CITRENZ
COMPUTING AND INFORMATION TECHNOLOGY RESEARCH AND EDUCATION NEW ZEALAND

Journal of Applied Computing and Information Technology

ISSN 2230-4398, Volume 18, Issue 1, 2014

Incorporating the NACCQ publications:
Bulletin of Applied Computing and Information Technology, ISSN 1176-4120
Journal of Applied Computing and Information Technology, ISSN 1174-0175

:: **Refereed Article A4:**

Management models in the NZ software industry

Holger Spill
School of Information Management, Victoria University of Wellington, New Zealand

David Mason
School of Information Management, Victoria University of Wellington, New Zealand

Spill, H. & Mason, D. (2014). Management models in the NZ software industry. *Journal of Applied Computing and Information Technology*, 18(1). Retrieved June 2, 2015 from http://www.citrenz.ac.nz/jacit/JACIT1801/2014Spill_SoftwareManagementModels.html

Abstract

This research interviewed eight innovative New Zealand software companies to find out how they manage new product development. It looked at how management used standard techniques of software development to manage product uncertainty through the theoretical lens of the Cyclic Innovation Model. The study found that while there is considerable variation, the management of innovation was largely determined by the level of complexity. Organizations with complex innovative software products had a more iterative software development style, more flexible internal processes and swifter decision-making. Organizations with less complexity in their products tended to use more formal structured approaches. Overall complexity could be inferred with reference to four key factors within the development environment.

1. Introduction

Government and policy makers are interested in how to encourage more innovation, especially in IT. Innovation is seen as a key component of the future wealth and prosperity of New Zealand (Callaghan, 2009).

New Zealand rates well in innovation internationally (INSEAD, 2012). It scores highly in terms of its institutions (the regulatory and political environment in particular); creative outputs (e.g. recreation and culture consumption); market sophistication (e.g. ease of getting credit and of protecting of investors); human capital and research (e.g. investment in primary and secondary education).

It not surprising therefore that software is becoming an important factor in New Zealand's economic output, for example Orion Health, Zero, Wildfire Interactive, Silverstripe, Litmos and many other NZ companies have contributed to a renaissance in overseas computer based earnings. NZ companies working in IT services and support and software development recorded a revenue growth of 13% in the financial year 2010-2011, a considerable growth in revenue compared to the whole high tech sector (Shanahan, 2011).

2. Models of Innovation

The literature shows that there is no consensus on how innovation in general takes place and gets managed (Sundbo et al., 2006). Neither is there much agreement even on its foundations, although there have been attempts at synthesis by a variety of disciplines (Loch & Kavadias, 2007; Pikkarainen et al., 2011; Smith, 2007).

Linear models break the innovation process into a pathway, a series of steps from idea generation through concept and development, to testing and commercialisation (e.g. Barczak, Sultan, & Hultink, 2007; Cooper, 2008; Sandberg, 2008). Linear models of innovation have dominated for many decades (Sandberg, 2008). Often there is an implied stage model: when one activity ends, another begins. Berkhout, Hartmann and Trott (2010) group linear models of innovation into two variations: technology-driven models and customer-needs-driven models. Both variations share the weakness of focusing on the initial driver, the starting point for the innovation, rather than treating the whole process as a dynamic system.

The advantage of linear models of innovation is that they break the process down into steps, and provide a useful way of thinking about how to develop new products, even if it is oversimplified. Linear models are less about actual practice and more about providing a formalised guideline to enable managers to find some way to control an unruly process. Linear models of innovation have strongly influenced thinking about software development methods. Having a model, even a bad one, creates a sense of order in a complex and sometimes chaotic process (Smith, 2007).

Another model comes from knowledge management theory. Nonaka and Kenney (1991) state that innovation is about managing information flows on the way to creating an actual product. The process is not one of logical deduction, or sequential steps. The manager's role is similar to that of a knowledge manager: facilitating the flow of information - tacit, intuitive and explicit - by removing constraints inside and outside the organisation. Thus, innovation can start from many places and can have many drivers. It does not follow a linear path and managing it always involves a series of interconnected activities.

This idea of interconnected activities led to another model by Ven, Polley, Garud, and Venkataraman (1999). They theorized that the innovation process takes place along six dimensions simultaneously. These are: theories of change; organisational learning; leadership; new business start-ups; and relationships both inside and outside the organisation. Innovation management thus is presented as a broad group of activities set in nonlinear dynamic patterns. These activities are constrained and enabled by various factors inside and outside organisations.

