

**A METHODOLOGY FOR DEVELOPING  
APPLICATIONS ON SOCIAL NETWORKING SITES**

Lahti University of Applied Sciences

Degree Programme in Business Information Technology

Thesis

Winter 2009

Ma, Jun

Lahti University of Applied Sciences  
Degree Programme in Business Information Technology

MA, JUN: A Methodology for Developing  
Applications on Social Networking Sites

Bachelor's Thesis of Degree Programme in Business Information Technology, 54  
pages, 2 appendices

Winter 2009

## ABSTRACT

---

These recent years witnessed a substantial emergence of social networking sites (SNSs) which some even refer as an arising phenomenon. Besides the main function of communication of SNSs, the applications integrated within the SNSs are of great popularity and thus help gain more users. Currently, there is no standard for social application development. In this context, the thesis proposes a methodology based on the traditional system development life cycle, with a view to help the developers in implementation of the applications.

The thesis is identified as a theoretical creative research. The study is conducted based on the qualitative approach. To be more specific, a series of experimental actions in developing a new social application are taken in order to examine the operation in practice. The methodology offers a collection of models, tools, procedures to assist the experiment.

The empirical part consists of a whole development life cycle of the proposed methodology and illustrative actions. The six stages include feasibility study, end-user investigation, developing specification, application analysis and design, application implementation, application publication, review and maintenance. With its own plans, timelines, controls and monitoring procedures, the activities of each stage are defined as their associated end products, which facilitates the use of project management techniques.

In conclusion, evaluation of the created methodology is given. The thesis thoroughly evaluates in terms of philosophy, model, techniques and tools, scope, outputs, practice and product, based on which the future possible development is introduced.

Key words: application, methodology, SNSADLC, social networking sites, system development

## TABLE OF CONTENTS

1	INTRODUCTION	6
1.1	Background	6
1.2	Statement of the problem	6
1.3	Objective	3
1.4	Research methodology	3
1.5	Scope and limitations	5
1.6	Thesis structure	6
2	SOCIAL NETWORKING SITES (SNS)	7
2.1	Introduction	7
2.1.1	A history of Social Networking Sites	9
2.1.2	The global phenomenon and the development	12
2.2	Applications on social networking sites	13
2.2.1	Advantages and disadvantages	15
2.2.2	Social Application developers	16
3	INFORMATION SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)	17
3.1	Overview	17
3.2	Systems development stages	18
3.2.1	Feasibility study	18
3.2.2	System investigation	19
3.2.3	Systems analysis	20
3.2.4	Systems design	21
3.2.5	Implementation	22
3.2.6	Review and maintenance	24
3.3	The strengths and weaknesses of the SDLC	25
3.3.1	The strengths of the SDLC	25
3.3.2	The weaknesses of the SDLC	26
4	A METHOD TO DEVELOP APPLICATIONS ON SOCIAL NETWORKING SITES	28
4.1	Introduction to the application development life cycle	28
4.2	Social networking sites application development life cycle	29
4.2.1	Feasibility study	29
4.2.2	End-User investigation	32
4.2.3	Developing specification	34

4.2.4	Application analysis and design	37
4.2.5	Application implementation	40
4.2.6	Application publication, review and maintenance	43
5	CONCLUSION	47
5.1	Evaluation of the SNSADLC methodology	47
5.1.1	Philosophy	47
5.1.2	Model	49
5.1.3	Techniques and tools	49
5.1.4	Scope	50
5.1.5	Outputs	50
5.1.6	Practice	50
5.1.7	Product	51
5.2	Further development of the SNSADLC methodology	52
6	SUMMARY	53
	GLOSSARIES	55
	REFERENCES	56
	LIST OF FIGURES	61
	APPENDICES	62

# 1 INTRODUCTION

## 1.1 Background

Since their introduction, social networking sites (SNSs) such as Facebook, MySpace, Cyworld, and Bebo have attracted millions of users, many of whom have integrated these sites into their daily practices (Boyd & Ellison 2007). At the time being, there have been 122 million visitors surfing Facebook and 56 million using MySpace (Compete 2009). Among these users, more than 70% of them engage with the applications on the social networking platform (Facebook Statistics 2009).

This has been made possible because sites such as Facebook and MySpace have opened up their API (Application Program Interface) to third-party developers who design applications for use on the sites. Each social networking site has its own brand of applications including games and activities. Meanwhile, some of these applications have proved very popular among site users and have helped to maintain the momentum of social networking sites use. (Office of Communications 2008)

## 1.2 Statement of the problem

While applications on social networking sites are moving into top gear, the specialized methodology to develop such applications is still in a state of emptiness. A methodology for developing information systems has been defined by Avison *et al.* (2003, 20):

*A collection of procedures, techniques, tools, and documentation aids which will help the system developers in their efforts to implement a new information system. A methodology will consist of phases, themselves consisting of sub-phases, which will guide the system developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects.*

In fact, few programmers would follow any formal methodology. Usually they would use rules-of-thumb and rely on experiences. They argue that a formal methodology is not always necessary for developing applications. For instance, if the project is small or non-critical, the use of a formal methodology may not be important and may even tend to impede the delivery schedule (Davidson, Hedrich, Leavy, Sharp & WA 2002).

However, without a formal methodology, it is difficult to estimate more complex applications such as on which date they would be operational, the result is the applications are frequently behind schedule. Moreover the programmers are usually overworked and frequently spend a very large proportion of their time correcting and enhancing the applications that were operational. (Avison *et al.* 2003, 19)

For this reason, a more formal methodology will enhance the product and the delivery schedule when a project reaches critical mass (Davidson 2002). In a same manner, despite human factors, the formal methodology which is keeping with both the rule of application development and the features of SNS will provide an effective application in a reasonable amount of time.

Figure 1 illustrates the relationship between a formal methodology and a proper application impersonally.

