Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

International Journal of Project Management 23 (2005) 65–74

INTERNATIONAL JOURNAL OF PROJECT MANAGEMENT

www.elsevier.com/locate/ijproman

A helical model for managing innovative product and service initiatives in volatile commercial environments

Eric Deakins [*], Stuart Dillon

Department of Management Systems, Waikato Management School, University of Waikato, Private Bag 3105, Hamilton, New Zealand

Received 19 November 2002; received in revised form 7 March 2003; accepted 11 May 2004

Abstract

Organisations operating in highly competitive and volatile commercial environments require the ability to cost-effectively build market share quickly, via rapid time-to-market of repeated high quality and innovative solutions. This has created the need for an approach to product development that encourages experimentation and mass customisation, yet also allows just in time (JIT) delivery of cutting edge solutions. This paper reports on field research with a dotcom start-up company that evolved a traditional spiral (software) development methodology into a generic management tool suitable for the design, production, and marketing of innovative, high-quality e-commerce products and services. Advantages and limitations of the method are presented and its generalisability to a range of projects is illustrated, with the aid of Action Learning undertaken within an academic setting.

© 2004 Elsevier Ltd and IPMA. All rights reserved.

Keywords: Managing projects; Technology; Innovation

1. Introduction

The key to success for contemporary organisations competing in a virtual marketplace is their ability to cost-effectively build market share quickly, via rapid time-to-market of repeated high quality and innovative solutions that exceed the expectations of every customer. Tensions between such factors is heightened in the pressure-cooker environments typically experienced by developer teams in dotcom start-ups and smaller high-tech companies – creating the need for an approach to e-commerce product and service development that encourages experimentation and mass customisation, yet also allows just in time (JIT) delivery of high quality, cutting edge solutions.

This paper begins with a brief description of the contemporary e-commerce development environment, and argues the need for a new approach for managing e-commerce product and service initiatives. Field research with a dotcom start-up company is outlined, including how a traditional 'spiral' model chosen for (software) product development was evolved into a modified spiral model that provided a superior fit between the conflicting needs for creative, high-quality solutions to be developed under rapidly changing conditions and under severe resource and time constraints. The new 'Helical' Model is then presented, together with its claimed advantages and limitations as a generic management tool for developing a wide range of products and services in volatile, unpredictable environments. Finally, limited support for the model is gained with the aid of Action Learning undertaken within an academic setting.

2. The contemporary product (service) development environment

While relatively few organisations have made significant inroads into selling their products over the Internet, many are deploying online customer services or are using Internet-based technologies to improve supply chain efficiencies. E-commerce developer teams need an effective approach for managing new initiatives, preferably one that is simple to administer and is cognisant of

[*] Corresponding author. Tel.: +64-7-838-4561/4565; fax: +64-7-838-4270.

E-mail addresses: edeakins@waikato.ac.nz (E. Deakins), stuart@waikato.ac.nz (S. Dillon).

0263-7863/$30.00 © 2004 Elsevier Ltd and IPMA. All rights reserved.
doi:10.1016/j.ijproman.2004.05.007

the limited available resources yet also encourages the creation of innovative and high-quality products (services) on time and within budget. This need is likely to increase as mobile commerce (m-commerce) makes possible the next wave of new products and services, all competing for consumer attention.

3. The need for a new method for managing new product and service initiatives

Projects relying on technology for their success have long had a poor track record and the authors support the view of others [1,2] that a new method is needed for developer teams that operate in the e-commerce setting, because:

- The product (service) development processes employed in an e-commerce environment are fundamentally different to those used when developing 'products' for in-house use.

One of the most significant phases in truly innovative product (service) development is that of content creation. Whereas for company systems this activity may be limited to drawing out user needs from focus groups, in the e-commerce environment the developer team must also elicit external customer expectations from consideration of known competitor products and services plus a vast range of external factors and trends such as global issues of taste and socio-cultural norms, product market positioning, and associated budgets [3]. The relationship between developers and customers has also changed with customers increasingly willing to assist with the development effort [4].

- The product (service) development processes employed for an e-commerce environment need to have a significantly different emphasis to give competitive advantage.

According to Gray and Larsen [5]: "Strategy is implemented through projects and every project should have a clear link to organizational strategy." It follows that the developer team must ensure that its product and service offerings are designed to achieve goals that are strategically aligned with the overall aims of their organisation. Thus, the team needs to be intimately knowledgeable about its new product or service and cognisant of, if not actually capable of handling, end-to-end development. This marks a departure from practice in many organisations where largely unconnected functional divisions handle the 'separate' aspects of adding value to the product (service) [6].

- The product (service) development processes employed for an e-commerce environment require a much wider skill set.

Whereas traditional product (service) developer teams tend to consist of functionally focused individuals, small high-tech companies benefit from having multi-skilled team members having a broad competence across the functional areas of marketing, communication, finance and operations. The ability to recognise opportunities afforded by converging technologies is also desirable [7,8] as is an appreciation of Systems Thinking, to understand how decisions made in one decision space have the potential to impact adjacent decision spaces [9].