This in turn led to the Cyclic Innovation Model (Berkhout, Hartmann, & Trott, 2011). This claims to provide a description of how complexity in innovation is determined. The determining factors are: the interaction of markets; customers; products ;and science. It argues that innovation is not a linear, sequential process. CIM emphasises the complex interactions of innovation. The model links product creation, market transitions, scientific exploration and technological research in specific ways (see Figure 1).

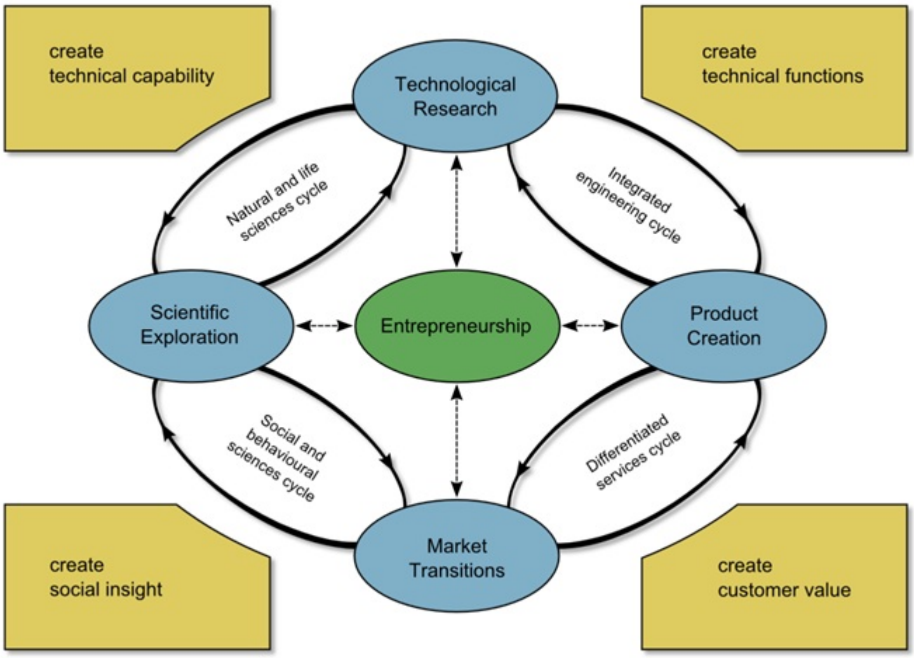

Figure 1. The Cyclic Innovation Model

These represent the four poles of industry (the supply), markets (the demand), basic science and technology research which interact in different ways. For instance, markets and industry together create customer value; science and technology together create technical capabilities; technology and industry create technical functions; and so on. Together science and markets create what the authors call social insights (and, it is implied, social change).

3. Development Methodologies

These models of innovation have influenced how software is developed. The classic example of the application of linear models to software development is the waterfall method, the traditional way software has been managed since the 1970s (Larman, 2004). The waterfall method of software development puts great emphasis on prior specifications and is a documentation driven process. It is sequential with distinct phases of development. Moving to the next phase is only allowed after the previous phase is complete, documented and signed off. Estimates are calculated early in the process and development is seen as managing set processes so as to execute a design that is more or less complete from the beginning.

The opposite approach recognises the dynamics of software as a product and managing the many conflicting factors by tightly defining tiny functions and joining them together. Agile is an umbrella term for a modern development methodology which combines a number of iterative and incremental software development methods so that work is broken into cycles (iterations) and small deliverables are created quickly. The software develops organically by coupling together tested modules that constantly extend the functionality towards what the client or market wants.

While these methods have been presented as deeply contrasting, in reality there are many variations on them. Waterfall and agile are sometimes combined, for instance, where iterations take place within phases of a waterfall project. Further, McConnell (2004) argues that "an iterative approach that ignores prerequisites can end up costing significantly more than a sequential project that pays close attention to prerequisites" (p. 34).

The models of innovation which apply to manufactured products do not easily fit software products. Innovation with software is different (Pikkarainen et al. 2011). Unlike manufactured physical goods, software can be endlessly changed; it can be delivered in small chunks and to some types of users first; there are almost endless ways of achieving functional goals; Software is intangible. It cannot be held in the hand and tried out. It must be experienced and played with to understand it; Barriers to entry in the software market are relatively low; Consumers are often involved in the design and testing of the product. Software development companies can take advantage of these different classes of users: consult power users, co-opt community champions, and use beta releases to test their designs; Individual programmers can have a disproportionate impact on productivity. The productivity of expert software engineers can be up to 10x greater than merely 'good' engineers (McConnell, 2011).

