NACCQ Fostering Computing Education in NZ

Fostering Computing Education in NZ

Bulletin of Applied Computing and Information Technology

Managing radical software engineers

03:02
2005, Jul

Kevin C. Desouza
Institute for Engaged Business Research, The Engaged Enterprise, U.S.A.
desouza@engagedenterprise.com

Yukika Awazu
Institute for Engaged Business Research, The Engaged Enterprise, U.S.A.
awazu@engagedenterprise.com

Desouza, K. C. & Awazu, Y. (2005). Managing radical software engineers. *Bulletin of Applied Computing and Information Technology*, 3(2). Retrieved June 2, 2015 from http://www.citrenz.ac.nz/bacit/0302/2005Desouza_SwEngineers.htm

Abstract

Innovations in software engineering organizations frequently emerge from risky behavior. Most often, these risks are taken by a small percentage of the software engineering staff - 'radical engineers' (REs). They go against the status quo, experiment with new methods or technologies, and have the burden of bringing the innovations into the mainstream of the organization. Most organizations do a poor job of adequately managing REs. They can be found at either end of the order-chaos management continuum. Successful software organizations are those that are able to balance between the extremes to manage REs effectively. In this paper, we discuss lessons learnt in managing REs.

Keywords

Software Engineers, innovation, knowledge workers, knowledge management, software development

1. Introduction

Numerous research studies and real-life case studies have attested to the fact that innovation is risky business (Rogers, 1995). The innovator (or innovation entity) must take risk in experimenting with new ideas, inventing novel solutions, and questioning the status quo. In today's economic environment, organizations and employees face a critical conundrum. While we have significant pressures on organizations to innovate and come up with new ideas to improve performance and add to shareholder wealth, there are also increased pressures not to make investments that do not pan out. Similarly, employees face a challenge - they are asked to do more with fewer resources, due to tough economic times. Nevertheless, there is pressure on them to innovate in order to improve work processes and devise new products.

The above conundrum is applicable and critical to the management of software organizations. Software organizations will not survive unless they can innovate. Consider the case of Google.com. Google has become the standard in terms of what is expected from a search engine on the Internet. The edge possessed by Google gives it sustained competitive advantages, so long as it continues to innovate and improve its tool. Failure to innovate and remain as a leader will allow companies such as Microsoft to claim a piece of the market share. Currently, Microsoft is doing its best to make headway with its search engine, but lags far behind Google.

1.1 Innovations in Software Organizations

Innovations occur in multiple ways and forms. We can have process innovations, where the software organization comes up with a new method or process to execute a given task. An example of this type of innovation is the creation of agile programming methodologies. We can also have product innovations in which the organization develops news software solutions. Finally, we can also have a combination of the two - a process innovation bundled into a product. Here, an organization develops a new process, packages it, and then sells it to other firms in the marketplace. Not all software organizations will engage in all types of innovations; some may focus on product innovations, while others focus their energies on improving the process.

Software engineering is knowledge work. Knowledge is used to guide work practices, shape products and services, and inform innovations (Basili & Caldiera, 1995; Davenport, Thomas, & Desouza, 2003; Desouza, 2003; Desouza & Awazu, 2005). Software organizations must invent new knowledge and reuse their existing knowledge in an effective and efficient manner. The challenge faced by software organizations is one of exploiting their current knowledge and becoming efficient, while exploring and inventing new knowledge. Inventing is only the first step; inventions need to be translated into innovations. Inventions need to make their way into the marketplace. Inventions are of limited value unless an organization can earn revenues from them.

Innovations in software organization are seldom planned, rather they are emergent. A light-bulb will click in the mind of an employee, who may then decide to pursue the idea by making it explicit. This will be followed by a period of trials and errors, and hopefully the outcome is an invention. Organizations can seldom order people to innovate; rather they must actively seek out inventions and nurture them into innovations. In our experience, only a handful of software engineers take the time, effort, and risks involved in devising inventions. These individuals are normally considered as radicals in their organizations, as they seldom go with the status quo and do not care about spending the time or effort to build a viable alternative. When Radical Engineers (REs) are successful they are given glamorous rewards, when they fail they are scorned upon and ridiculed. The history of innovation tells us that the REs will fail more often than they succeed. Failure however is not a bad thing. New knowledge is often discovered during a failed experiment, and must be used to improve future experiments.