- Project management in an e-commerce environment has a significantly different emphasis.

Project managers in a traditional transaction-processing domain have often progressed through the developer ranks and tend to view issues through that lens. In the dotcom environment they are likely to be from a more diverse background – perhaps from graphic design or a functional specialism such as marketing. The models, priorities and assumptions they bring to project management may therefore be substantially different and more effective/disruptive.

- Project teams in an e-commerce environment are largely cross-disciplinary.

Related to the previous point, dotcom developer team personnel may have substantially different skills and backgrounds [10]. Managing such a diverse team may be even more difficult than managing a more traditional group of business analysts and programmers.

- A highly competitive e-commerce environment gives rise to new, or significantly enhanced, development requirements.

Web technologies bring new opportunities for developers to be creative and to delight their customers via direct enhancement of information-based products and services, on-line maintenance and support, and rapport building. Consequently, mass-market competitor-driven products and services require a development approach that *promotes* design experimentation, JIT delivery practices, and mass customisation, as well as efficiency of cost and effort with no quality penalty [3].

It may be concluded from the above that the demands made of dotcom product (service) developer teams are many and varied and their offerings are expected to be relevant to the end-customer, innovative, good value and of high perceived quality. A lack of resources in smaller companies may also mean that developer teams have to handle their own marketing, distribution and support of the finished product via (evolving) e-commerce channels. Such demands call for truly creative and multi-skilled developer teams and a project management approach that provides support to them as they operate within an unpredictable and volatile commercial environment.

Given the plethora of project management methods it may reasonably be asked whether there is a need for yet another one for the e-commerce environment; could an existing method be used or are those available not being implemented properly? While a lack of the literature makes it difficult to answer the question of inappropriate implementation [11–13], there exists the widespread view that traditional development methods are cumbersome and not appropriate in today's rapidly changing and volatile development environments and they need to be either modified or replaced, e.g. [14,15]. Indeed, it is claimed that methodologies designed to support WWW-based applications development are soon left behind [16,17] as companies are forced to adopt the latest web technologies to gain a strategic advantage over competitors. This view is backed by field researchers observing time and again that no specific development methodology is used when Internet-based information systems are being developed, with such development often described as being ad hoc and dominated by the challenges of working with new technology [12,13,18]. This tendency for frantic and ad hoc [14,15] applications development is also accompanied by a lack of guidelines and documentation [13].

This discussion indicates that a new market-driven method for managing e-commerce initiatives is needed, and that an ideal method for the e-commerce development environment would encourage:

- Rapid development of high-quality products and services.
- Innovative and creative solutions via experimentation.
- Continual improvement to product specifications via high levels of customer feedback and responsiveness to the external environment.

The following section describes fieldwork that led to a new method for managing e-commerce initiatives.

4. Case study: product development in a virtual organisation

4.1. Task description and framing

There has been a widespread call for fieldwork to be included as part of the development of information system development methods (ISDMs) citing claims that the approach provides an effective method of data collection, especially for understanding the many complex human issues that are often involved [19–22]. At a time when the authors were questioning the ability of traditional ISDMs to meet the demands of the emerging e-commerce environment, they became aware that a dotcom start-up company was intending to use a traditional ISDM to create an innovative software product that would be marketed and supported via the Internet. The company had already decided on its preferred development methodology and agreed to let the authors observe the project.

The fieldwork study centred on a small dotcom start-up that was formed in 1998. The project was begun in November 1998 in anticipation of a product release date at the end of the second quarter in 1999. The new product was to be a self-help Y2K assessment system that would enable small and mid-sized businesses to assess their Y2K exposure and to track contingency plans without the need for expensive consultants. Time constraints were very severe given the intended use for the product. The developer team comprised one full-time member and three part-time members with skills in (traditional) project management, decision support system design, and website design. No e-commerce or product marketing skills were represented in the team and it was intended from the outset that specialised software coding skills would be purchased at an appropriate time. Given these constraints, a traditional project management software package was utilised to plan and communicate the future project. Fig. 1 indicates the highly concurrent nature of the expected development activities going into the project.

Essentially, the dotcom company wished to develop an innovative decision support system. Consequently, before formal systems design commenced the ISDM literature was reviewed for candidate methodologies and tools to guide the activities of the software product developer team.