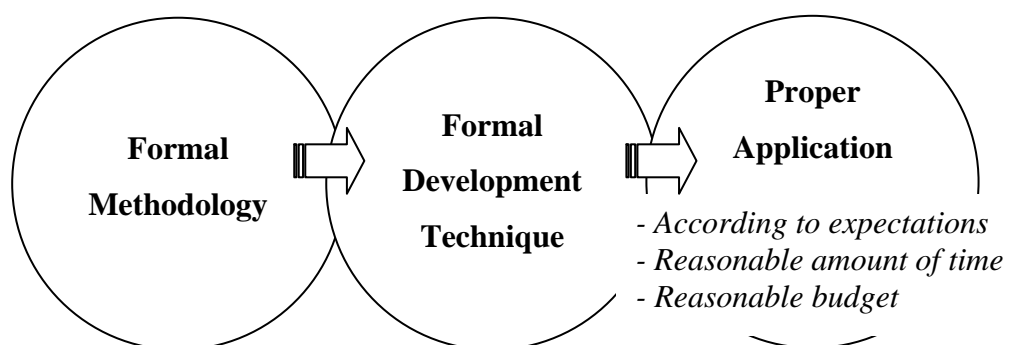


Figure 1: The relationship between a formal methodology and a proper application impersonally

### 1.3 Objective

The purpose of this study is to create an appropriate methodology for developing applications on social networking sites (SNSs). Therefore the SNS application developers and the potential people with interests in developing SNS applications can use it as a guidebook in their future development work.

By using this methodology in implementing, developers are more likely to complete the applications on SNSs with less time, fewer budget and more interactive elements. This can be achieved since the methodology concerns the features of both system development life cycle and social networking sites adequately.

The existing traditional system development life cycle (SDLC) and other variant methodologies are either limited in terms of usage scope or outdated that are no longer suitable for web development (Cutter, 2000). Meanwhile, applications on social networking sites are gaining more and more popularity due to their obvious features of social communication, low entry barriers and casualness. The social graph or friend network makes playing applications built on SNSs more fun than playing them with real-life friends. (Playfish Case Study 2009)

The proposed methodology then inherits the frame of development life cycle, extracting a proper procedure starting from feasibility study to maintenance, and being involved in new elements such as understanding the SNS user data.

“Practice is the sole criterion for testing truth”, thus a series of simple actions are conducted synchronously by following each step of the proposed methodology aiming to examine its pros and cons.

### 1.4 Research methodology

Research is rather about answering unanswered questions or creating that which does not currently exist than just a process of gathering information (Goddard & Melville 2004, 1). According to Kumar (2005, 6), research procedures can also be employed to formulate intricate theories or laws that govern our lives. Therefore this part aims to clarify the research methodology that is employed in this thesis.

Deferent perspectives with regards to research are discussed in the next few paragraphs including research type, research design and method of data collection.

Above all, the thesis is identified as a theoretical creative research. Theoretical creative research involves the development or creation of new models, theorems, procedures, etc (Goddard *et al.* 2004, 8). While Kumar (2005, 9) categorizes it into pure research which containing such as, developing a sampling technique that can be applied to a particular situation and developing a methodology to assess the validity of a procedure.

Next, the research design plans the procedure that is adopted by the writer to answer questions validly, objectively, accurately and economically (Kumar 2005, 84). On this basis, the thesis is therefore written with theoretical study and empirical study. Based on the theoretical study, a theory or a hypothesis is developed first, and then a research strategy is designed to test the hypothesis.

Moreover, a method for collecting data is used to find answers to the research question. The two basic approaches to inquiry data are quantitative and qualitative methods. According to Kumar (2005, 12), quantitative research is a method to quantify the variation in a phenomenon, situation, problem or issue. While Creswell (2003, 75) argues that qualitative research is best used to explore a concept or phenomenon when the variables and theory base are unknown.

The thesis is written based on qualitative methods. Action processes such as the development of advisory conversations are common qualitative methods to collect data (Flick, Kardorff & Steinke 2004, 13). Therefore an action is applied in the thesis to try and see if the hypothesis is good.

Based on the proposed methodology, the action to develop an application on social networking site Facebook is conducted in each life cycle stage. In Chapter five, a thorough measurement of the methodology is given according to the actions' performance.

Figure 2 summarizes the research process:



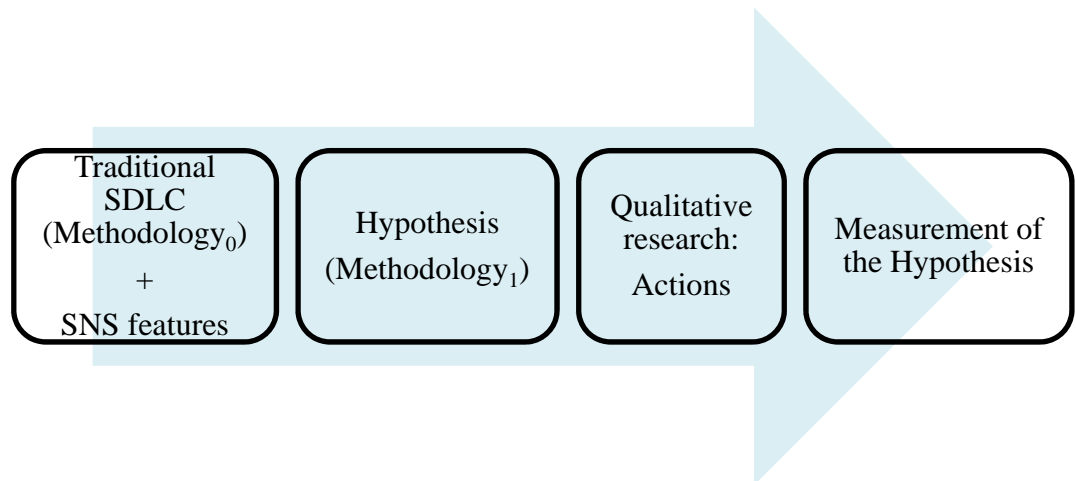


Figure 2: Research process

### 1.5 Scope and limitations

As introduced in the thesis title, a methodology is proposed and analysed in the study. It focuses on the procedures of developing applications. The procedures separated into six steps which are also known as life cycle aim to guide the developers implementing new social applications. Moreover, in a particular scope of application development area, this study only focuses on the application on social networking sites. Therefore the proposed methodology may only be suitable when applications are to be developed on SNSs.