Organizations that know how to manage REs have a lot to gain from their efforts. Organizations that struggle with managing REs usually loose REs to other organizations or stifle their creativity. In our experience and research, we found that most organizations do a poor job managing REs. Organizations can be found at either end of the order-chaos continuum. In organizations where order is preferred, REs are shunned upon and are seldom given room to experiment, resulting in poor or no software innovations. At the other extreme, we see the dominance of chaotic behavior. Here, REs conduct experiments, take risks, and come up with innovations in an ad-hoc and unsystematic manner. Under the chaotic regime, there is lack of an effective channel for inventions to enter the mainstream of organizational work and improve software practices or products. Successful software organizations are those that are able to balance between these extremes to manage REs effectively. In this paper, we will share insights on managing REs and suggest a few interventions to better engage REs for successful attainment of organizational goals. We will present a model towards effective management of REs that integrates the strengths of each extreme. This model is based on our research and consulting efforts in several organizations that have been successful in managing REs and leveraging their inventions into innovations.

2. Methodology

To prepare this paper we gathered data from multiple avenues. Predominantly we conducted semi-structured interviews with software engineers in five organizations. In all, we interviewed 30 engineers. 12 engineers interviewed considered their organizations to be predominantly order based, while 14 thought of their management practices being chaos dominated. Four engineers found their management practices to be somewhere in between the two extremes. This paper is an experience report; hence we supplemented data collected by recollecting our own experiences working for software engineering organizations. We understand the limitations associated with not following a thorough and scientific method of data collection. However, even after acknowledging for these, we do feel that the paper brings some interesting insights in an exploratory fashion. Future research can build on the ideas presented to conduct theoretical and empirical investigations.

3. Radical Engineers

REs are engineers who, more often than not, like to come up with new inventions and push the innovative limits of the organization. In our experience, all organizations have REs. Some REs may be outlandish and bold while others are "closet REs" or "hidden REs".

Consider the case of a software organization in the Chicago area. Three software engineers would meet every other Saturday morning at a local café and spend almost half-a-day working on what they called the "Revolution ABC". ABC, a pseudonym, stood for the current version of the software they wanted to revolutionize. The software engineers were REs, but were hidden from the organization - closet REs. Contrast this with DELTA, another software organization based in the Illinois area. DELTA would hold a one-day "Show-Me" event every month where engineers were encouraged to display their inventions, discoveries, and new insights. The Show-Me day would feature presentations, critiques, discussions, and debates around novel concepts that engineers were working on. Those who presented on the Show-Me days represented only about ten percent of the entire engineering staff; these individuals are REs and were open about it.

3.1 Identifying REs

REs have several peculiar characteristics. First, they thrive in independent workspaces where they have autonomy. This is in contrast to the common engineer who likes to follow prescribed outlines, schedules, and meet specifications. Independence is an important characteristic to acknowledge as REs resist micro-management.

Second, REs have excellent contacts with the external software communities. They are more likely to turn to external sources of knowledge than those that are internal. For example, REs will participate in altruistic software communities such as Open Source Forums. REs are likely to bring innovations into the organizations from the external sources, thus, providing the organization with access to valuable knowledge beyond its boundaries.

Third, their agendas are global and go beyond the scope of the organization. They are very passionate about the work they do. This can be fascination about a technology platform, a programming language, or even a technology. They view their work in the context of the global user community. In one organization, a RE would document his inventions or breakthroughs on a Developers Listserv. He wanted to better the experience of all users of the software by sharing his insights and knowledge.