4.2. Software development methodologies considered

Several developer team members had experience in traditional information systems development and this appeared to strongly determine the choice of methodology. With the aid of the ISDM review it was judged that the sequential nature of grand design lifecycle models would significantly reduce the team's ability to incorporate changes in requirements and might also cause problems of expense, complexity and, most critically, time consumed [23]. The team judged that a spiral model [24] with its emphasis on refinement of user requirements via prototyping, would have the best chance of delivering an innovative software product, since spiral models essentially represent several iterations of the lifecycle model and provide for a risk analysis at every iteration. It was intended that the first design 'round' would consist of user and system requirements determination, to be followed on the next round by software requirements, architectural design... and so on. Consequently, once the broad user requirements had been agreed via focus groups of Y2K specialists, team members were assigned separate tasks as indicated by the Gantt chart. The self-help requirement of the final product meant that considerable emphasis was placed on the user interface using a rapid application development (RAD) regime based on prototyping.

It is interesting that the developer team did not choose to adopt a proprietary RAD method, such as the (now well-known) dynamic systems development

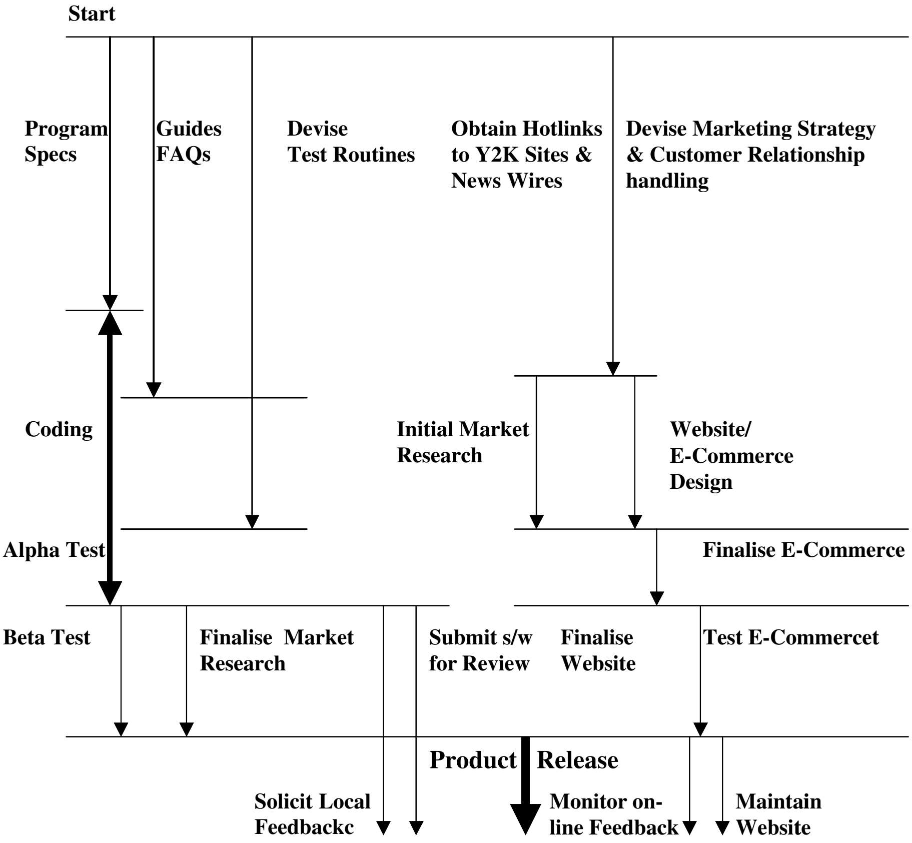

Fig. 1. Project Gantt chart.

method (DSDM) – which is claimed to be especially suited to e-business projects with tight deadlines and a need to respond to rapid changes in the marketplace [25]. Questioning revealed that such an all-encompassing method (DSDM has seven distinct phases) was unlikely to be adopted due to the need for competence: "…typically requiring awareness and briefing sessions, skilled facilitation resource, help with team selection, help with project selection, enhancement of application development standards, general DSDM project set up and planning and team mentoring" [25]. While the members of the project team were experienced in traditional information systems design, they lacked experience in designing for volatile e-commerce environments so attempts were made to adhere closely to the requirements of the chosen spiral methodology. This action is in line with earlier studies that have noted developer teams tending to closely follow a particular methodology in areas where they lack experience, while experienced teams prefer either to not use a 'cumbersome' standard method or to use one that has been heavily modified for the task in hand [12,17,18]. Other all-inclusive software development methods that had just become widely available, but of which the developer team was not aware, include those by McConnell [26] and Royce [27].

4.3. Project narrative

Microsoft PowerPoint™ was used to work up the user interface in the form of a slideshow. The skills available in the team and the nature of the project meant that, in the first instance, the 'users' were the developers. During the course of these early software design activities the conceptual design of a company website dedicated to the product was also initiated. In addition to product features and specifications, this also involved consideration of broad marketing strategies, product promotion to a global audience, on-line distribution, and electronic payment systems.