The study assumes all the relevant technologies for web 2.0 are capable of. These web technologies are thought as basic abilities to the developers. There are a variety of web technologies with similar use and different developers might use totally different ways of technologies. Without a good methodology, there is no problem to create applications by suitable technologies. However, as “no land without lord”, it is impossible to create any applications without any technologies no matter how good the methodology is.

Lastly, as there are a series of actions conducted in the thesis, it usually requires approximately three months in minimum for most cases (Zimmerman 2008, 58). Unfortunately, there is not enough time to take a three-month long action, instead only less than one month is used. In addition, the type of the application in fact is limited to just for fun while a dozen of types existing on SNSs, such as games, business, entertainment and sports.

## 1.6 Thesis structure

The thesis structure is comprised of two parts which are theoretical review and empirical study. It goes through the general methodology of design research described by Vaishnavi & Kuechler (2004) which contains awareness of problem, suggestion, development, evaluation and conclusion.

The theoretical review is a part of problem awareness. It starts with Chapter two – general introduction of social networking sites, subdivided into the first part reviewing the development of social networking sites, and the other part focusing on the differences between building social applications and traditional ones.

In Chapter three, the principle study on the traditional system development life cycle is carried out by describing the six stages of the life cycle in detail and then giving its merits and drawbacks.

On that basis, Chapter four suggests and develops a methodology based on system development life cycle. This methodology is mainly concerning developing applications on social networking sites as reviewed in Chapter two, including empirical study and a series of actions.

Finally, Chapter five concludes the findings from all afore discussed issues, the use of the formulated methodology on the actions is analyzed in order to give an evaluation. In addition, a summary is provided in Chapter six to finish off the thesis.

The following figure 3 illustrates the thesis structure, with note that numbers attached with some titles indicating chapters.

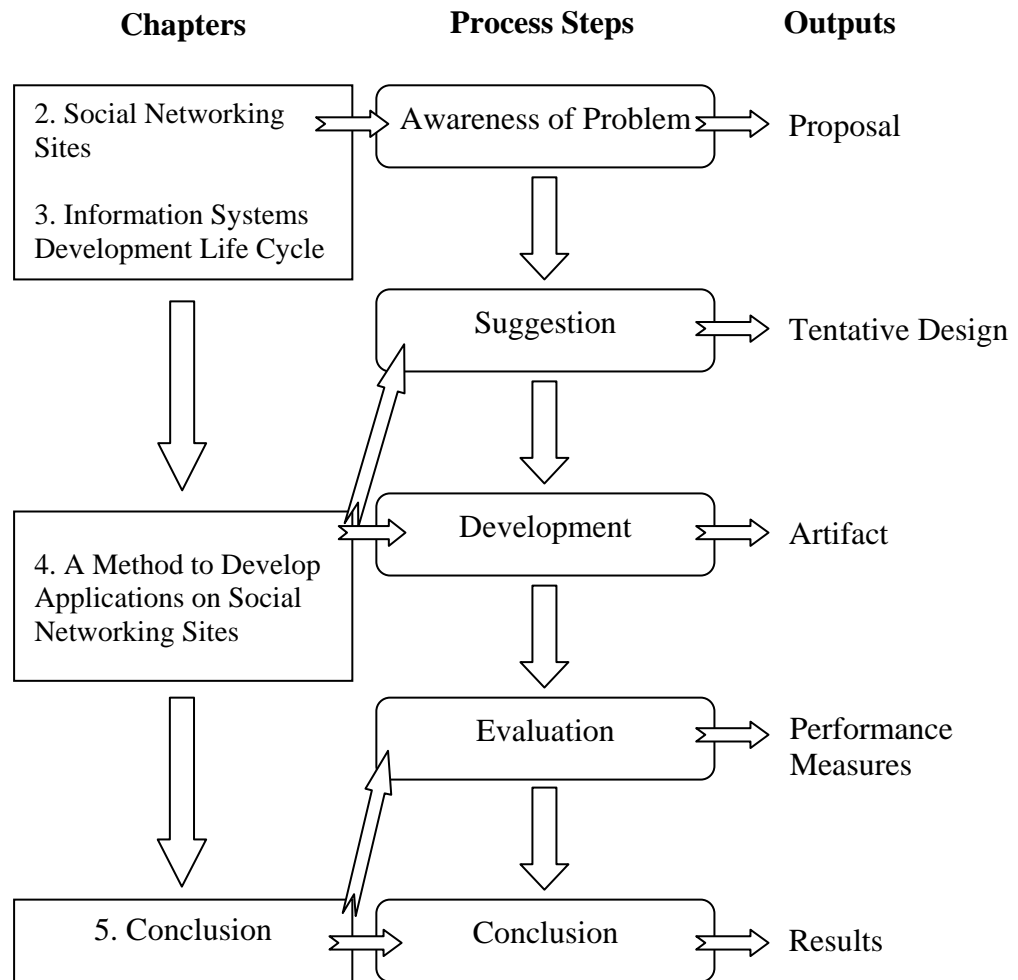


Figure 3: Thesis structure (Vaishnavi *et al.* 2004)

## 2 SOCIAL NETWORKING SITES (SNS)

### 2.1 Introduction

Social networking, the art of meeting and building contacts on the Web, is an increasingly popular way to gain personal and professional contacts, make new friends, develop professional relationships, promote personal or business Web sites, and even find a new job (Warner 2008, 277). Over the past few years, social networking sites (SNS) have received growing media attention. At many high schools and colleges, every student has a profile on at least one of the SNS such as MySpace and Facebook. (Farkas 2007, 109)

According to Farkas (2007, 109), people do not specifically use the sites to discuss common interests, but instead, they use SNS to display their identity and social network publicly and make new connections. This indicates a change in the

way people build identity online and get to know each other. As shown in Figure 4, if A and B know each other very well, meanwhile, A and C also know each other very well, then it is highly likely that B and C also know each other (Rettberg 2008, 59).