Fourth, REs are experts. They seldom have patience to clearly articulate their assumptions, train novices on the basic concepts, or even accept help from people within the organization unless it is from another RE. In doing so, they isolate themselves from the main group of the organization and are often seen as outsiders. In DELTA we asked Alec, a RE, about how he viewed himself in the context of his team and the organization at-large, he remarked:

> *"I am an employee...but I am often viewed as a rebel...I often do not go with the status quo...because I know they are wrong...they do not get it...we are doing something for short-term gains that is ridiculous and will come back to bite us in the...I do not care what Jack or Stacy think...if I am of no value to the organization then fire me..."*

Finally, REs rarely have a propensity to get involved in organizational politics. They view such efforts as a waste of time and are not inclined to spend their time appeasing others or pushing political agendas. As one engineer we interviewed remarked,

> *"I leave the political battles to my peers...I am here to build, design, and test...yes I will follow the organizational methodology...But I will also invent my own method and work using it...Who cares if Suzy (the department head) does not like it...too bad".*

Sentiments such as these are common. However, we must stress that such emotions are not negatives. REs, at least the ones we know, are good-hearted people who are above the meager organizational politics. This is a good thing as they devote their complete energies to their work and are not easily distracted.

4. Between Order and Chaos

As we mentioned earlier, most organizations fall on two extremes in their approach to managing REs. We have order-based regimes, where employees are encouraged (or forced) to abide by over-arching rules and regulations. On the other hand, we have a chaotic situation setup, where in there are no strict rules governing inventions and employees are left to their own devices. We will now briefly describe the two extreme management approaches, following this we will elaborate on a hybrid model.

4.1 Order

Regimes that prefer an ordered style of operations are normally extremely efficient in what they do. Efficient work practices and processes are an outcome of following prescribed specifications, schedules, policies, and directives. These regimes resist change in most forms, as they call into question the way work is conducted in the organization. Most often, these organizations are usually the last to appreciate new ideas, innovations, and methods. Most of the time, external pressures force change upon such organizations (DiMaggio & Powell 1985). At best, order-based regimes can get better at doing what they are presently doing. They are not successful in sensing future opportunities and exploiting them. They lag the industry leader and play catch-up by imitating innovations.

The prominence of order-based management in an organization makes it difficult for employees to invent and innovate. We find *closet REs* in these organizations. They seldom want to be bold about their inventions as they will attract undue attention from senior authorities, resulting in putting their jobs in jeopardy. Another issue one must contend with in such organizations is rigidity and focus. Seldom are order-based regimes agile and flexible, rather they are focused and rigid. Those organizations think they have a narrow niche and will try to attain their objectives using the most efficient route. Hence, they are not likely to expel resources to try something new or even listen to new ideas in an open-manner. Closet REs will have a difficult time introducing inventions in these organizations also if they are valuable. Seniority of personnel plays a critical role in decision making in order-dominated organizations. Newcomers and REs are given less weight in decisions as they are "new" and have not been immersed into the organizational culture and values. As one employee remarked:

> *"I have only been here for six months...who am I to question...Marci...she has the support of the boss...you know, she has twenty years of seniority on me..."*

4.2 Chaos

Under chaotic-management styles, REs are left on their own to conduct inventions and experiment. On the positive side, the organization does not hamper an individual's creativity or experimentation efforts, so long as it does not interfere with the completion work assignments. On the negative side, the organization seldom gains from the REs discoveries. There is an absence of a systematic process on how to bring personal inventions into the mainstream of the organization. Put another way, there is no mechanism that states how personal *inventions* can be transformed into organizational innovations.

Consider the case of an organization that provided its software engineers with access to "think rooms". These rooms were intended to be used by engineers to take a break from coding assignments. The rooms housed a video arcade, some board games, a fridge stocked with refreshments, couches, whiteboards, and a music system. This room became very popular with the engineers, especially for meetings. As a project manager remarked:

> *"We like to have meetings here as it is more informal that the other rooms...the*

whole team can engage in some entertainment such as team competitions in car racing [a video game] after the meeting".

