Artikel-5.pdf

by

Failure to Launch:
Scope Creep and Other Causes of Failure from an Actor-Network Theory Perspective

Samiaji Sarosa, Accounting Department, Atma Jaya Yogyakarta University, Yogyakarta, Indonesia

Arthur Tatnall, Victoria University, Melbourne, Australia

ABSTRACT

One of the main causes of delayed and failed information systems project development is scope creep. The increasing number of features demanded by stakeholders to be built into the applications within a fixed time limit is a recipe for failure. This article looks into the process of a web application development failure, where scope creep was deemed as the main cause. An in depth look into the timeline of the project also reveal another cause, which was the failure of the application itself along with the platform (hardware and software) to actually execute the software. It is believed that an Actor-Network Theory framework is appropriate to analyse this case where a number if both human and non-human actors were involved. Data for this research was collected using participative observation. An analysis was conducted to find patterns of negotiations and communications between all the stakeholders during the design process. Actor-Network Theory was used to explain the power plays between actors. A model was constructed showing all the actors (stakeholders) and how the interplay among them developed.

Keywords: *Actor-Network Theory, Failure, Requirements Triage, Scope Creep, Stakeholders, Systems Developers*

INTRODUCTION

Scope creep is one of the problems of information systems project development (Avison & Fitzgerald, 2006). In his model of Requirements Triage, Davis (2005) proposes that a system requirements specification is a balancing act of three axis, namely resources (usually represented by financial constrains), technical (usually represented by the development team which means required technical effort to build the requirements) and the customer (who usually represents demand for software features). Turban and Volonino (2012) argue that a systems development project is constrained by three factors, which are time (similar to customer), budget (similar to resources) and scope (similar to technical). Where requirements keep growing during the project, this scope creep means that more time and effort is needed to build the systems. More time is usually translated into increases in budget. At the end this could mean a runaway or failed project (Brooks, 1995). In time-constrained development, scope creep is something that is most undesirable for developers.

DOI: 10.4018/ijantti.2015100101

This article analyses a web application development project that ran over time and over budget due to scope creep. The power play of the application developer team (technical aspect), managing director (financial aspect), and president director (customer aspect) were viewed from Requirements Triage (A. M. Davis, 2005) and Actor-Network Theory (Callon, 1999; Latour, 2005; Law, 1999). It is not only why but also how those three triage factors worked and were played through by these actors that is analysed.

SCOPE CREEP AND REQUIREMENTS TRIAGE

Scope Creep is defined as any additional requirements arising during the course of a software development project (Nurmuliani, Zowghi, & Fowell, 2004; Thakurta, 2013; Zowghi & Nurmuliani, 2002). Scope creep is a specific type of Requirements Volatility where the additional requirements are added instead of changed or removed. Any requirements changes that occurs throughout the development process will likely affect the completion of the project. Additional resources (likely including human, technical and financial resources) are then needed, and this will also affect the time needed to finish the project (Brooks, 1995; A. M. Davis, 2005). The Project Management Body of Knowledge (PMBoK) considers that project scope is a serious issue and has a whole section on Project Scope Management and the problems of scope creep (Project Management Institute, 2013). This includes discussion on the collection of project requirements, defining project scope, creating a Work Breakdown Structure, verifying scope and controlling scope.

Although scope creep is considered undesirable by most developers, the reality is that scope creep will often emerge during the development process and is often inevitable (Khan, 2006). Any change in the users' business needs, changes to the external environment, or even only changes in users' minds would justify the need to add additional features (A. M. Davis, 2005; A. M. Davis, Nurmuliani, Park, & Zowghi, 2008) and managing scope creep becomes an essential task within a software development project (Thakurta, 2013; Thakurta & Ahlemann, 2011; Zowghi & Nurmuliani, 2002).

Scope creep is not something that is unusual in any engineering of information technology project, including software engineering. Good project management is required to keep the project on the right course towards completion. All the project's stakeholders need to understand the impact of scope creep toward project completion and also at the end of the project, the functionality of the software produced. Having to understand the impact of any change in requirements the stakeholders can then decide if the changes were needed and justified. However, different stakeholders have different views on what changes are needed. The project manager needs to try to achieve compromise and consensus among the stakeholders.

Davis (2005) proposed a Requirements Triage as a way to balance the different views. A Requirements Triage assumes that in a software development project there are at least three factors to be considered. These are the number of requirements to be built, the desired or available time and the available budget. Requirements Triage tries to balance those three factors by selecting the requirements to be included in the software released in the desired time frame.

Any change in requirements, in this case additional requirements, would most likely require changes in the completion time (or release time) and additional budget. A compromise could be made by only building the reasonable requirements within a specific time and budget. Adding more resources (which translate into additional budget) into the project to finish the software with additional requirements in time, or maybe change in the release time to accommodate the additional time needs to incorporate additional requirements.