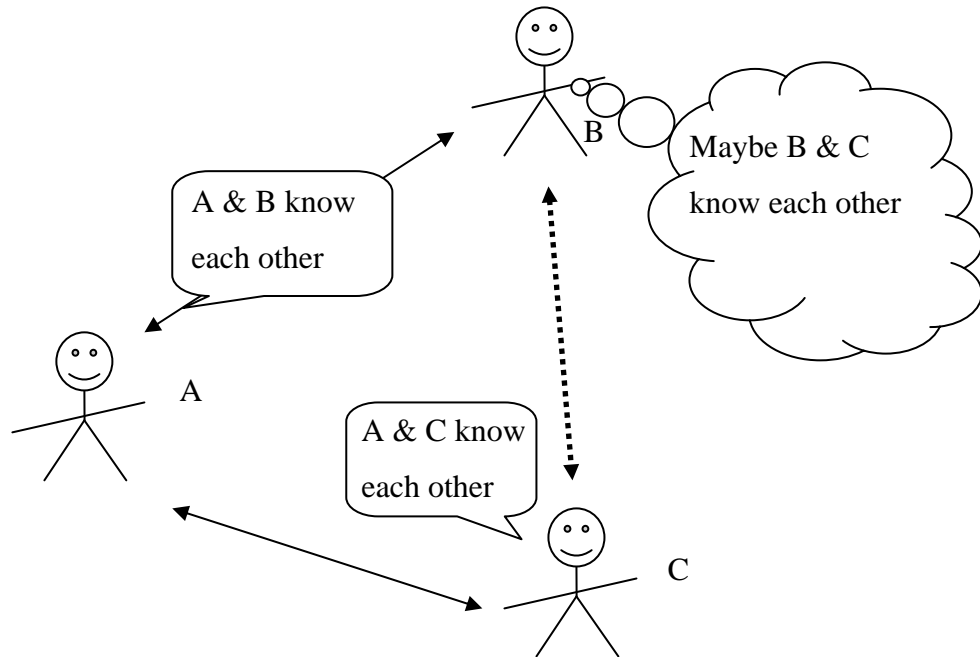


Figure 4: Identity on and social network on SNS

In the mid-1960s, Stanley Milgram experimented to determine the structure of social networks. Milgram, a social psychologist, had a tremendous impact on the field of network theory, taking it from the realm of mathematics into the social world. The experiments are named from the common exclamation “What a small world!” when people realize they both know the same person. In the experiments, Milgram’s team asked each participants to send a letter to a stockbroker by forwarding it to a person they knew who was more likely than themselves to know the target. Although many of the letters never reached their target and the project had significant problems, the letters that did attain their goal had an average of six connections in the chain, leading the popular idea of there being “six degrees of connection” between any two people in the world. (Farkas 2007, 110)

In terms of Social networking sites, the theory posits the idea that one person’s connections to other people are more important than the person’s attributes. Following this logic, people can be judged for better or worse by their friends and

acquaintances. On the other hand, people's social networks can help them succeed in life, love and business. Therefore, the goal of SNS is to allow users to develop an online identity and grow a social network through establishing business contacts, dating, or simply making new friends, while other kinds of online communities only enable communication, provide support and promote action. (Farkas 2007, 110)

Following are some of the most popular social networking sites and their brief description.

**Facebook (www.FaceBook.com):** The site was originally considered a vanity site and a place for students to connect and trade stores, but it is growing fast now that anyone who wants to participate can join the site's community. Its professional power is growing with its ever-expanding audience. (Warner 2008, 277)

**MySpace (www.MySpace.com):** One of the all-time most popular social networking sites, MySpace makes it easy to create a profile site, add music, write a blog, and post as many photos as possible to share with the world. Its huge online audience is a popular place for musicians, performers, and many others to promote themselves and their Web sites. (Warner 2008, 277)

**Bebo (www.Bebo.com):** It has been nothing but good news and consistent growth. Extraordinarily popular in the UK, Bebo is beginning to give some of the other social networking sites a real run for their money. Like other sites, users can personalize their profiles, add apps, set privacy settings and join groups. (Social Networking Websites Review 2009)

### 2.1.1 A history of Social Networking Sites

Social networking started with Generation X (a term used to refer to the generation born after the baby boom ended, extending from the early-to-mid 1960s to late 1970s), and dozens of social networking sites are geared primarily toward that demographic. Friendster (www.friendster.com), the first site that really made a name for itself, opened to the public in 2002 and attracted 1 million users in less than a year. The site was initially geared toward 25- to 35-year-old urbanites, but as their social networks grew, friends around the globe began to join and network. (Farkas 2007, 112)

Anyone can join Friendster; they just need to create an account, create a profile, and add photos. Members can add their friends to their network, but a friend can only be added if that friend adds you as well. This is a measure designed to prevent people from adding hundreds of “friends” to their profile whom they do not really know. Once you have a group of friends, Friendster will show you your “second-degree” friends, or people who are listed as friends of your friends. The assumption is that you are more likely to have something in common with your friends’ friends than with a random person you might meet at a traditional dating site or networking event. (Farkas 2007, 112)

Friendster lets users post testimonials—things your friends write about you that are in many cases more telling than a profile. Friendster also lets users create individual blogs, share photos and videos, post to discussion boards, and chat with other Friendster. Finally, the site lets people create and join groups where they can meet members outside of their social circle with similar interests. This group membership information is geared to creating a richer profile, which is central to the site. (Farkas 2007, 112)

