitive and users are much less tolerant of unprofessional standards of work. For such vital aspects of system quality as response time, reliability, ease-of-use, visual attractiveness, and security, excellence is a commercial imperative. Given the demand for high-quality, low-cost productions in short timeframes, firms within the Web design industry have adapted their practices to extensively avail of re-usable pre-tested components, making it possible to rapidly develop reliable and robust systems:

“Our content management system is now in phase 1.6. There’s a new version every few months, and it gets fully tested before the new phase goes live. Let’s say you come to me and say ‘I want a Web site’. We have a function on our system where you just press a button, enter some basic parameters into a form, and fill in the HTML templates for the header and the footer. So instantly we can launch a fully proven Web site because the system has already been built. So a lot of the need for testing has been taken out because of the productisation.” (Managing Director, IBUS)

In situations where deadlines are pressing and available resources are tightly constrained, what can happen is that a sub-optimal, but nevertheless acceptably good,

solution is delivered. This practice, as exemplified by the following excerpt, might be called “pragmatic satisficing”, a form of “negotiated quality”:

“What often happens is that the producer of the show has 2 or 3 weeks to get it done, they’re ‘busy, busy, busy’, and then they go ‘Hey, we should have a Web site’. So then at least you get everything all in one lump, but you might only have a few days notice. All we can really do is put out a formalised, *set* kind of a Web site, based on one of our standard templates.” (Web Project Manager, Broadcorp)

It is the combination of acute time and resource constraints that leads to this practice of pragmatic satisficing. At Broadcorp, which is in the media industry, these pressures are especially pronounced; as the Web project manager explained, their concept of “time” is measured in hours, not days, while resources to hand are scarce and fixed. At OEG, resources are similarly limited, but there is not the same commercial imperative to deliver systems quickly so quality is not compromised. At JobsPortal, projects with high urgency are handled by bringing in hired contractors, and/or by re-negotiating the relative priorities of backlogged projects with management (“negotiated schedules”). Elsewhere, the strategies typically employed by commercial Web design agencies to manage this scenario of acute time and resource constraints are either to haul in the client’s expectations (“negotiated scope”) or to outsource some work, and before all of this there is also a widespread practice of factoring buffer time into project estimates. In all of these approaches, quality is the paramount concern and it is not subverted, as Baskerville et al [2] suggest, by time and cost considerations. Even in the worst case scenario, referred to herein as the tactic of “pragmatic satisficing”, a tried-and-tested solution is delivered, albeit it is neither “award winning” (Strata) nor “progressing the boundaries of design” (Broadcorp).

3.6. Requirements clarity: need to “freeze” and sign-off

The clarity and stability of requirements is an age-old issue in systems development, so it was not surprising that our earlier survey had found the most acute problem in Web-based systems design to be the control of scope/feature creep. The other principal challenge revealed by the survey was the preparation of time and cost estimates [9]. Of course, time/cost over-runs and scope creep are intrinsically linked. A major cause of scope creep is that projects often kick-off with a very vague idea of the requirements. As one interviewee explained,

“When you go into a pitch for a job you’ll say ‘Yeah, we can turn this around in 6 weeks’, but at that stage you don’t know what their specific requirements are. So that timeline will be altered after we find out what they want.” (Creative Director, Strata)

Baskerville & Pries-Heje [3] had previously noted this, making the point that “an inability to pre-define system requirements is the central, defining constraint of Internet time development”. The pattern which emerged from interviews is that when clients make initial contact with designers, they typically have little more in mind than a loose set of aspirations. In the initial meeting, these are usually documented in a one- or two-page brief, which also captures such essentials as: project budget, timeframe, main competitors, target audience, and project goals. A de-

tailed requirements specification is then produced by way of negotiation over the course of a number of meetings.

In general, the requirements specification document seems to be predominantly the vision of the designers, wherein they describe what they can do for the client, taking resource allocations into consideration. In all the commercial Web design agencies that were interviewed, the dominant constraint is usually the client's budget. Clients often have naïve expectations at the outset so the sales team, after consultation with the project team, must come to an arrangement as to what can be delivered, for what price, within what timeframe. Though most of the functional requirements are typically standard and can therefore be readily described and costed "à la carte", the bespoke elements take time to specify, as does a considered analysis of the fine details of the overall package including the "non-functional" requirements (usability, accessibility, security, performance levels, etc.). As initially revealed by the survey and later substantiated by follow-up interviews, it is common practice to produce and "freeze" a detailed requirements specification before commencing full scale production.