Each alternative has different consequences. If the time of release cannot be changed, then the budget needs to be changed (adding more resources into the project) and the requirements need to be changed. The change in requirements could mean the original requirements need to be formally changed to accommodate new additional requirements. Another problem might arise where adding new resources (especially new developers) into the project is that this might not make the project run faster or finished on time (Brooks, 1995). Changing time of release might be problematic since it could mean business' disadvantages such as competitors beating them to the market, a date imposed by government, etc. Changing budget might also be problematic if the organisation has limited financial resources and the rest of the company budget has already been allocated for other business needs.

Davis (A. M. Davis, 2005; A. M. Davis et al., 2008) argues that to reach compromise the stakeholders need to change their standing. Developers need to really look at the additional requirements and find a strategy to build them faster. Deadlines need to be negotiated and set at a reasonable time frame. Budgets need to be reallocated if necessary to enable the desired requirements and deadline changes.

In any case, adding new requirements is not good practice (Andriole, 1996; Balakian, Young, & Veerapaneni, 2002). Project managers need to freeze the requirements at some point to enable the developers to build them. Therefore it is necessary for developers to have some time to properly build, test and integrate the requirements into the software.

APPROACHES TO INNOVATION ADOPTION

To investigate any innovation adoption (and software development does involve this) it is worthwhile to follow one of the major theories of technological innovation. These include Innovation Translation (informed by actor-network theory) (Callon, 1986c; Latour, 1996a), Innovation Diffusion (Rogers, 1995, 2003) and the Technology Acceptance Model (TAM) (F. Davis, 1986, 1989; F. Davis, Bagozzi, & Warshaw, 1989), along with variants on these, the most recent being the Unified Theory of Acceptance and Use of Technology (UTUAT) (Venkatesh, Morris, Davis, & Davis, 2003).

An important difference between these theories however is the degree to which the adoption decision is seen as being completely rational, and whether provision is made for partial adoption. In this article we make use of Innovation Translation as it does not presume that the adoption decision is completely rational and does make provision for partial adoption. It thus fits better in describing the processes of scope creep and their consequences.

ACTOR-NETWORK THEORY

ANT shows how an idea or technology is adopted by an organisation or individual by means of various associations and interactions between human and non-human actors (Tatnall, 2002; Tatnall & Burgess, 2004). Bardini (Bardini, 1997) says that ANT deals with:

> *... progressive constitution of a network in which both human and non-human actors assume identities according to prevailing strategies of interaction. Actors' identities and qualities are defined during negotiations between representatives of human and non-human actors. The most important of these negotiations is 'translation', a multi faced interaction in which actors: construct common definitions and meanings, define representatives* and *co-opt each other in the pursuit of individual and collective objectives.*

The way that actor-network theory handles and describes the innovation adoption process of by what it calls Innovation Translation (Callon, 1986c; Latour, 1996a). This consists of four 'moments' (Callon, 1986a) or stages:

1. **Problematisation:** Key actors proposed solutions to the problem. The key actors then persuade the other actors that they all have the same interests and that the answer to the problems is in the solutions proposed by key actors.
2. **Interessement:** This involves processes that attempt to impose the identities and roles defined as solutions in the problematisation on the other actors. The key actors along with various other actors join a newly created network and then try to lock in other unconvinced actors. They then gradually dissolve the existing networks, replacing them with new networks created by these actors. These actors then try to impose the new identities on all the other actors.
3. **Enrolment:** A process where one set of actors (key actors) imposes their will on other actors and convinces them to join – to become enrolled. The other actors within the organisation will then be persuaded to follow the identities and roles defined by the key actors which will lead to the establishment of a stable network of alliances. To achieve its goal the enrolment process includes among other things coercion, seduction, and voluntary participation.
4. **Mobilisation:** This is where the proposed solutions gain wider acceptance. The network would grow larger with the involvement of other parties that were not previously involved. This growth is due to the influence of actors who actively promote the new network to others. ANT recognises that the key actors initiate adoption of the innovation and then build a network of individuals or organisations and non-humans (machine, tools, etc.) to adopt the innovation.

RESEARCH METHOD

Participative observation (Myers, 2009; Neyland, 2008) was used as a tool to collect data. Data collected included minutes of meetings, various design documents, email communications, field notes and discussions with stakeholders. Using this method enabled collection of a lot of data that would otherwise be only available to an insider. This also enabled insider insight into the problems that might not be immediately visible to an outsider.

The analysis was conducted using ANT (Callon, 1999; Latour, 2005; Law, 1999; Underwood, 2001) along with Requirements Triage (A. M. Davis, 2005). Davis suggests that Requirements Triage involves determining the requirements that should be satisfied given the time and resources available for the product's development. Using Requirements Triage, the problem is presented as a power play between financial, scope, and technical constraints (Brooks, 1995; A. M. Davis, 2005; Markus, 2002; Turban & Volonino, 2012). ANT provides a framework to investigate how these three factors were used by the actors to achieve their own agendas. A model showing the actors (stakeholders) and how the interplay among them was built is provided later in the article.