Through the early project stages the evolutionary spiral model appeared to serve the team reasonably well and, provided that only small changes to product specifications were required, these could be accommodated using the chosen project management tool. However, it soon became evident that there would probably be many late (and potentially large) changes required, both to the software product and to the supporting website. This was in part because trends in software and e-commerce had moved on since project initiation. For example, the trend to tighter integration of external databases, tighter integration of web features, advances in online customer support services, and advances in security measures for online transactions. Commensurate heightening of customer expectations following the release of competitors' products made it desirable to incorporate such web-enabled technologies into the final product. These major shifts in specifications were, in the main, not foreseeable by the project manager.

The inherent need for shortest time to market, to achieve early 'share of mind', soon came into conflict with the need to incorporate the late design changes at short notice. Such problems came to a head when focus group feedback confirmed the need for the changes. To address such issues in a design sense it was necessary to 'back track' to earlier conceptual stages when such design features may have been tentatively explored but not followed through, for a variety of reasons.

It was also deemed necessary to release the highest quality software product possible to minimise the risk of a potentially financially crippling global recall, or the need for a product patch (with its attendant adverse publicity). Again, this market-driven decision heightened awareness of the conflicting requirements for highest quality with an early product release date.

Downstream a new challenge began to emerge regarding the overall cohesiveness of the project activities. The wide variety of activities and the limited skills available in the team meant that some design components were moving ahead very quickly, with others hardly progressing. In addition, although the design team was being very creative and were performing well individually, the project began to lose focus. It was felt that each design component was independently 'spiralling' towards its own unique destination instead of contributing to a coherent product as intended by the evolutionary design spiral methodology. Thus, a major management challenge that emerged was the need to more strongly coordinate progress across the various (iterating) components of software design, product promotion and marketing, etc.

By this time the limitations of the evolutionary spiral model were becoming very apparent. Although the overall cyclical nature of the spiral ISDM for assessing risks encouraged the team members to constantly define the best way forward overall, and to refine the *overall* design concept, the need to revisit earlier *component design* 'solutions' – in response to late changes in product specifications, was not able to be so easily accommodated by the project manager. Consequently, a more flexible process began to emerge.

Cyclical prototyping methods that had been used extensively in the early stages for interface design had also been found invaluable for much of the *non-technical* development activities. For example, work on the marketing plan that had begun at project initiation, to test the original concept, was sometimes put aside for weeks at a time. This inactivity would be followed by periods of refinement and adjustment as product features became finalised and as time permitted. As the product release date approached, the 'soft-coded' marketing plans were dusted off and refined some more until the final plans became 'solidified'. This cyclical mode of development was similarly utilised for website design, product testing, and for the building of help files and other support mechanisms. The programmer also chose to follow a similar approach; rather than coding a module or program component to completion, basic functionality was installed and approval then sought from the system architect before advancing further. This was found to all but eliminate misinterpretations of the *evolving* program specifications.

By the late stages of the project a new product development method had emerged that largely overcame the limitations of the original. As a result of the time needed to fine-tune the design of the user interface, strict deadlines had to be enforced to keep the project on schedule. Also, careful planning was required so that concurrent activities with different start dates could be completed by a common due date. This was largely necessary because, at any time, there would be several distinct developmental activities being actioned, each with its own cyclical design process involving prototyping. Resources that had originally been allocated to distinct activities were being fluidly assigned according to the importance of each at that particular point in time. This suited the designers who could be provided with a variety of work. Not only did this contribute to high levels of productivity, it also appeared to encourage creativity and enthusiasm.

The software product was launched soon after its planned release date and needed no rework or alteration. Clearly, the development method that evolved was critical to the success of the project. The constraints imposed on the young dotcom company had forced a new product development approach to emerge that enabled concurrent, cyclical design activities to be changed at short notice in response to emerging marketplace requirements. It became apparent to the authors that such a generic approach would be useful for managing a wider range of new-technology projects, not just those concerned with software product design.

5. Description of the Helical Model

This section presents the approach to e-commerce product (service) development that evolved while the authors were working closely with the young dotcom organisation. Fig. 2 illustrates the so-called Helical Model and shows a design component *Design Spiral* that is enclosed by its own *Decision Space*. The Decision Space is the domain of decision making for the developer team, which expands or contracts as the project progresses in response to changing internal and external

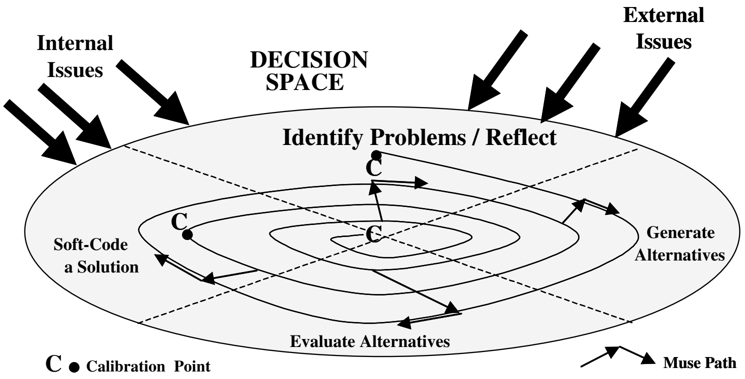

Fig. 2. Helical Model – component decision space.

issues (problems and opportunities). The developer team has complete freedom (is empowered), subject to time and resource constraints, to operate anywhere within that space as they develop a major design component such as the design concept, market assessment, associated product marketing, and so on. Ideally, the composition of the developer team will provide a balance of disciplines and skills and team members will be enabled to effectively work and communicate together.