In 2003, social networking was all the rage among those in their 20s and 30s. By 2005, however, the frenzy over social networking among members of Generation X had calmed down significantly. Not only did usage of Friendster decline, but a number of social networking sites shut down because of the lack of a loyal user base. Surprisingly, the trend is quite the opposite among members of Generation Y (The Millennial Generation, young people born in 1980-2000). Social networking sites designed specifically for that demographic have flourished. Young people see sites like MySpace ([www.myspace.com](http://www.myspace.com)) and Facebook ([www.facebook.com](http://www.facebook.com)) as spaces designed for them, where they can express themselves freely and connect with friends online. (Farkas 2007, 113)

MySpace began in mid-2003 after Friendster had already developed a significant membership base. MySpace quickly rolled out many features that distinguished it from Friendster and made it very attractive to teenagers and young adults. One of its most significant features is the ability to customize profiles. MySpace members can use HTML and CSS to personalize the look of their profiles and can add video and audio clips for visitors. (Farkas 2007, 113)

Hundreds of outside sites help MySpace users customize their profiles. As in Friendster, a member's friends are displayed in his or her profile. Instead of member testimonials, a comments section attached to each profile works much the same as a bulletin board where users post messages to each other. Blogs are also an integral part of MySpace. Each member has a personal blog with posts displayed on their profile. Because of these features, MySpace attracts those interested in self-expression as well as networking. By mid-2006, about 61 million people were registered MySpace users, and in July 2006, it became the No.1 Internet site in terms of traffic. (Farkas 2007, 113)

The image shows a screenshot of a Facebook profile for a user named Ma Jun. The interface includes a top navigation bar with 'facebook', 'Home', 'Profile', 'Friends', and 'Inbox'. The user's name 'Ma Jun' is displayed, along with tabs for 'Wall', 'Info', 'Photos', 'Boxes', 'Links', 'Video', and a plus sign. Below the profile picture, there are links for 'View Photos of Me (3)' and 'Edit My Profile'. A quote 'idea is always more important than technique' is visible. The profile is divided into several sections: 'Basic Information' (Gender: Male, Birthday: 02 September 1983, Hometown: Shanghai, China, Relationship Status: In a relationship, Interested in: Women), 'Personal Information' (Interests: API design, Programming; Favourite Music: POP), 'Contact Information' (Email: jun.ma@lpt.fi, Mobile number: +358 449241057, Current address: Rauhankatu Lahti, Finland 15110, Yahoo!: Jun Ma, Website: http://Coming, http://soon...), and 'Education and Work' (University: Lahti University of Applied Sciences '06 Business Information Technology... Business Information Technology...).

Figure 5: A Facebook profile contains photos, a list of friends, and “The Wall.”  
(Reproduced with permission. ©2009 Facebook)

Facebook, as shown in figure 5, is undoubtedly the representative of the newer social networking sites. It was founded in early 2004 to bring college students together online (though Facebook now allows anyone can join the site's community). As with all social networking sites, the profile is central to Facebook: it includes photos, a list of friends, and “The Wall,” where friends can

post messages. People can make new friends by viewing the profiles or friends or their friends, or by searching for people who are taking the same classes. Users can also create groups. (Farkas 2007, 115)

As of August 2009, Facebook attracted 122 million unique visitors (Compete 2009). The site, which also earns revenue from companies looking to market to its members, has even allowed companies including Apple and Electronic Arts to create groups within Facebook to communicate directly with their potential consumers. (Farkas 2007, 115)

### 2.1.2 The global phenomenon and the development

While MySpace attracted the majority of media attention in the U.S. and abroad, social networking sites were proliferating and growing in popularity worldwide. Friendster gained traction in the Pacific Islands, Orkut ([www.orkut.com](http://www.orkut.com)) became the premier SNS in Brazil before growing rapidly in India, Mixi ([www.mixi.jp](http://www.mixi.jp)) attained widespread adoption in Japan, LunarStorm ([www.lunarstorm.se](http://www.lunarstorm.se)) took off in Sweden, Dutch users embraced Hyves ([www.hyves.nl](http://www.hyves.nl)), Grono ([www.grono.net](http://www.grono.net)) captured Poland, Hi5 ([www.hi5.com](http://www.hi5.com)) was adopted in smaller countries in Latin America, South America, and Europe, and Bebo ([www.bebo.com](http://www.bebo.com)) became very popular in the United Kingdom, New Zealand, and Australia. Additionally, previously popular communication and community services began implementing SNS features. The Chinese QQ instant messaging service instantly became the largest SNS worldwide when it added profiles and made friends visible, while the forum tool Cyworld ([www.us.cyworld.com](http://www.us.cyworld.com)) cornered the Korean market by introducing homepages and buddies. (Boyd *et al.* 2007)

According to a report carried out by Office of Communications (2008), there are several factors that help to explain the recent growth of today's social networking sites and the mainstream use of similar technologies. The following is a list of some of the biggest changes.

- Home internet penetration has increased as have connection speeds
- Increasing ICT confidence
- User-friendly programmes



- Communication based around social relationships
- Social networking sites are part of the wider Web 2.0 context
- Applications have increased the versatility of social networking sites

Among the above list, applications which operate on social networking sites are a new category of fun (O'Reilly Home 2008). SNS such as Facebook has released its developer API and was immediately interested by over 1,000 developer-users in the first four days (Baloun 2007, 105).

Companies such as Playfish, one of the largest and fastest growing social gaming companies, are creating game applications for people to play together using social networking sites like Facebook as its platform. Statistics show that there were 25 million users played the games designed by Playfish on Facebook. (Playfish Case Study 2009)

In tradition, people use an application created for a particular console, and then look for local friends or new friends online who also enjoy the application. Generally only the most dedicated users do this. Today, social networking sites act as social platforms with low barriers to entry like Facebook are making casual application easier and more accessible for all users. (Playfish Case Study 2009)

## 2.2 Applications on social networking sites

Social networking sites have become increasingly versatile (Office of Communications 2008). They are not limited to messaging, communicating and displaying networks. Nearly all sites allow users to post photos, videos and often music on their profiles and share them with others (Farkas 2007, 112). Beyond these, users can even keep up with favorite bands and add applications such as games, quizzes and virtual gift giving.