THE CASE STUDY

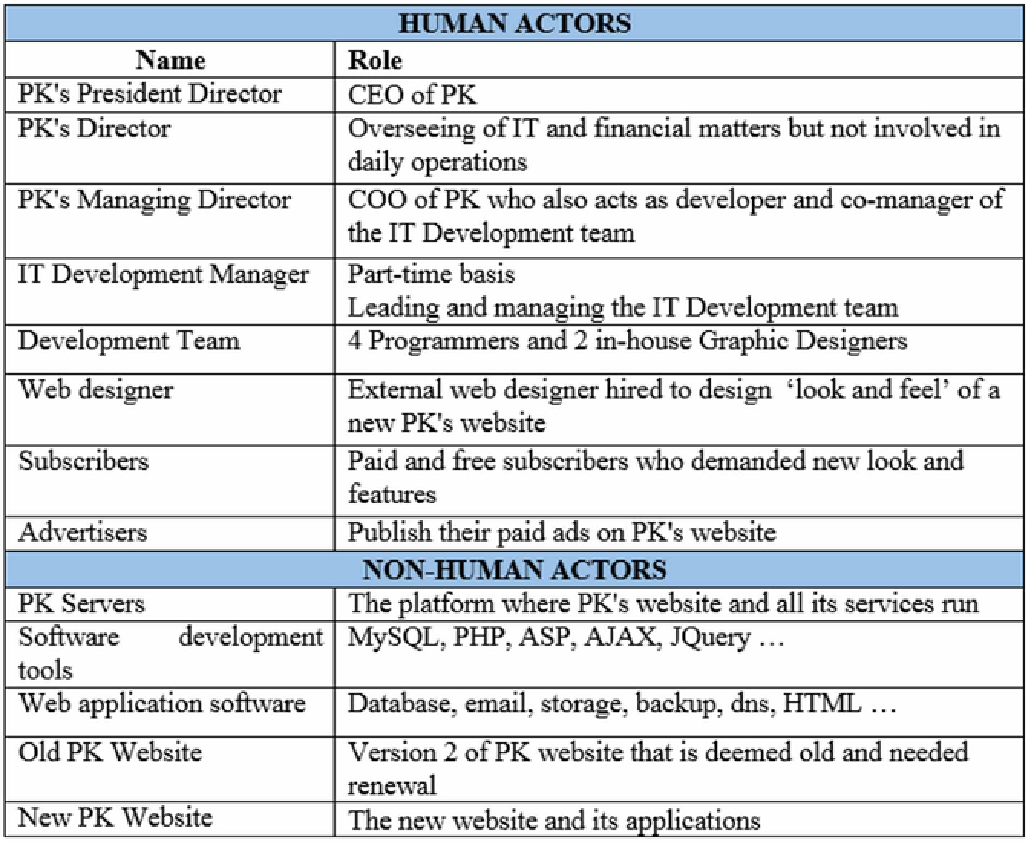
PK[1] is a small company in Indonesia (with less than 20 employers), which operates and manages a website. PK was founded 13 years ago. This website and its applications are quiet popular with more than 400,000 subscribers. The main revenue of PK comes from paid subscription, merchandise sales, and paid advertising on the website. The list of PK's actors can be seen in Table 1 below.

The website itself was originally build based on a Microsoft Access database and ASP scripting language running on a Microsoft Windows Server Platform thirteen years ago. The rapid growth of subscribers forced PK to port the website, after six months, using MySQL database and PHP scripting language running on a Linux Platform. Despite the change in platform, the basic structure and framework of PK's website remained the same. For the last 5 years there have been many complaints, suggestions and requests from subscriber to update the look and feel of PK's website. There were many social-media inspired features demanded by subscribers. Adding new features to accommodate demands was not an easy task with the old structure and frameworks. The look and feel of PK's website was also deemed old and out of date compared to the latest social-media networking sites.

PK's management and stockholders thus decided to build a new website. At that time PK only had two programmers but PK's managing director also had programming and networking skills. The managing director usually managed the development team. They also had one graphic designer.

Due to the enormous tasks ahead, PK's management decided to beef up the development team. First they hired a part-time development manager to oversee the development team. They

Table 1. List of actors



HUMAN ACTORS	
Name	**Role**
PK's President Director	CEO of PK
PK's Director	Overseeing of IT and financial matters but not involved in daily operations
PK's Managing Director	COO of PK who also acts as developer and co-manager of the IT Development team
IT Development Manager	Part-time basis Leading and managing the IT Development team
Development Team	4 Programmers and 2 in-house Graphic Designers
Web designer	External web designer hired to design 'look and feel' of a new PK's website
Subscribers	Paid and free subscribers who demanded new look and features
Advertisers	Publish their paid ads on PK's website
NON-HUMAN ACTORS	
PK Servers	The platform where PK's website and all its services run
Software development tools	MySQL, PHP, ASP, AJAX, JQuery …
Web application software	Database, email, storage, backup, dns, HTML …
Old PK Website	Version 2 of PK website that is deemed old and needed renewal
New PK Website	The new website and its applications

also hired two additional programmers. For web design, they hired an Australian web designer to work with the resident graphic designer. The structure of PK's organisation could be seen as follows in Figure 1.