Once the baseline requirements have been agreed, developer team activities will tend to spiral inwards towards the 'final solution', transiting the four important design phases (quadrants) of:

- *Identify problems/reflect* – identification of the immediate problem set to be addressed, or opportunity to be maximised. This phase ends when the developer team is convinced that the identified problem/opportunity set is both important and deserving of immediate attention. This is also referred to as the 'Reflect' quadrant because an interim design component 'solution' will often be temporarily set aside to allow time for reflection by the design team, or for developments to become aligned in a related design spiral.
- *Generate alternatives* – alternative solutions generated as the result of a formal requirements determination. This phase ends when the developer team is convinced that the design alternatives generated cover the range of achievable solutions (solutions which usually stretch the team when developing innovative products for a commercial setting).
- *Evaluate alternatives* – alternatives evaluated according to relevant decision space criteria, including a formal (technical and market) risk analysis for each alternative. This typically identifies a list of possible risk events and then uses the probability of the risk event (P_e), the probability of (undesirable) impact (P_i), and the (potential) lost time (L_T) to estimate the expected time loss for the risk event ($L_e = P_e \times P_i \times L_T$). As well as helping achieve team consensus in the selection of the best design alternative, this information together with knowledge about each risk event driver and each impact driver, enables the team to prioritise risk events and formulate effective prevention and contingency plans for dealing with them – after Smith [28].
- *Soft-code a solution* – solutions remain soft-coded (remain 'live') until either the project ends naturally within time and budget – when the last solution becomes the hard-coded solution used in the final product, its distribution, and promotion, or until the project team runs out of time due to an approaching hard deadline or cash limit, when the last solution becomes the hard-coded solution.

It is rare for the design specification to be known ahead of time and even rarer for customers to not change their requirements [29]. Thus, each preceding phase informs the next phase and rapid prototyping is used to generate and quickly evaluate a business case for proposed design alternatives *throughout* the project – even at the *fuzzy front end* of the development process ahead of the concept design where huge potential time savings and extra profits can be gained [30]. Constant appraisal and reappraisal of emerging design components and their fit with the other components occurs in all quadrants. The actual number of design iterations depends on the nature of the problem being addressed, since the desire to produce an innovative, high-quality product is tempered by the need for a rapid time to market. Project team activities iterate around the same four basic phases, refining the solution on each cycle and typically achieving less progress per unit of time.

Fig. 2 also shows so-called 'muse paths', which are discontinuities in the (intended) spiral path of progress towards the completed product or service. The number and size of these paths is a reflection of the volatility and changeability of the development environment, in which discontinuities are the result of inevitable and often unforeseen events that occur in adjacent decision spaces or in the internal and external environments. By keeping clear and careful records of their development activities the team is able to backtrack to earlier stages of product development (within the same quadrant) so as to quickly resume a fresh line of investigation, should the need arise. This is an important feature of the model and one that has been recognised by earlier authors [31].

Calibration points are defined as predetermined (programmed) points in time at which progress is assessed across the various component activities by the project manager and any needed resource adjustments made with the aid of traditional project management software. Given the often highly creative nature of the work it is intentional that the project is not over-managed while being cognisant of the need for:

- An early set of satisficed [32] solutions.
- Overall project terminal points and milestones to be met.
- Design decisions to be recorded to enable backtracking to earlier decisions and activities via muse paths.
- The effect of breakthrough developments, which occur in one major design component, to be immediately fed into its related decision spaces, if judged appropriate.

From the earliest stages, the project manager must identify any *critical* components of the design which, for information-rich products and web supported services, usually means factors or features that could 'kill' the product on its launch (such as a dated or otherwise poor design), or could require expensive after-sales support or remediation (such as error fixing or support services realignment). Such parts, particularly if they are likely to be difficult to resolve, deserve early attention by the designers. Conversely, non-critical components may justify adopting a satisficed solution – being a solution that is not necessarily the best or the 'optimal' but which does the job. Early identification of such components is vital to allow sufficient time and resource to be assigned to the critical, and often more creative, product/service design elements.