These requirements specifications are essentially pseudo-legal bargaining chips that are used to control creep, cost, and scheduling, but they also serve a defensive purpose whereby project managers can insulate themselves from political fall-out by insisting upon a clear signed-off brief:

"I'm responsible for delivering projects on time. If anything goes wrong, I'm answerable for my team. My team has to deliver to me, and I am answerable to the top people ... If there is a communication gap between the users and the developers, the project doesn't go well! That's why we try to get it signed off as much as possible. If it is not signed off, we could be in trouble." (Web Project Manager, JobsPortal)

3.7. Streamlined processes and procedures to support a sustainable pace

Jayaratna et al [8] make the point that "methodologies are time ordering mechanisms". Where project cycles are customarily tight, it seems reasonable to expect that time-efficient working procedures would be in place. In the survey [9], it was found that processes tend to be more formalised and explicit in Web Development companies than in traditional IT/Software Development companies. A possible explanation for this which emerged in the interviews is the sales-driven high-speed nature of work practices in Web design agencies:

"You have to streamline how you do things. You have to build processes, put them in place, and just *follow* them ... So when a Web design project comes in, you know exactly what to do, you take it, and you go bang-bang-bang-bang." (Web Editor, OEG)

Consistent with the results of Baskerville & Pries-Heje's [3] study, it was found that much Web design/development work is done in parallel, similar to the notion of "concurrent engineering" in manufacturing, thereby speeding up development times.

In high-speed development environments, an important issue is how to support a sustainable pace whereby the project team consistently manages to deliver short-

cycle projects on time without their stamina being diminished. Here again, streamlined work procedures can be beneficial, as exemplified by the experiences at Bizweb:

“When you’re doing makeshift things you end up reinventing the wheel, which isn’t cost effective or productive. And then that leads to you overworking your staff, morale is low, and you can’t motivate them. If you ask some people how the company has improved in the past 2 years, they’ll say ‘Whoa, in the old days we had to ...’, it all ties in with standardising things and putting procedures in place.” (QA Manager, Bizweb)

3.8. Project management: collectively determined schedules, cohesive teams

Where project timeframes are short, it is important that time estimates are accurate because small over-runs, in relative terms, are more significant. In the survey, it was found that, despite the intrinsic difficulties in preparing time and cost estimates, Web project managers are faring quite well.

In their study of Web design practices, Rodriguez-Garcia & Harrison [13] found that project management estimates are most commonly formed by analogy and judgement, and also that most organisations collect time/effort metrics as timesheets for billing. The findings of this field study reveal a similar picture. Nearly all of the commercial Web design agencies/development houses visited spoke of the use of job management systems wherein *collectively* agreed estimates are recorded and change requests are logged. By asking developers to set their own schedules, those schedules are more likely to be reasonable, therefore facilitating a sustainable pace. This practice contributes to enhanced staff morale because it reduces the need for overtime and also because the development team are empowered to determine and take personal responsibility for their workloads.

The opposite effect was also found to exist. As explained by two different interviewees, each speaking of practices in former places of employment, project time estimates which are dictatorially imposed rather than democratically negotiated can lead to resentment and coercion, and ultimately to staff turnover:

“Of all the issues, – people are even leaving their jobs because of it, – the most serious one is down to project management. Let’s say, the job comes in, the project manager talks with the client, the sales guy signs the deal. That’s the timeline. It’s agreed with the Web development team, and everyone’s happy with that. An issue arises with the customer where the project manager changes the dates according to the plan without even discussing with the Web team, and that causes mayhem. Because it’s a very tight process, even if you move things out by a day or two that will affect other jobs, and it just becomes a mess.”