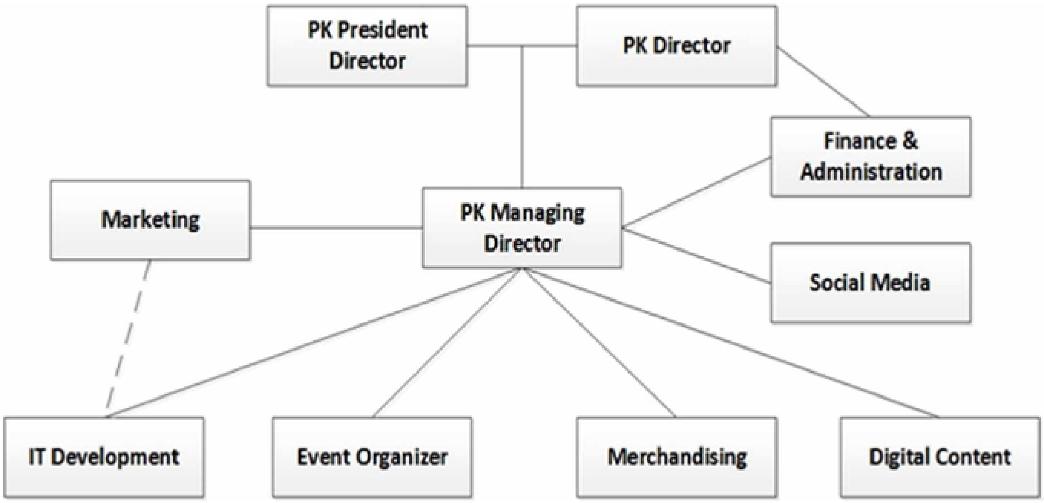
PK's management wanted the web to be modern, simple and full of functionality. The website architecture was built using an n-tier model (Hoffer, George, & Valacich, 2011; Hoffer, Prescott, & McFadden, 2007). This separates presentation layer, application layer, database layer and data storage layer. It was built on a Linux Platform using PHP and MySQL and utilizing AJAX and JQuery. The hardware itself remained the same and consisted of ten servers for various applications (web, database, email, storage, backup, dns, etc.). The website itself was to have backend parts which were planned to be developed by PK's IT development team. The front end parts were planned to be developed by the web designer.

The development process itself started August 2011 and the target was to finish the development and officially launched the website by 1st February 2012. PK's managing director set up an internal development environment that mimicked the production system. The development team worked five days a week meaning approximately 40 work hours weekly.

First the web designer and PK's team (President Director, Managing Director, Graphic Designers and IT Development Manager) discussed the design (graphical and functionalities) of the new PK website regularly through Google+ Hangout[2]. Once agreed, the web designer would make the detailed design. At the end the web designer would make the required HTML and Java Scripts files (the front end) for the website. The developers' duty was to integrate the front end from the designer into their own backend.

In order to meet the deadline, PK's management through the development manager, set up a project management file. Various deadlines and milestones were established to make sure the February 2012 launch date was met. The February 2012 deadline was decided, based on an advertisers' agreement. If the launch date was missed PK would have to compensate the advertisers who paid for advertising on daily rates. For this reason PK's development manager's proposal to use parallel conversion (Hoffer et al., 2011) was rejected. PK's managing director opted for direct cutover instead to save time.

Figure 1. PK's organisational structure in late 2011



Various examples of scope creep emerged during the development time. The first scope creep appeared in the front-end functionalities. PK's president director asked for additional features to be incorporated into the design each time they met with the web designer on Google+ Hangout session. The web designer needed to design a complete set of website looks and functionalities before starting to write the required HTML and Java Script files. The first set of files was supposed to arrive in early December 2012. It was actually delivered in the middle of December 2012. This left PK's development team very little time to learn about the files and incorporate them into the back end. The problem was exacerbated by the fact that the files sent by the web designer could not be understood and used by the development team. In the end, they recreated all the HTML and Java Script files themselves, which only added new workload to their already heavy burden.

The scope creep was intensified during January 2012 close to the launch time. PK's president director through the managing director pushed many new features deemed as the highest priority to the development team. When PK's management suddenly had ideas to be incorporated into the new website they immediately ordered the development team to include these without first consulting the development manager. The development manager, as a new and part time employee, was often bypassed. This was understandable due to the part-time nature of the development manager's position. The development manager was only available about half of the normal working time due to his commitment on his main job. The programmers, who were already overwhelmed, did not dare to voice any objections. Any objections from the development team were immediately crushed by PK's managing director. He was often remarking that he was once a programmer for a wig company and his experienced showed him that such change could be done.

To accelerate development time and to overcome the required additional time due to scope creep, PK's managing director made some adjustments. First he relaxed the working hours for the development team. They did not need to obey official office hours (9 AM to 5 PM) but were allowed to work overtime. They also worked on weekends and holidays. As a financial incentive, PK's management promised additional bonuses if the development team was able to finish their work ahead of schedule. Unfortunately, as Brooks (Brooks, 1995) pointed out, if a project was behind schedule, additional resources will not accelerate the project into completion. Intensified scope creep exacerbated the situation in January 2012. The development team worked frantically overtime, close to 18 – 20 work hours per day every day. The development manager also dedicated more time away from his main job to concentrate on the completion of the project. Despite all the bad signs, within the development environment the website worked, albeit with many features needing to be polished.