Started by Facebook, social networking sites have increasingly opened their interfaces to third-party applications. This has led to an expansion in what users can do on social networking sites, from taking part in film trivia quizzes to playing mini-games. Established applications and functions have also found their

way onto social networking sites in recent years. It is now possible to make voice calls through certain sites using Skype (A software that allows users to make voice calls over the Internet), while Bebo has led the way in incorporating video drama with its KateModern series. Bebo has also signed a deal with the BBC and Channel 4 (British TV broadcasters), among others, to provide some of their broadcast content to Bebo users. (Office of Communications 2008)

There are a vast number of computer applications which users can add to their profiles at the click of a button. This has been made possible because sites such as Facebook and MySpace have opened up their interfaces to third-party developers who design applications for use on the site. Each social networking site has its own brand of games and activities which the user can download. These range from applications which let you turn your friends into zombies, to maps on which you can record all the places that you have visited. Some of these applications have proved very popular among site users and have helped to maintain the momentum of social networking sites use. (Office of Communications 2008)

To better understand the relationship or communication among users, social networking sites and applications built for it, now we have a look at their data flow. In the social application model, the social networking sites send social data to the application server (the third-party server) where the web pages are rendered for the users with the existing presentation layer. The data flow is shown in figure 6: (Social Application Tutorial 2009)

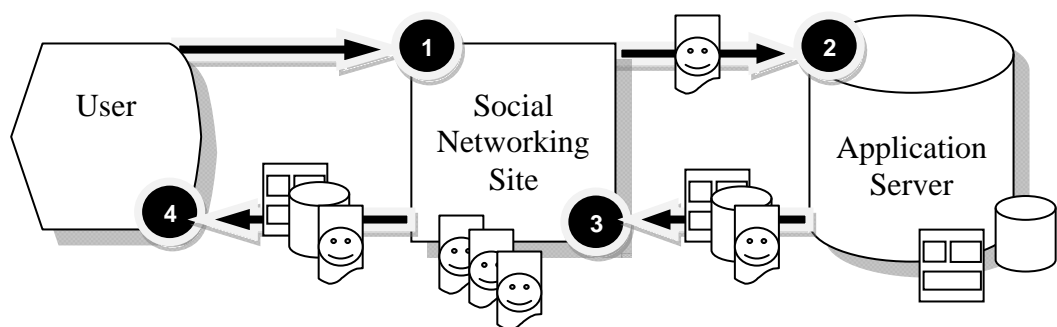


Figure 6: Data flow among users, social networking sites and social applications

Based on figure 6, there are four steps through the data flow which are explained as following: (Social Application Tutorial 2009)

Step 1: The user requests an application view.

Step 2: The Social Networking Site sends social data to the application server.

Step 3: The third-party server combines social and application data, returning HTML and JavaScript.

Step 4: The Social Networking Site sends the content to the user.

### 2.2.1 Advantages and disadvantages

Bernoff (2008) points out that those applications on social networking sites are about consideration, not awareness. People are connected with each other through social networks, a person may resist advertising if his finance is tight, however if his friend tells him that the new movie is really worth watching, that is more persuasive than advertising. Playfish case study (2009) states that applications built on social networking sites are even more fun when playing them with real-life friends. The friend network in social networking sites is termed as social graph, since graphs are visual representations of numbers, but also can be visual representations of relationships (Feiler 2008, 6). The quality of SNS' social graph enables users to compete, collaborate and engage with their friends. Therefore Bernoff (2008) believes that the consideration phase is more important than awareness, especially in a recession, this would result in the failure of advertising and the success of social applications.

Bernoff (2008) continue indicates that developing applications on social networking sites is cheap or nearly free (e.g. Ning, Facebook). Comparing with those applications built in traditional ways, it saves a lot of budgets which are used to pay significant sized ad campaign.

Finally, information through social networks will be spread in the same way that viruses causing disease are spread. Because of the links among friends in SNS, information can spread in this way. The adoption of a new application can be phenomenally fast in this environment. In most cases, if one person adds an application, that news can be passed along to friends automatically and they may choose to add the application themselves. (Feiler 2008, 5) As of February 2009,

Playfish had over 50 million registered players, out of which around four million play every day. It currently has five of the top ten games on Facebook and its users spend altogether more than two billion minutes playing every month. (Playfish Case Study 2009)

On the other hand, social applications also received a few criticisms. Wanjiku (2009) argues that viral potential of applications on social networking sites is a drawback. On the social networking platform, news or items to share can spread like bushfire, while if the news is negative or bad, it can be disasters to the application developers. In other words, it is hard to undertake damage control in dealing with social networking sites and the good reputation might be soiled within hours.

Besides, Wanjiku (2009) implies another disadvantage of developing applications on social networking sites is the need for intensive labour. Because once the application is published, it is possible to have a very big crowd and it can be difficult to meet the needs of all people. This is the essence of social networking and if the problems are not handled properly, the users might lose faith and leave.

### 2.2.2 Social Application developers

Since Facebook opened its APIs to the third-party developers in 2006, other social networking sites such as Myspace, LinkedIn, Friendster and Orkut all followed suit and opened their code to kick off a spurt of innovation in social networking (Kharif, 2007). “Part of what’s exciting about a developer community is you don’t know what people are going to do,” says Lucian Beebe, director of product management at LinkedIn. As a result of opening APIs to developers, a flood of creativity has been produced (Baloun 2007, 105).