Because of this, modern software development methods usually take an iterative approach based on some form of Agile methodology. However, it is hypothesised that the particular demands of trying to produce products which are ill defined to start with, and which depend on technologies that still evolving, and aimed at markets that are embryonic, might require a different approach.

Most research into innovation management focuses on large companies and originates in the United States (Dodgson, Gann, & Salter, 2008) or Europe (Curzio & Fortis, 2005). These situations are quite different from the typical NZ software company, and there are concerns about whether that research can be applied here (Pikkarainen et al., 2011; Sundbo, Gallina, Serin, & Davis, 2006). Development of software products needs specialized management, but little research has been published on how companies actually manage at the leading edge of software innovation. This research therefore aimed to determine how development is managed in much smaller companies in NZ.

This study therefore researched leading New Zealand software companies to establish which model or models they use to cope with these conflicting characteristics. Eight high tech companies whose primary revenues are from commercialised software products were invited to participate. All interviewees were senior product development managers, heads of development, Chief Technology Officers, or company founders close to the development process.

4. The Empirical Research

Complexity has multiple aspects (Brown, 2011): a complex task requires a long time span; will require many decisions along the way; involves multiple factors and unknown contingencies; may take years to become clear. Creating a complex piece of software, perhaps one which will need to be patented, will take time to plan, develop and make the 'thousands of decisions' needed (Pikkarainen et al., 2011, p. 108). Less complex tasks are shorter, require fewer decisions and hence are less ambiguous. They are less challenging.

Accordingly, each company was rated in terms of its current level of complexity for the four CIM poles *product creation, market transitions, scientific exploration* and *technological research*. Ratings of *low, medium* and *high* complexity were given and then these were used to seek correlations with each organization's observed management style. Using this framework the research succeeded in providing a way of classifying the management style of each organization.

Four broad factors associated with complexity determined how the organization managed its developments: the nature of the customer, how workflow was controlled, how teams were set up, and how the build process was treated. All but one used iterative software development methods but there were many variations. All participants except one described having to adjust their software development methods to the particular challenges of developing an emerging product.

High complexity organizations were more likely to favour an iterative NPD method. They did not create upfront specifications but allowed the functionality of the new product to emerge from multiple iterations. They involved the customers at every stage. Workflow practices included: being able to chunk the project into manageable complexity, using prototyping, partial builds, upfront design dummies, experimentation and learning from small and early failures. It was clear in high complexity organizations that agile was the preferred software development method due its ability to handle rapid changes in specifications and technologies. Team management was crucial in all eight organizations but some had developers in different cities or even different countries so teams had to be actively managed and controlled. Working in the same physical space was thought to bring benefits which are so important that it warranted one organization flying in a remote team member on a regular basis. The usual development team size was four to seven people. These teams either had a lead developer or operated as a self-managing team. Teams were also given a great deal of freedom. Developers often were given time to work on projects of their own choice. These experimentation days, such as are well established at Google, were put in place with the intention of boosting creativity and keeping developers interested. 'Every second week you have half a day where you can just down tools and work on anything you like'.

As ways of managing the build, developers usually enjoyed a high degree of autonomy and were expected to work towards set goals without much supervision. They had varying degrees of authority to make changes depending on the management style of the particular organisation. Balancing this autonomy with the organization's development plan needed a constant flow of communication and feedback between the teams and all participants. Examples were given of daily stand-up meetings, internal forums, water cooler chats and accessible office spaces.

Medium Complexity organizations used more formal methodologies. Two participants used a formal stage gated software development method. This was appropriate to their innovation complexity because in their cases it was very clear what their new products needed to do. The challenge for them was in ensuring that the software would do this reliably. Their complexity lay in one specific area: the science. Converting new knowledge into tangible functions and applying that knowledge was the basis of their success. One drawback to this more formal process driven model was that fast prototyping development was not possible and the organizations were aware of this. The one company which adopted a structured methodology for their software development method had to follow that methodology because it was specified as a condition of their customer's regulating agency.

5. Conclusion

This study found that while there is considerable variation, the management of innovation was largely determined and predicted by the level of complexity. The findings complied with the theory of the CIM model and showed that complexity could be analysed even in situations where the final design of the product was not known to either the developers or the customers, and the way to achieve it was unclear. Organizations with complex innovation challenges will have more iterative software development, more flexible internal processes and swifter decision-making styles. Companies with less complexity in innovation will tend to have more formal structured approaches; fewer reviews of process and less product experimentation.