On the launch date of February 1st, 2012 the new website was moved into the production environment. It took an all-night work effort to finished and polished features along with data migration from the old website to the new. Before launch, the managing director and development team tested the website in its production environment setting (not launched yet). For unknown reasons at that time, the presentation layer (front end) was not able to render the website using data from the application layer (back end). The launch was postponed until they could fix the problems. The old website was put on static hold for the launch and stayed static until the new website was ready.

The development team desperately tried to solve the problem. It took almost 24 hours (giving them almost 48 hours of nonstop work) to make it work. The development team needed to review and fix codes from all the scripts. By the morning of the second day, the website was launched.

Within few hours of launch, complaints were coming in. Most of the complaints were about the lacklustre performance of the new website and error 2003[3]. Due to the complexity of the problems, PK's managing director revived the old website and postponed the launched of the new website until they could fix the problems.

The causes of the lacklustre performance were two: firstly the quality of the application scripts and the database were inadequate. There was not enough time to test the scripts due to time and human resources constraints. They were only able to conduct User Acceptance Tests (Kotonya & Sommerville, 1998; Meyers & Oberndorf, 2001; Pfleeger, 2001; Sommerville, 2001) instead of more rigorous tests. There was never any chance to optimize the scripts and new database for performance improvement. The second problem was apparently that the new website demanded more hardware resources compared to the old one. It was only known after the new website was put into a production environment. Direct cutover is the most risky conversion plan (Hoffer et al., 2011). Luckily, in this case the old system was still available. At the end PK needed to add three more servers (one web server and two database servers) and a significantly changed hardware configuration to cope with the workload of the new website. The new hardware configurations also meant that the files needed to be rewritten to accommodate changes.

It took almost 4 weeks of reworks and help from PK's Director[4], before the new PK website finally launched. PK's Director guided the programmers and PK's Managing Director on how to handle large volumes of data and on optimizing both applications and databases. After the second launch, the complaints were mostly regarding the features and the look and feel of the website instead of its performance. Another a month of fine-tuning, the new website was ready.

RESULTS AND ANALYSIS

If we looked at the case description in the previous section, it is quite clear that the cause of failure was intensified scope creep near the launch date. Additional requirements and features to be built into any system requires additional resources (budget) and time to finish (Brooks, 1995; A. M. Davis, 2005). Davis (A. M. Davis, 2005) suggested that a negotiation is required to achieve compromise within the limits of available time, budget, and technical resources. Brooks (1995) went further by saying adding resources into an already late project would not accelerate the project completion. However, in this case, PK's president director (obeying the demands of advertisers) did not budge – he made no attempt to change his problematisation of the situation. He demanded that ever-increasing features be built into the new website with unchanged budget and technical resources. PK's president director as the main actor had imposed his will (driven by business deals with advertisers) to the managing director and development manager, who in turn conveyed these demands to the developer team (programmers) (Latour, 1987, 1988, 1996c, 2005). It was not only the scope creep that threatened the completion of the new website, but there were also no additional resources and no time that could be allocated to the project. The programmers, the IT development manager and the managing director had somewhat of a naive view that the scope creep could be solved by adding more programmers or more working hours. The developer team themselves had almost had no voice in objecting to the additional workloads.

The project management implemented within PK did not work at all. PK's IT development manager was aware of the requirements management and even the requirements triage. His effort to negotiated relaxing the term, adding additional resources, changing the requirements (reductions), the method of conversion and change in release date was unsuccessful. PK's President Director insisted that all the new requirements were important and had to be incorporated into the February 1st 2012 release. While at the same time, PK's President Director did not add any new programmers[5] and only promised a new bonus scheme if the developers finished on time[6].

The problems were exacerbated by the actions of non-human actors (Alcouffe, Berland, & Levant, 2008; Latour, 2005; Tatnall & Burgess, 2004; Underwood, 2001). These were the new website itself, the platform (hardware and software) it was to run on and the production environment that refused, despite the repeated efforts of the development team, to work. This had worked well in the development environment but not now. The failure was causing troubles and it was also affecting all of the stakeholders as shown on Figure 2 below. Figure 2 shows how pressure from PK's President Director and Managing Director on the Development Manager and on the team to finish on time while at the same time add more requirements to be built caused the problem. As the result, the development team had limited time to conduct quality assurance and sufficient preparation to launch the application. One non-human actor (the machine) due to poor QA and lack of preparation had failed to work when the application launched.

Limited time and resources had prevented the new website to be thoroughly and rigorously tested. The development team was only able to test the functionality and find errors by asking fellow employees of PK to test the new website. There were no other testing methodologies that could be employed. The development manager's proposal to have parallel conversion was also rejected due to time restrictions. The development manager argued that if his proposal was accepted they could find the problems and fixed them without sacrificing the subscribers who at that time was unable to use PK's website (old or new) for about 48 hours.