Of course, design team efforts expended on the design spiral activities occur over time and the Helical Model takes its name from the helix formed by the spiral paths that are traced out within the Project Decision Space, Fig. 3. This figure indicates the trajectory of a simple project containing three major activities. It shows that two activities will begin concurrently at the top of the project with the third activity scheduled to begin some time later. Each spiral path represents progress being made with one component of the product or service, e.g. website design, software design, and so on. The time axis runs from left to right whereas effort in the design spiral follows the spiral path. More design spiral iterations per unit time, indicative of greater design effort, gives a finer helical 'thread'. Conversely, the middle spiral shows a period of time during which work was suspended on that design component, perhaps to concentrate effort into the newly activated (lowest) spiral. Although the developer team is seeking an optimal solution for the product or service, their initial aim is to achieve a satisficed solution since without this, by definition, a product or service does not exist.

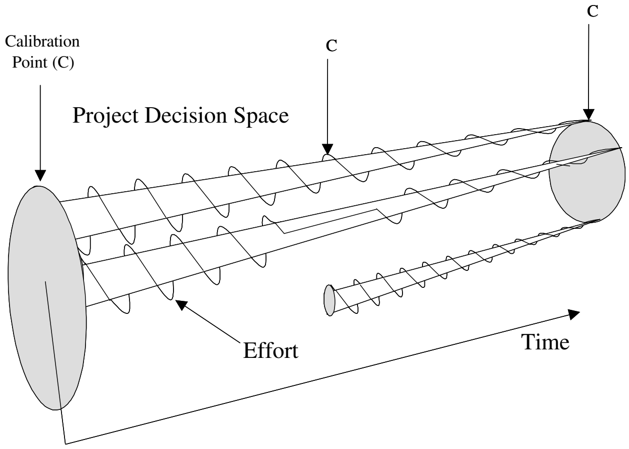

Fig. 3. Helical Model – action trajectories.

The model as presented is a relatively non-prescriptive conceptual approach that has the virtue of administrative simplicity. It does, however, have the flexibility to incorporate more specific and detailed development activities that may be appropriate to the specific design components under consideration. In contrast to the spiral model originally used by the developer team, the Helical Model recognises that a separate design spiral exists for each major design component, that learning occurs continuously, and that each spiral has its own associated decision space concerned with detailed design decisions (the complete set of such subordinate decision spaces comprises the overall Project Decision Space). The need for the overall direction of the project to be aligned with the organisation's strategic direction also cannot be over-stressed if the final product or service is to be effective in pure business terms [33].

6. Advantages and limitations of the Helical Model

The model described is cognisant of the nature of contemporary product (service) development in an information-rich e-commerce environment, -being often random, ad hoc, and visionary [34]. Further, as advocated by several authors, e.g. [35,36], it is suited to various contexts of use and is framed at a high level of granularity, prescribing broad guidelines rather than complex over-intellectualised lists of tasks. The main advantages claimed for the Helical Method when used to develop innovative products and services are:

1. It is rapid; its main aim being to achieve a satisficed solution that meets baseline requirements. Layers of refinement and quality are then iteratively and

Table 1
The value of ISDMs for product development in dotcom environments

	Lifecycle models	Spiral models	Helical Model
Rapid development	Low	Low	Medium
Innovative and creative solutions	Low	Medium	High
Continual improvement to product specifications	Low/medium	Medium/high	High

incrementally added to a set of soft-coded *product/service components*, making possible JIT delivery of last minute design changes.
2. It is completely scaleable; in principle there is no limit to the number of design components that can be accommodated. Similarly, design components can be deconstructed into sub-components, each having an associated decision space.
3. Creativity via experimentation is encouraged since all changes are reversible and no requirements are frozen. Team members are empowered (within the context determined at the outset by the high level strategic plan) to make decisions subject to a collaborative, co-operative approach being encouraged.
4. A high level of customer feedback and responsiveness to the internal and external environment is incorporated via prototyping and a propensity to seek out new developments occurring in the marketplace; testing is integrated throughout the development cycle.

The Helical Model is also broadly in line with the commercial systems development principles proposed by Howard [37], having the following additional advantages over existing methods for managers of innovative products and services in volatile, unpredictable environments:

5. It provides a lens through which to view especially creative and innovative projects
6. Methodologically, it has intuitive appeal as well as visual appeal that helps the project manager and developer team appreciate the nature of the processes occurring across and within the team(s)
7. The focus is on the end product and on opportunities for refining the business case rather than on time to 'complete' the design activity. This lessens the need for close project control
8. Non-critical 'satisficing' elements are explicitly identified, making project planning to a clearly defined set of objectives much easier; the distinction between satisficed and optimal solutions encourages the reconciliation of business and design tensions

The Helical Model has the following major advantages for the members of the developer team

9. Freedom to create innovative solutions subject only for the need to produce an initial set of satisficed non-critical design components. Optimal solutions for the identified critical components must also be produced, constrained only by the available time and resources remaining for the project
10. (Relative) freedom from what is often perceived to be 'petty' administration and restrictions on creative freedom. It is also simple for small developer teams to administer.

Table 1 tentatively compares the value of the Helical Model against some traditional ISDMs for product (service) development in dotcom environments, indicating some significant improvements.