According to Smithh (2009), there are several reasons that attract developers go for social networking sites:

- The large user base of SNS such as 300 million of Facebook allows for optimal promotion and marketing of social applications. With a diverse user base, development of social applications can be tailored towards specific demographics. (Smithh 2009)

- Strong Viral distribution is a prominent feature of SNS that notifies users every time an application is used by their friends. For instance, if one user has 30 friends and he plays Farmville (a popular application on SNS), then his activity along with his scores will be published as story on all of his 30 friends' Home page. (Smithh 2009)
- Reliability of the SNS platform for application developers is very important. Every application requires separate and customized code for its incorporation on a SNS platform. For instance, the game application "Vampires" (a popular application on SNS) can be played on both MySpace and Facebook but both have the application integrated with a different coding. (Smithh 2009)
- Another key attraction is the funds where developers have a chance to submit their applications in order to qualify for investments to grow their venture. Facebook's platform is leveraging the social gaming industry by providing funds, mentorship and marketing to the people who make contribution to that field. (Smithh 2009)

### 3 INFORMATION SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)

#### 3.1 Overview

SDLC stands for System Development Life Cycle or Software Development Life Cycle (Folleth 2002). It is used to describe functional systems development activity, to gain control of the complexities of systems development, and to ensure the needs of customers and users are the basis for technical activity (U.S. House of Representatives 1999, 4.).

The SDLC has made a great impact on developing information systems as a general approach. Stages of the SDLC may vary from different references, for instance, "conventional systems analysis", "traditional systems analysis", "classical life cycle model", "linear sequential model" and "waterfall model". However, the most similar point is that they all start from the feasibility study stage and end at the review stage. The basic process of the SDLC is shown in figure 7:

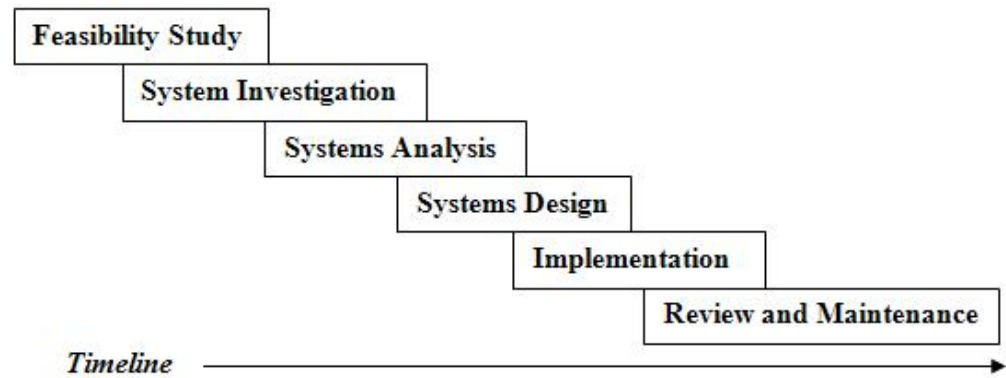


Figure 7: Basic structure of SDLC (Avison *et al.* 2003, 27)

In the last two decades, computers have been widely used in a variety of industries. Even multi-million-dollar manufacturing facilities are run by computers (Folleth 2002). Therefore people need to seriously take the SDLC into consideration. The stages of SDLC are regularly and necessarily used in investigating activities such as “Do we have sufficient hardware” or “Do we have the required skills” to handle the system.

Each stage of the SDLC requires documentation, reporting, and approval. This assures that a project will not lose control by changing the direction or becoming a financial black hole. (Folleth 2002) All the six stages mentioned will be discussed in the next section.

## 3.2 Systems development stages

### 3.2.1 Feasibility study

The feasibility study considers the possibility of solving a particular problem by spending a reasonable cost (Doyle 2001, 98). Usually feasibility study is undertaken when the old system is overloaded, too many errors occur, new technologies are innovated or changes are made in the work type. If the feasibility study is taken improperly, it could result in the termination of a valid project, or the continuation of a project that is not economically feasible, nor technically (Kerzner 2009, 420).

According to O'Brien & Marakas (2008, 379), feasibility study should include the terms of description as shown in figure 8:

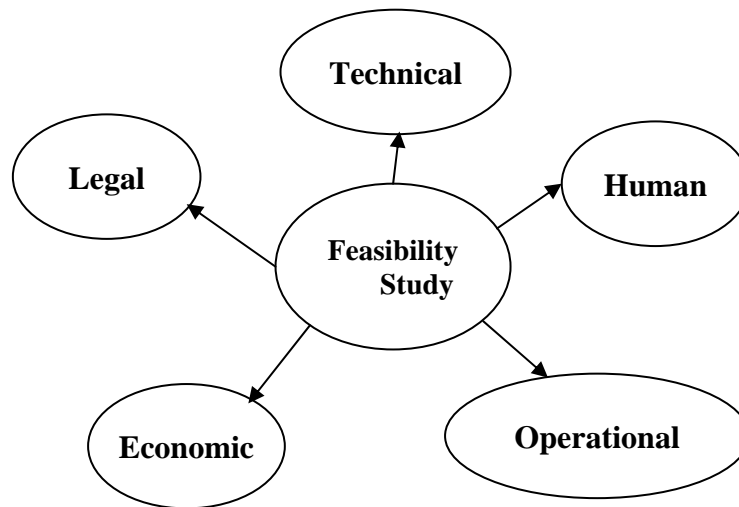


Figure 8: Terms of description in feasibility study

Feasibility study as the first stage of the SDLC is one of the decision point since it determines the whole system life cycle to proceed or not. At the end of this stage, a recommended solution is proposed with an outline functional specification to management. (Avison *et al.* 2003, 28) If the final decision of the feasibility study approves termination of the whole project, it usually means resources are saved to a great extent although it is also possible to stop the system life cycle in the later stages.

### 3.2.2 System investigation

Once the management approves continuation of the SDLC, the next stage is a detailed fact-finding phase (Avison *et al.* 2003, 28). In this case, the system investigation studies one step further. It attempts to find out “What is the problem, and is it worth solving” (Stair & Reynolds 2009, 497). According to Kerzner (2009, 420), the fact that the system investigation looks for is mainly about “defining the requirements” or in other words requirements specification. The initial work of the requirements specification then refers to recognition of the goal of the system and the problems that exist within the current system (Faulkner 2000, 95).