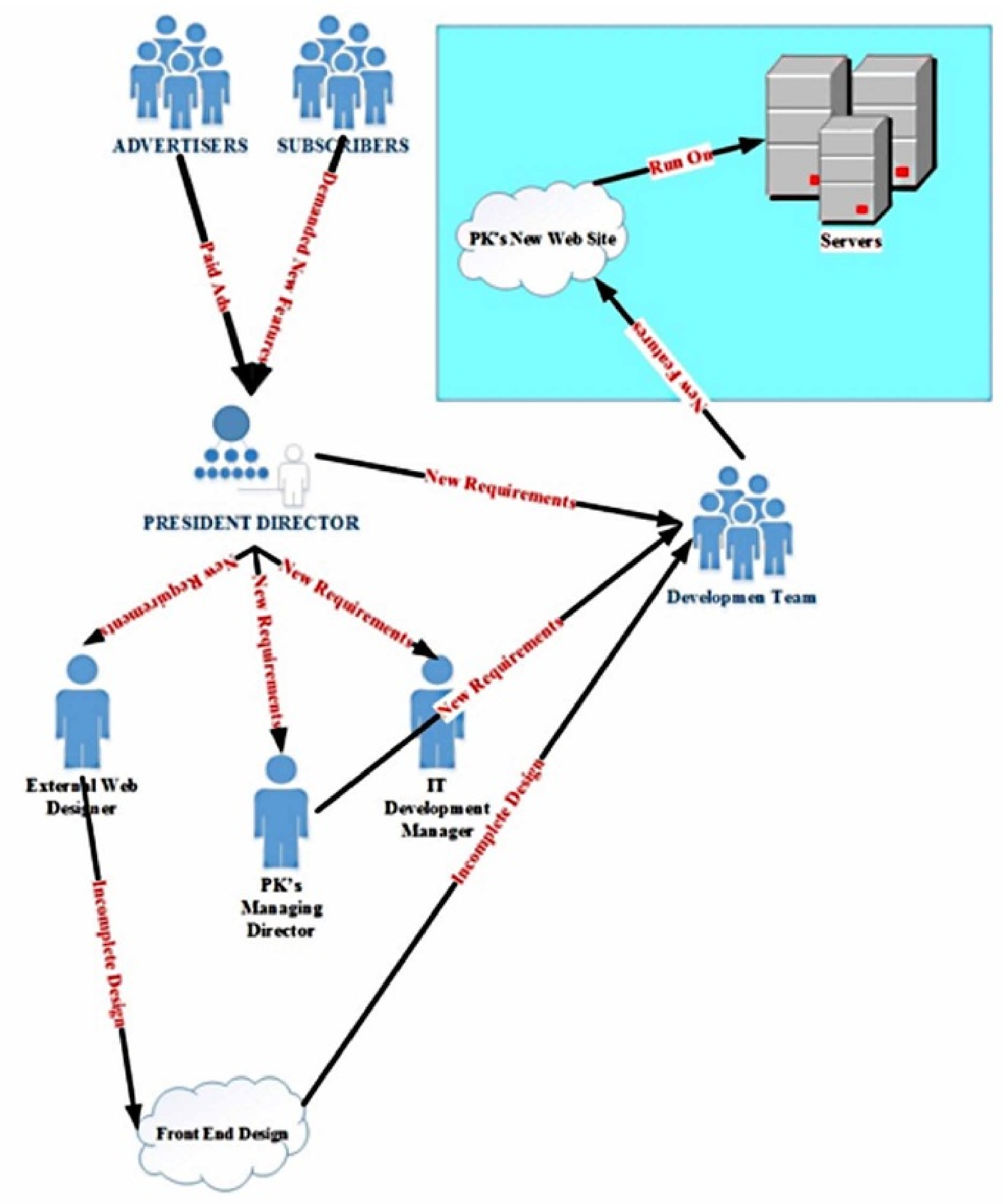
From an Innovation Translation perspective the project did not progress very far through the four moments of translation. The first problem was that the problematisation for the website design kept changing as more and more requirements were added. Scope creep meant that the project's problematisation never came to be stabilised by the project team. Without a stable problematisation, interessement was not able to proceed to achieve useful enrolment. From the viewpoint of the company and the developers, the main interessement came from PK's President Director promising a new bonus scheme if the developers finished on time. This was not seen as very convincing, but along with the team's own internal interessement to do a good job and receive the bonus this did finally produce an enrolment. It is, however, unlikely that mobilisation followed.

Due to scope creep, the principal non-human actor – the website itself took a long time to come into being, and when it did so it did this in a different form than originally proposed. It could in no way be considered to be a stable actor, but its influence was still felt by all the human actors in the project.