Recognised limitations of the model include:

1. The desire for a hands-off project management approach that encourages creativity has the potential to encourage 'gold plating' of a design. By itself the Helical Model lacks internal control elements but it can be readily combined with traditional project management tools
2. It may be difficult for the project manager to plan and manage a large number of concurrent design spirals, especially when the internal or external environment is changing rapidly
3. Developers may underestimate the implications of major design issues; knowing that they can "back track" at any time they may be tempted to postpone difficult decisions.

7. Support for the model

When the Helical Model was first recognised the authors believed that it showed great promise for a wider range of product and service initiatives than software design. It was also recognised that further research would be needed to test and possibly refine the model in other settings. Would it need re-shaping for other situations? With a lack of opportunity to test the model in other workplace settings the opportunity was taken to test the flexibility of the model using Action Learning in an academic setting. Thus, the model's ability to cope with detailed product and service development activities has been tested within the relatively controllable environment of two major classroom assignments, utilising teams of students in the roles of dotcom business owners and developer contractors/project managers, respectively.

The first major assignment involved self-selected teams of students at the University of Waikato creating and refining their ideas for a new e-commerce venture. Each student team would take the part of dotcom business owners and package their ideas for a new online venture into a comprehensive business plan. Records were kept of classroom discussions with the teams and, over a three-year period, thirty student teams completed the Business Planning exercise. To minimise possible bias, feedback was gathered anonymously using third parties to collect and collate the data. The main findings were:

1. The Helical Model helped the design teams to appreciate and describe the nature of the processes occurring within the team; in this regard the diagrammatic descriptions (earlier Figs. 2 and 3) were found to be especially helpful when students were struggling to visualise the task domain
2. The model enabled students to quickly scope a satisficed business plan solution to meet baseline (assignment) requirements; non-critical 'satisficing' elements were explicitly identified and addressed early on, making project planning to a clearly defined set of objectives much easier. Layers of refinement were then iteratively and incrementally added to each section of the plan
3. Creativity via experimentation with ideas was encouraged and team members felt empowered to make decisions and to collaborate and co-operate within their team; high levels of customer feedback (a role played by the lecturer) were incorporated, as were changes arising from developments in the external environment detected via Web searching of 'competitor products'
4. 'Gold plating' was detected in a small number of the teams.

The second assignment involved project management and website development activities. It required students to identify the major issues determining the commercial success of a fledgling e-commerce venture, both from the business plan and the client (class lecturers), and to translate them into a compelling online presence. The main findings over a 2-year period were:

1. Baseline requirements were harder to quantify in advance because website design requires a set of sequential, though overlapping, design activities
2. Creativity via experimentation was definitely encouraged since all changes to design components were reversible using the medium of a storyboard. Similarly, high levels of customer (lecturer) feedback was incorporated via prototyping, and hyperlink and visuals testing was integrated throughout the development cycle; JIT delivery of last minute design changes was possible
3. No gold plating' of designs was detected and it was relatively easy for students to plan and manage the smaller number of concurrent design spirals.

8. Concluding remarks

This paper has argued the need for a new approach for managing innovative product and service initiatives development in volatile e-commerce/m-commerce environments. It described how a new spiral methodology evolved during fieldwork with a developer team that had limited resources. This team found itself operating in an aggressively competitive e-commerce environment and against a hard time constraint. Key lessons learned from the case study include:

- The dotcom development environment is highly volatile and requirements can change rapidly in response to competitor offerings and customer needs; customers are unreliable predictors of their future needs.
- Developer teams need to be multi-skilled and creative, particularly when they are denied access to the functional expertise available in larger, well-established organisations.
- Adaptiveness must take precedence over efficiency; a context created by management must establish what the developer team is to achieve and the constraints on how it does it (and updates this in response to environmental and internal events). Provided the developer teams remain within the parameters defined by the context, the business empowers them to determine for themselves how best to deliver the results for which they are accountable.

The resulting so-called Helical Model offers an improved development methodology that is likely to be of value for new-technology projects in volatile product and service development environments that require:

- Rapid, high-quality development.
- Innovative and creative solutions via experimentation.
- Continual improvement to specifications via high levels of customer feedback and responsiveness to the internal and external environments.
- JIT delivery practices.

To-date, the efficacy of the Helical Model has undergone limited testing using Action Learning within an academic setting. It required no re-shaping for this setting and some support for its application to projects where e-commerce is the development environment, was achieved. At this stage the proposed Helical Model continues to show great promise, although further research is needed to test and refine the model in other real-world settings, perhaps using an Action Research approach.

Acknowledgements

The authors acknowledge the constructive and helpful comments made by the reviewers on an earlier version of this paper.

References

[1] The Standish Group, Charting the Seas of Information Technology, The Standish Group, Dennis, MA, 1994. Available from: http://www.pm2go.com/sample_research/chaos_1994_1.php, accessed 15th December 2003.