“The sales people and the project managers can cause *hell* for developers by over-promising, by not understanding what’s involved and not consulting with the developers on the project timelines ... And unfortunately what seems to happen is that programmers roll over, they work all hours to meet these deadlines, and that’s not noticed by management, but when they complain that they’re overworked, the management typically just say ‘Well, you’ve done it before, you mustn’t really mean it!’.”

Whereas Web design agencies prepare detailed timelines and breakdowns, – such as at Webcorp where the commercial manager explained that a typical project plan if printed out would cover one whole side wall of his office, – a somewhat different picture emerged for the in-house development teams. At Broadcorp and Job-sPortal, “elegant” project plans are not drawn up on such a grand scale simply because relative priorities are driven by the organisation’s business imperatives which can change dramatically from day-to-day, so they must be very flexible and responsive. Web design agencies typically operate within comfort zones, making allowance for a certain amount of slack. As laid out in project task breakdowns and agreed work schedules, individual team members can focus their attention on specific projects for dedicated blocks of time. In contrast, in-house development teams usually find themselves facing multiple urgent deadlines with little room to manoeuvre. Interestingly, the Web project manager at Broadcorp used the metaphor of a “flight controller” to describe how he copes with this challenge:

“It’s coming to the stage where it’s just like landing planes. There’s 10 or 15 projects flying around up there, so you just pick the one that needs to be done.” (Web Project Manager, Broadcorp)

3.9. Closeness of relationship with client project team

Rather like the agile methods concept of an “on-site customer”, it was found that a prerequisite for rapid Web-based systems development is a close relationship with the client, as well as the unity and commitment of the client organisation’s project team. In the absence of any of these factors, communication becomes protracted and jumbled, inevitably causing project delivery times to slip:

“If a client sits on something for a couple of days, the project will be delayed. Schedules are like concertinas all the time. We can tell a client that we’ll have this done for you in 6 weeks, but that often gets pushed out because we’re waiting for something to come back ... There’s one project we’re working on, it’s been stalled now for a year because on the client side they’ve no real project manager. They have about 15 people looking after the various elements of the Web site so no-one knows who’s doing what and they’ve all got other more important things to do.” (Creative Director, Strata)

This pattern of procrastination and aimlessness is exacerbated where the dreaded phenomenon of “design by committee” presents itself, as one interviewee explained:

“The minute you’re in-house, you’re prone to more politics. In my last job, I had one boss who made all the decisions when it came to anything. That doesn’t happen in here, it’s all committee-based. Sometimes it’s extremely frustrating, - when you’re pushing forward with a project, you might have to stop for three weeks or longer just to wait for someone to take a look. It can really disrupt and is probably slowing things down ten-fold.”

3.10. “Working software over documentation”

“When you engage with clients they want to see something, they want to get programming started. But we’ll say ‘now hold on, the most important job is the definition

document, you agree the plans with a builder who builds a house, so we're not even going near the bricks until this is done'." (Managing Director, IBUS)

Though in most cases, as in the above excerpt, interviewees are firmly of the opinion that jumping straight to coding without a robust design specification is ill-advised, there is also a widely held view that the production of documentation is simply a means to an end and that beyond a certain point of "good enough" it becomes a resource-sapping, non-value-adding, unnecessary activity. Thus, the value of light, essential documentation is accepted, but given the imperative to turn projects around quickly, prototypes and working software is developed as early as possible, and refactored and evolved as required depending on change requests:

"Not necessarily straight away, but I think as early in the process as possible you should start coding. If you have a good idea of what you're doing, say you've got 70% or 80% of the requirements tied down, I'd be inclined to move on. I suppose that's more to do with my background as a software engineer, I would be itching to get into the actual implementation of it as early as possible, I'm not a huge fan of too much paperwork although I think it is important to capture the gist of the functional specification." (Managing Director, W3M).