CONCLUSION

It can be concluded from both the case description and analysis that scope creep was the main reason for PK's new website launch failure. Scope change requires change in technical resources and time in order to complete the system (Brooks, 1995; A. M. Davis, 2005; Turban & Volonino, 2012). The president director's actions were based on pressure from advertisers who pushed more requirements to be built in limited time and a fixed dead line. The managing director succumbed to the pressure and channelled this to the developer's team. Scope creep could have been prevented and managed using negotiation with all the stakeholders to achieve compromise. If PK's management demanded new features be built into the new website, they should have considered the technical resources and time availability. They should have negotiated with advertisers to allow additional time. In addition to that, they should also have allowed time for testing and fixing, which is inherently part of any systems development. Additional time was also required due to added features being built into the new website. However, due to limited resistant from the development team (the programmers) and limited presence and influence of the Development Manager, the negotiation never took place.

Figure 2. PK's structure Model (based on ANT)



REFERENCES

Alcouffe, S., Berland, N., & Levant, Y. (2008). Actor-networks and the Diffusion of Management Accounting Innovations: A Comparative Study. *Management Accounting Research*, *19*(1), 1–17. doi:10.1016/j.mar.2007.04.001

Andriole, S. J. (1996). *Managing Systems Requirements: Methods, Tools, and Cases*. New York: McGraw Hill.

Avison, D. E., & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, Techniques, and Tools* (4th ed.). Maidenhead: McGraw-Hill.

Balakian, W., Young, K., & Veerapaneni, R. (2002). *Managing E-Business Projects: A Project Manager's Guide for E-Business*. Bloomington: 1st Book Library.

Bardini, T. (1997). Bridging the Gulfs: From Hypertext to Cyberspace. *Journal of Computer-Mediated Communication*, *3(2)*. Retrieved from http://www.ascusc.org/jcmc/vol3/issue2/bardini.html

Brooks, F. P. (1995). The Mythical Man-Month: Essays on Software Engineering (Anniversary ed.). Reading, Massachusetts: Addison-Wesley Pub. Co.

Callon, M. (1986a). The Sociology of an Actor-Network: The Case of the Electric Vehicle. In M. Callon, J. Law, & A. Rip (Eds.), *Mapping the Dynamics of Science and Technology* (pp. 19–34). London: Macmillan Press. doi:10.1007/978-1-349-07408-2_2

Callon, M. (1986c). Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St Brieuc Bay. In J. Law (Ed.), *Power, Action & Belief. A New Sociology of Knowledge?* (pp. 196–229). London: Routledge & Kegan Paul.

Callon, M. (1999). Actor-network Theory - the Market Test. In J. Law & J. Hassard (Eds.), *Actor Network Theory and After* (pp. 181–195). Oxford: Blackwell Publishers.

Davis, A. M. (2005). *Just Enough Requirements Management*. New York: Dorset House.

Davis, A. M., Nurmuliani, N., Park, S., & Zowghi, D. (2008). *Requirements Change: What's the Alternative? Paper presented at the 32nd Annual IEEE International Computer Software and Applications Conference*. doi:10.1109/COMPSAC.2008.216

Davis, F. (1986). *A technology acceptance model for empirically testing new end-user information systems: theory and results. (Ph.D. Thesis)*. Boston: MIT.

Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *Management Information Systems Quarterly*, *13*(3), 318–340. doi:10.2307/249008

Davis, F., Bagozzi, R., & Warshaw, P. (1989). User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science*, *35*(8), 982–1003. doi:10.1287/mnsc.35.8.982

Hoffer, J. A., George, J. F., & Valacich, J. S. (2011). *Modern Systems Analysis and Design* (6th ed.). Boston: Pearson.

Hoffer, J. A., Prescott, M., & McFadden, F. (2007). *Modern Database Management* (8th ed.). Upper Saddle River: Prentice Hall.

Khan, A. (2006). Project Scope Management. *Coastal Engineering*, *48*(6), 12–16.

Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. New York: John Wiley.

Latour, B. (1987). *Science in Action*. Milton Keynes: Open University Press.

Latour, B. (1988). *The Pasteurization of France*. Cambridge: Harvard University Press.

Latour, B. (1996a). *Aramis or the Love of Technology*. Cambridge, Ma: Harvard University Press.

Latour, B. (2005). *Reassembling The Social: An Introduction to Actor-Network-Theory*. Oxford: Clarendon.

Law, J. (1999). After ANT: Complexity, Naming, and Topology. In J. Law & J. Hassard (Eds.), *Actor Network Theory and After* (pp. 1–14). Oxford: Blackwell Publishers.

Markus, M. L. (2002). Power, Politics, and MIS Implementation. In M. D. Myers & D. Avison (Eds.), *Qualitative Research in Information Systems: A Reader* (pp. 19–48). London: Sage Publications.

Meyers, B. C., & Oberndorf, P. (2001). *Managing Software Acquisition: Open Systems and COTS Products*. Boston, Mass.; London: Addison-Wesley.

Myers, M. D. (2009). *Qualitative Research in Business and Management*. Los Angeles: Sage Publications.

Neyland, D. (2008). *Organizational Ethnography*. Los Angeles: Sage Publications.