[2] Johnson J. Turning chaos into success, reported in Software-Mag.com, 1999. Available from: http://www.softwaremag.com/archive/1999dec/Success.html, Accessed 15th December 2003.

[3] Deakins E, Dillon SM. Managing innovative IS projects in dotcom companies. In: Proceedings of the Pacific Asia Conference on Information Systems (PACIS2000), 10 pages on CD ROM, Hong Kong, June 2000.

[4] Prahalad CK, Ramaswamy V. Co-opting customer competence. Harvard Bus Rev 2000;(January–February).

[5] Gray CF, Larson EW. Project management: the managerial process. New York: Irwin–McGraw-Hill Education; 2000. ISBN 0-07-116317-4, p. 23.

[6] Keen PGW, Knapp M. Business processes. Harvard Business School Press; 1996.

[7] Haeckel SH. Adaptive enterprise: creating and leading sense-and-respond organizations. Harvard Business School Press; 1999. ISBN 0-87584-874-5.

[8] Boar BH. Strategic thinking for information technology. New York: Wiley; 1997.

[9] Forrester JW. Principles of systems. Cambridge: Wright-Allen Press; 1968.

[10] England E, Finney A. Managing multimedia. Reading, MA: Addison-Wesley; 1996.

[11] Wynekoop JL, Russo NL. Systems development methodologies; unanswered questions. J Inform Technol 1995;0:65–73.

[12] Fitzgerald B. The use of systems development methodologies in practice: a field study. Inform Syst J 1997;7(3):201–12.

[13] Paynter J, Pearson M. An analysis of WWW-based information systems. In: Chow WS, editor. Multimedia information systems in practice. Singapore: Springer; 1999. p. 53–63.

[14] Yourdon E. Developing applications for the Internet: advice for the Java generation. Am Program 1996;(December):36–41.

[15] Gellersen H-W, Wicke R, Gaedke M. Web composition: an object-oriented support system for the web engineering lifecycle. Comput Networks ISDN Syst 1997;29:429–37.

[16] Morrell P. Building intranet-based information systems for international companies. Aslib Proc 1997;49(2):27–31.

[17] Wynekoop JL, Russo NL. Studying system development methodologies an examination of research methods. Inform Syst J 1997;1:47–65.

[18] Barry C, Lang M. A survey of multimedia and web development techniques and methodology usage. IEEE Multimedia (Special Issue on Web Engineering) 2001;8(3):52–60.

[19] Orlikowski WJ. CASE tools as organizational change: investigating incremental and radical changes in systems development. MIS Quart 1993;17(3):309–40.

[20] Wastell DG. The Fetish of technique: methodology as a social defence. Inform Syst J 1996;6:25–49.

[21] Fitzgerald B. Formalized systems development methodologies: a critical perspective. Inform Syst J 1996;6:3–23.

[22] Nandhakumar J, Avison DE. The fiction of methodological development: a field study of information systems development. Inform Technol People 1999;12(2):176–91.

[23] Veryard R. What are methodologies good for? Data Process 1985;27(6):9–12.

[24] Ould MA. Strategies for software engineering: the management of risk and quality. London: Wiley; 1990.

[25] DSDM 2002. Available from: http://www.dsdm.org/en/products/ebusiness.asp, accessed 15th December 2003.

[26] McConnell SC. Rapid development: taming wild software schedules. Microsoft Press; 1996. ISBN 1-55615-900-5.

[27] Royce W. Software project management: a unified approach, Addison-Wesley Object Technology Series; 1998, ISBN 0-201-30958-0.

[28] Smith PG. A portrait of risk. PM Network 2003;(April):44–8.

[29] Remertsen DG. Managing the design factory: a product developer's toolkit. New York: Free Press; 1997.

[30] Smith PG. Leverage the cycle time capability of your rapid prototypes. Time Compress Technol (European Edition) 1999;7(2):50–7.

[31] Livari J. Hierarchical spiral model for information system and software development – Part 1: theoretical background. Inform Software Technol 1990;32(6):386–99.

[32] Simon HA. Rational choice and the structure of the environment. In: Simon HA, editor. Models of man. New York: Wiley; 1957.

[33] Deakins E, Makgill HH. The importance of alignment in business process change projects. In: Proceedings of the Ninth Australasian Conference on Information Systems Sydney, Australia; 1998. p. 148–60.

[34] Tong G. Software development process improvement: the forgotten son? World Class Design to Manufacture 1994;1(3): 21–5.

[35] Fitzgerald B. Systems development methodologies: the problem of tenses. Inform Technol People 1999.

[36] Boahene M. Information systems development methodologies: are you being served? In: Proceedings of 10th Australasian Conference on Information Systems, Wellington, New Zealand, December 1–3, 1999.

[37] Howard A. A new RAD-based approach to commercial information systems development: the dynamic system development method. Ind Manage Data Syst 1997;97(5):175–7.