REs took a particular liking to the room. They would use it to brainstorm for ideas. Almost every day, the cleaning staff would see a note on the Whiteboard stating, "*Do not erase this*". The REs would use the board to facilitate their discussions and record ideas. However, no one outside of the REs paid any attention to the material on the board. Senior management never knew how the Whiteboard was being used. Only after some consultants intervened, did they realize that the Whiteboard had a collection of solutions to critical organizational problems. Unfortunately some of the REs had already left the organization by the time of this realization.

Chaotic regimes are excellent thriving spots for REs. However, they are not beneficial for the organization as no gains can be made from the inventions of REs. REs love chaotic regimes as they are not micro-managed and are given space to explore. The organization usually incurs high resource costs to provide such an environment. This cost is rarely recovered as inventions are seldom commercialized. In the end, the organization will have a collection of inventions scattered in its midst, but it may not be able to identify and leverage them into innovations for revenue generation purposes.

5. Ordered Chaos or Chaotic Order

The solution to adequate management of REs is found between the extremes of order and chaos.

The organization should have an *ordered* innovation process. The organization should take a conscious and deliberate effort to promote software inventions and leverage these into innovations. In order to do so, the first step is to allow REs to be *chaotic*. A successful organization will learn to provide REs their space to invent and innovate. In some organizations that we know, REs are pulled out of *mundane* projects and are given research assignments. These assignments help them focus their energies on their problem of interest. However, by giving REs room, we must not allow them to be free-birds. The role of the organization will be to manage the *research project*. REs should be treated as managers. They should be given a set budget to explore their fascinations, encouraged to work with their fellow REs, and be required to report on their experiments. Synergies from work between REs can lead to better solutions and inventions. By exerting control over the research assignment, the organization can manage such projects and be aware of the work conducted by REs.

Once the RE or REs come up with an invention, of some significant value, the role of the organization increases. Now, it becomes important to move away from a chaotic process to one that is ordered and defined. The organization must have a process in place to test out the invention, review it, enhance it, produce it, and market it. In doing these tasks, the organization needs to demonstrate high levels of efficiency, and hence ordered regimes are preferred. The REs rarely want to get involved in these aspects of the innovation, and are happy if their invention is recognized by the organization as valuable. Successful organizations have a "commercialization group" that oversees the movement of an invention to a product or solution in the market. This group is responsible for being effective in how they market new inventions. We must note that not all inventions are meant for the *external* world, some inventions have bearing on the *internal* workings of the organization. For example, the calibration of a new work process model is of use to the organization to streamline internal processes. In cases of internal-inventions, the commercialization group will work with other departments in the organization to push the invention into the work processes.

Balancing between order and chaos is critical. To summarize, the role of management is to provide a platform for inventions and manage, from a high-level, the work that is conducted on the invention platform. REs will be REs, and their inventions will usually be an outcome of their chaotic inquiries into problems. While there is no changing this, management must try to provide REs with some kind of structure around which they can be chaotic, for example by giving them a budget and asking them to discuss their deliverables and experiments. Providing structure demonstrates to the REs that the organization is interested in what they are doing, and prevents them from thinking about leaving the organization for a rival. REs are rational-beings and understand that some management control needs to be in place. Once the inventions are developed, the organization must have an efficient process in place to get them to their desired markets. Commercializing inventions needs to be efficient and effective, and there can no room for guesswork or randomness here. Order dominates this aspect of the innovation process.

5.1 When to Have REs?

Having REs on software engineering teams is not always desirable. There are situations under which they may actually hamper team performance and be disruptive to project completion. In our experience it is best to think about the viability of having REs on a software project by looking at the kind of work that needs to be accomplished.

If the work is structured, routine-oriented, simplistic, and well understood, REs should not be made of the software engineering team. For structured and routine projects it is best to have engineers who are able to follow rules and execute steps with high efficiency and effectiveness. It is best to have novices and junior engineers handle these assignments. Having REs as part of these teams will be counterproductive as they may question the premises of the structured plan and transform a rather simple task to a complex thought exercise.

REs are best placed in projects that are unstructured, novel, and those that have uncertainty components. For instance, assignments that involve use of new tools and techniques are prime examples of such projects. REs thrive in these environments and can provide value here, as they will be able to innovate. Following pre-determined rules will not work too well in these contexts. Success in these environments is more a function of innovating and adapting solutions to the new problem rather than simple application of rules.