Nurmuliani, N., Zowghi, D., & Powell, S. (2004). Analysis of requirements volatility during software development life cycle. *Paper presented at the 2004 Australian Software Engineering Conference*, Melbourne. doi:10.1109/ASWEC.2004.1290455

Pfleeger, S. L. (2001). *Software Engineering: Theory and Practice* (2nd ed.). Upper Saddle River, N.J.: Prentice Hall.

Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge* (5th ed.). Newton Square, PA: Project Management Institute.

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

Rogers, E. M. (2003). *Diffusion of Innovations* (5th ed.). New York: The Free Press.

Sommerville, I. (2001). *Software Engineering* (6th ed.). Harlow, England: Addison-Wesley.

Tatnall, A. (2002). Modeling Technological Change in Small Business: Two Approaches to Theorizing Innovation. In S. Burgess (Ed.), *Managing Information Technology in Small Business: Challenges and Solutions* (pp. 83–97). Hershey, PA, USA: IDEA Group Publishing. doi:10.4018/978-1-930708-35-8.ch005

Tatnall, A., & Burgess, S. (2004). Using Actor-Network Theory to Identify Factors Affecting the Adoption of E-Commerce in SMEs. In M. Singh (Ed.), *E-Business Innovation and Change Management* (pp. 152–169). Hershey, PA, USA: IDEA Group Publishing. doi:10.4018/978-1-59140-138-4.ch010

Thakurta, R. (2013). Impact of Scope Creep on Software Project Quality. *Vilakshan. XIMB Journal*, *10*(1), 37–46.

Thakurta, R., & Ahlemann, F. (2011). Understanding Requirement Volatility In Software Projects. *Engineering Management Journal*, *23*(3), 3–7. doi:10.1080/10429247.2011.11431900

Turban, E., & Volonino, L. (2012). *Information Technology for Management* (8th ed.). New York: Wiley.

Underwood, J. (2001). *Meaning in IS Development: Understanding System Requirements and Use with Actor-network Theory (Doctor of Philosophy)*. Wollongong: University of Wollongong.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *Management Information Systems Quarterly*, *27*(3), 425–478.

Zowghi, D., & Nurmuliani, N. (2002). A Study of the Impact of Requirements Volatility on Software Project Performance. *Paper presented at the The 9th Asia Pacific Software Engineering Conference*, Gold Coast. doi:10.1109/APSEC.2002.1182970

ENDNOTES

[1] All names have been changed and coded to protect participants' privacy and confidentiality.

[2] The web designer was living in Sydney throughout the project

[3] Error 2003 is an error code which informed that the database server was out of connections and so it could not be reached by the client

[4] As in PK's organizational structure in Figure 1

[5] Which might not be useful in the later stage of project anyway

[6] Later the developers were denied a bonus citing failed to deliver on time and failure of launch as reasons

Samiaji Sarosa is Senior Lecturer of Information Systems at Accounting Department, Atma Jaya Yogyakarta. He holds Bachelor in Accounting Degree from Atma Jaya Yogyakarta University, Master of Information Systems from University of Wollongong, and PhD in Computing Sciences from University of Technology, Sydney. Apart from his main duties as academia, he is also well immersed into the current state of IT industry. He has been involved as manager and consultant for various systems development projects for small to big companies and event national level.

Arthur Tatnall is an Adjunct Professor in Information Systems in the College of Business at Victoria University. For his PhD he used actor-network theory to investigate the adoption of Visual Basic in the curriculum of an Australian university. Prior to this while completing in his MA, he investigated the origin of the information systems discipline in Australia. He is currently Editor-in-Chief of the Education and Information Technologies journal and the International Journal of Actor-Network Theory and Technological Innovation.

Artikel-5.pdf

ORIGINALITY REPORT

18%	17%	10%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to RMIT University Student Paper	6%
2	studylib.net Internet Source	3%
3	www.safaribooksonline.com Internet Source	2%
4	icts.if.its.ac.id Internet Source	1%
5	is.its.ac.id Internet Source	1%
6	core.ac.uk Internet Source	1%
7	Submitted to University of Lincoln Student Paper	1%
8	dl.ifip.org Internet Source	1%
9	Submitted to Monash University Student Paper	<1%
10	www.irma-international.org Internet Source	<1%

11	Arthur Tatnall. "chapter 4 Innovation Translation, Innovation Diffusion and the Technology Acceptance Model", IGI Global, 2011 Publication	<1%
12	Aguirre-Urreta, Miguel I. Marakas, Georg. "Exploring choice as an antecedent to behavior: incorporating alternatives into the technology accept", Journal of Organizational and End User C, Jan-March 2012 Issue Publication	<1%
13	Submitted to Colorado Technical University Online Student Paper	<1%
14	Bill Davey, Arthur Tatnall. "Using ANT to Guide Technological Adoption", International Journal of Actor-Network Theory and Technological Innovation, 2012 Publication	<1%
15	brage.bibsys.no Internet Source	<1%
16	www.tandfonline.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	< 8 words
Exclude bibliography	On		