4. Conclusions

Consistent with the previous work of Baskerville & Pries-Heje [3,4], this study found, as one would expect, that time pressure is the central determinant of design practices. However, there are discrepancies between this research and that of Baskerville & Pries-Heje, most notably with their finding that developers may resort to the practices of "coding your way out" and "negotiated quality" because of the pressures of high-speed development environments. Whereas in Baskerville & Pries-Heje's study such practices were endemic, in this research hardly any such incidents were discovered. This can be explained in a number of ways. Firstly, the interviewed companies were mostly award-winners, a likely indicator that they make special efforts to strive for excellence and quality. Secondly, the marketplace has become more competitive in recent years and users are much less tolerant of unprofessional standards of work, meaning that expectation levels have risen. Thirdly, the use of pre-fabricated "productised" solutions that are already fully tested means that robust systems can be rapidly delivered without compromising cost or quality. Even in the worst case scenario for a development team, where they face the dreaded "backs-to-the-wall" combination of acute time and resource constraints, a tactic herein coined as "pragmatic satisficing" is engaged, meaning that a tried-and-tested solution is re-used, albeit it may not be the best possible outcome.

Given the high-speed nature of Web-based systems development, the emphasis of development practices is very much on agility, speed, efficiency and productivity. Streamlined processes are necessary in order to maximise throughput, and also to sustain a continual pace by eradicating the need for ongoing overtime (which has fatiguing and demoralising effects). Interestingly, many of the Web developers interviewed have evolved practices that are markedly similar to those of the "ag-

ile” methods family, such as: collective code ownership; an emphasis on simplicity; the use of regular informal team briefings; insistence on a close working relationship with the client; the pursuit of continuous process improvement through reflective evaluation; and a general emphasis on people, communication, and working software over processes, documentation, and adherence to a plan.

References

- [1] Barry, C. & Lang, M. (2003) A Comparison of "Traditional" and Multimedia Information Systems Development Practices. *Information and Software Technology*. 45(4), 217-227.
- [2] Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B. & Slaughter, S. (2001) How Internet Software Companies Negotiate Quality. *IEEE Computer*. 34(5) May, 51-57.
- [3] Baskerville, R. & Pries-Heje, J. (2001) Racing the E-Bomb: How the Internet is Redefining Information Systems Development Methodology. In Russo, N. L. et al. (eds), *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective*. IFIP WG8.2 Conference, Boise, Idaho, USA, 27-29 July 2001, pp. 49-68. Boston: Kluwer Academic Publishers.
- [4] Baskerville, R. & Pries-Heje, J. (2004) Short cycle time systems development. *Information Systems Journal*. 14(3), 237-264.
- [5] Brooks, F. P. (1987) No Silver Bullet / Essence and Accidents of Software Engineering. *IEEE Computer*. 20(4) April, 10-18.
- [6] Glaser, B. G. & Strauss, A. L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*. New York: Aldine de Gruyter.
- [7] Glass, R. L. (2001) Who's Right in the Web Development Debate? *Cutter IT Journal*. 14(7) July, 6-10.
- [8] Jayaratna, N., Holt, P. & Wood-Harper, T. (1999) Criteria for Methodology Choice in Information Systems Development. *Journal of Contemporary Issues in Business and Government*. 5(2), 30-34.
- [9] Lang, M. & Fitzgerald, B. (2005) Hypermedia Systems Development Practice: A Survey. *IEEE Software*. 20(2) March-April, 68-75.
- [10] Linden, T. & Cybulski, J. (2003) Capturing Best Practices in Web Development. In Isaías, P. & Karmakar, N. (eds), *IADIS International WWW/Internet 2003 Conference*, Algarve, Portugal, November 5-8, Vol. 1, pp. 427-434. IADIS Press. ISBN 972-98947-1-X.
- [11] Locke, K. (2001) *Grounded Theory in Management Research*. London: Sage.
- [12] Miles, M. B. & Huberman, A. M. (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd Edition. Thousand Oaks, CA: Sage.
- [13] Rodriguez-Garcia, D. & Harrison, R. (2000) Practitioners views on Web development: an industrial survey by semi-structured interviews. In 13th International Conference on Software and Systems Engineering and their Applications (ICSSEA 2000), Paris, France, December 5-8. CNAM.
- [14] Strauss, A. & Corbin, J. (1998) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd Edition. Thousand Oaks, CA: Sage. ISBN 0-8039-5939-7.
- [15] Suchman, L. A. (1987) *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge: Cambridge University Press.
- [16] Thomas, D. (1998) Web Time Software Development. *Software Development Magazine*. 6(10) October, 78-80.