6. Conclusion

Managing REs will be a critical competency for software organizations to master. Several factors contribute to the urgency of acquiring this competency. First, the software market continues to be one of the most fiercely competitive arenas. The keys to success will lie in a firm's ability to invent at a constant pace, and more importantly being successful in introducing inventions in the marketplace.

Second, in order to innovate successfully an organization must have the ability to hire and nurture the best minds. The best minds are those that can come up with inventions by seeing things the majority of the workforce does not. These minds are curious and creative; they seldom work in structured spaces and often think outside the box. Rather than trying to manage them in a rigid and authoritative fashion, which will stifle their drives and innovation capacities, the organization must embrace their radicalism and provide them with alternatives to release such energy in a fruitful manner.

Third, innovations in the software industry have a rather short life span. Seldom does the industry hold on to one invention, technology, or software method for long. New methods are calibrated as quickly as one is released. This again puts pressure on companies to stay abreast and be leaders rather than followers or laggards in the innovation cycle. Leaders usually dominate their markets; the followers are left fighting over a minuscule share with other followers.

Fourth, the nature of software is undergoing dramatic change, which calls for greater care in new idea generation. In the past, it was essential to possess an advanced degree in Computer Science, Information Systems, Computer Engineering, or even Mathematics to be hired as a programmer. Today, a skilled high school student can program in Visual Basic and develop rather sophisticated applications. The future of the software business is going to be one where users of software will take it upon themselves to program and customize technologies. The role of the software organization will be to provide them platforms to do this. Programming applications used by individual customers will become easier, because the consumer's knowledge of software and computing continues to rise at an astonishing rate. Software organizations can no longer charge lofty fees for systems development and programming assignments. They however will be able to command high revenues for innovative products that cannot be duplicated in the marketplace. This is where the need for constant innovations becomes critical.

Finally, the most lucrative areas for the future of software businesses are un-chartered waters. Software organizations will be called upon to generate solutions for meeting the needs of ubiquitous and pervasive computing, complex bioengineering and medical informatics problems, sophisticated electronic trading and negotiation systems, mobile devices, and logistics problems. These will call for novel thinking. Applying knowledge from old endeavors to new problems will be of limited value. New ideas will be needed to meet the new challenges posed. REs are an organization's best bet for getting at these ideas, fostering their growth, and calibrating successful innovations. As one manager we know remarked:

> *"A hammer is always useful, you know the wooden one...most software organizations are thinking how they can build an electronic hammer...this is futile...the smart ones are thinking of getting rid of thorny nails...what worked in the past, may not even survive a initial review in the future".*

REs can be highly valued assets or costly liabilities to organizations depending on how they are managed. Successful organizations use their REs to help foster inventions and innovations thus becoming market leaders. Less-capable organizations eventually face competition from REs as they leave the organization and start-up their own organizations or join rival firms where they are given room to grow and develop their ideas. In conclusion here is our prescription, identify your REs, make it known that you want them to invent and experiment, provide them alternatives and avenues for channeling their energies, and ensure that the organization can take advantage of their breakthroughs.

References

Basili, V., & Caldiera, G. (1995). Improve software quality by reusing knowledge and experience. *Sloan Management Review*, 37(1), pp. 55-64.

Davenport, T.H., Thomas, R.J., & Desouza, K.C. (2003). Reusing intellectual assets. *Industrial Management*, 45(3), pp. 12-17.

Desouza, K.C. (2003). Barriers to effective use of knowledge management systems in software engineering. *Communications of the ACM*, 46(1), pp. 99-101.

Desouza, K.C., & Awazu, Y. (2005). *Engaged Knowledge Management: Engagement with New Realities*. Hampshire, UK: Palgrave Macmillan.

DiMaggio, P. J., & Powell, W. W. (1983). The iron cage revisited - institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, 48(2), pp. 147-160.

Rogers, E. (1995), *The Diffusion of Innovations* (4th edition). New York: The Free Press.