References

Barczak, G., Sultan, F., & Hultink, E. J. (2007). Determinants of IT Usage and New Product Performance. *Journal of Product Innovation Management*, *24*(6), 600-613. doi:10.1111/j.1540-5885.2007.00274.x

Berkhout, G., Hartmann, D., & Trott, P. (2010). Connecting technological capabilities with market needs using a cyclic innovation model. *R&D Management*, *40*(5), 474-490. doi:10.1111/j.1467-9310.2010.00618.x

Berkhout, A., Hartmann, D. & Trott, P. (2011) The role of entrepreneurship in innovation. *International Journal of Entrepreneurship and Innovation Management*, *14*(1).

Brown, J. (2011). The legacy of Elliott Jaques. *International Journal of Applied Psychoanalytic Studies*, *8*(1), 89-94. doi:10.1002/aps.273

Callaghan, P. T. (2009). *Wool to Weta : Transforming New Zealand's Culture and Economy*. Auckland, N.Z.: Auckland University Press.

Cooper, R. G. (2008). Perspective: The Stage-Gate Idea-to-Launch Process-Update, What's New, and NexGen Systems. *Journal of Product Innovation Management*, *25*(3), 213-232. doi:10.1111/j.1540-5885.2008.00296.x

Curzio, A. Q., & Fortis, M. (Eds.). (2005). *Research and technological innovation the challenge for a new Europe*. Heidelberg; New York:Physica-Verlag.

Dodgson, M., Gann, D., & Salter, A. (2008). *The Management of Technological innovation : strategy and praxis*. Oxford, UK: Oxford University Press.

INSEAD. (2012). *The Global Innovation Index 2012*. Fontainebleau. Retrieved from http://www.globalinnovationindex.org/gii/main/fullreport/index.html

Larman, C. (2004). *Agile and Iterative Development : A Manager's Guide*. Boston: Addison-Wesley.

Loch, C. H., & Kavadias, S. (2007). *Handbook of New Product Development Management*. Amsterdam; New York: Butterworth-Heinemann.

McConnell, S. (2011). What Does 10x Mean? Measuring Variations in Programmer Productivity. In A. Oram & G. Wilson (Eds.), *Making Software : What Really Works, and Why We Believe It* (pp. 567-574). Sebastopol, CA: O'Reilly.

Nonaka, I., & Kenney, M. (1991). Towards a new theory of innovation management: A case study comparing Canon, Inc. and Apple Computer, Inc. *Journal of Engineering and Technology Management*, *8*(1), 67-83. doi:10.1016/0923-4748(91)90005-C

Pikkarainen, M., Codenie, W., Boucart, N., & Heredia Alvaro, J. A. (2011). *The art of software innovation eight practice areas to inspire your business*. Berlin; Heidelberg: Springer.

Sandberg, B. (2008). *Managing and Marketing Radical Innovations: Marketing new technology*. Routledge Studies in Innovation, Organization and Technology. London; New York: Routledge.

Shanahan, G. (2011). *TIN100 Industry Analysis: New Zealand 2011* (7th ed.). TIN100 Industry Analysis. Auckland, N.Z.: Technology Investment Network Ltd.

Smith, P. G. (2007). *Flexible product development : Building agility for changing markets.* Hoboken: John Wiley.

Sundbo, J., Gallina, A., Serin, G., & Davis, J. (Eds.). (2006). *Contemporary Management of Innovation : Are We Asking the Right Questions?* New York, N.Y.: Palgrave Macmillan.

Ven, A. H. V. de, Polley, D. E., Garud, R., & Venkataraman, S. (1999). *The Innovation Journey*. New York, N.Y.: Oxford University Press.

Copyright © 2014 Spill, H. & Mason, D.

Journal of Applied Computing and Information Technology (JACIT): ISSN 2230-4398

(Incorporating the Bulletin of Applied Computing and Information Technology, NACCQ: ISSN 1176-4120 and
Journal of Applied Computing and Information Technology, NACCQ: ISSN 1174-0175)

Copyright ©2014 CITRENZ.

The author(s) assign to CITRENZ and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced.

The author(s) also grant a non-exclusive licence to CITRENZ to publish this document in full on the World Wide Web (prime sites and mirrors) and in printed form within the Journal of Applied Computing and Information Technology. Authors retain their individual intellectual property rights.

Alison Clear, Tony Clear, Michael Verhaart, Donald Joyce, and Nick Wallingford (Eds.)

An Open Access Journal, DOAJ #22304398, (✓zotero)