If a new system is built to replace the current system, functional requirements of the latter one need to be investigated and be determined whether they are achieved or not, and if not, what problems exist. Otherwise, if there is no such existing system, the goal of the new system could be gathered by interviewing personnel (both management and operational staff), through using questionnaires, by observation of the application area of the interest directly, by sampling, by checking records and other written materials related to the application area (Avison *et al.* 2003, 28-29).

To gain the information effectively, it is necessary to have a lot of skills involved in and use some of the above mentioned approaches. Sometimes, a couple of similar systems implemented elsewhere which may share invaluable experiences are possible to be found out (Avison *et al.* 2003, 29). Finally, the information collected during this stage will be useful for the next stage of the SDLC (Curtis & Cobham 2005, 407).

### 3.2.3 Systems analysis

Systems analysis stage is the phase that establishes the need for and the extent of an information system (Rob & Coronel 2007, 373). This stage attempts to answer the question “What must the information system do to solve the problem”, it involves studying existing systems and work practices to identify strengths, weaknesses and opportunities for improvement (Stair *et al.* 2009, 497).

According to Curtis *et al.* (2005, 407), the systems analysis stage also includes the creation of a logical system design which will be based partly on information collected during the systems investigation stage and partly on new information gathered from the current system.

For the sake of creating a logical design, some process techniques might be used such as data flow diagrams (DFDs), hierarchical input process output (HIPO) diagrams, and entity-relationship diagrams (ERDs) (Rob *et al.* 2007, 376). From the logical design, no decisions on the way the system will be physically realized should be apparent (Curtis *et al.* 2005, 408). However, it must specify the appropriate conceptual data model, inputs, processes and expected output requirements (Rob *et al.* 2007, 376).



In a case that a database is needed for the new application, data-modelling activities or data analysis (Curtis *et al.* 2005, 408) may also take place in this stage to clarify all entities, together with their attributes, and the relationships among the entities within the database (Rob *et al.* 2007, 376). Figure 9 shows an example of entity modelling:



Figure 9: Relationship of employee and department (Ponniah 2007, 188)

At the end of the systems analysis stage, the output will be a logical design model as revealed by data flow diagrams, together with entity-relationship diagrams and the data model, etc. After reviewing the output, the management and users need to establish an agreement on whether the analysis is approved or not. (Curtis *et al.* 2005, 408)

#### 3.2.4 Systems design

Once the output of the systems analysis states that a new system will be worth having, designing the new system can be started (Doyle 2001, 100). Systems design seeks to answer the question “How will the information system do what it must do to obtain the problem solution” (Stair *et al.* 2009, 497). The purpose of this stage is to design a system, a site or an application that fulfils the requirements determined in the analysis stage (Tol 2009).

The primary output of systems design is a physical design model that either describes the new system or describes how the existing systems will be modified (Stair *et al.* 2009, 497). According to Curtis *et al.* (2005, 408), there are a number of ways to incorporate the logical designed model that is generated in the previous stage into a physical design model. For example, to implement the data stores as a series of files, to incorporate some of the data flow diagram processes into a computer system and remain the rest manual, and of those to be computerized, which are to be processed online and which are to be run only offline.

During the procedure, further investigation should be undertaken to consider what inputs, processes and outputs will be needed (Doyle 2001, 100). Moreover, Stair *et al.* (2009, 498) even point out that systems design not only details inputs, processes and outputs, but also details user interfaces, specifies hardware, software, database, telecommunications, personnel, procedure components and shows how these components are related. Likewise, Avison *et al.* (2003, 30) list the details that the systems design documentation should contain:

- What is the input data and how is the data inputted into the system
- What is the output data
- What are the processes that converting the input data to output data
- Structure of the computer and manual files
- Security issues and back-up provisions
- Plan for system implementation and testing

By the end of this stage, system analysts will present two or three design alternatives to the management. Again the management is going to see and determine whether the logical design model can be physically implemented or not. (Curtis *et al.* 2005, 409)

### 3.2.5 Implementation

The implementation of the new system is led by various procedures followed by the previous stage (Avison *et al.* 2003, 30). It involves creating or acquiring the various system components detailed in the systems design, assembling them, and placing the new or modified system into operation (Stair *et al.* 2009, 498).

Based on the physical design model, if it includes computer programs, these have to be written and tested individually, and if it requires hardware and software systems, those also need to be purchased and installed (Avison *et al.* 2003, 30). It is important that the system is formally tested and accepted before changeover (Curtis *et al.* 2005, 409). By contrast, failure will cause a lack of confidence and, possibly, future computer applications (Avison *et al.* 2003, 30).

Unlike the previous stages, instead of system analysts, computer programmers are involved in this approach to carry out the design and coding of the program (Avison *et al.* 2003, 30). Particular attention will be paid to security features surrounding the conversion of existing files, whether manual or computer-based, to the new system (Curtis *et al.* 2005, 409). Consequently, only authorized users can access to the system and recovery is possible in case of failure (Avison *et al.* 2003, 30). In addition, the database or file structure is created and historical data from the old system, manual or computer-based, is loaded (Curtis *et al.* 2005, 409).

An important task during implementation is to train the users (Stair *et al.* 2009, 498). Without thorough training, users will be unfamiliar with the new system and unlikely to cope with the new approach, especially with the radical new system (Avison *et al.* 2003, 30).

Once all those mentioned above have been done, the new system can be operated and the old system can be discontinued. There could be problems associated with the new system, direct conversion is usually too risky an approach to changeover (Avison *et al.* 2003, 30). One good possibility is that the period is short enough that at the time the new system starts running, the old system is discarded immediately (Curtis *et al.* 2005, 410).

One alternative is to run the old system in parallel with the new one. As a result, it provides a backup system to rely on if the new system fails although the labor costs increase. (Curtis *et al.* 2005, 410) According to Avison *et al.* (2003, 30), to run a pilot scheme is the other alternative, that is, to run a small version of the system before the full system “to test the water”. Figure 10 illustrates the three mentioned forms of system conversion. After the system becomes fully operational, the next stage of the SDLC will start.

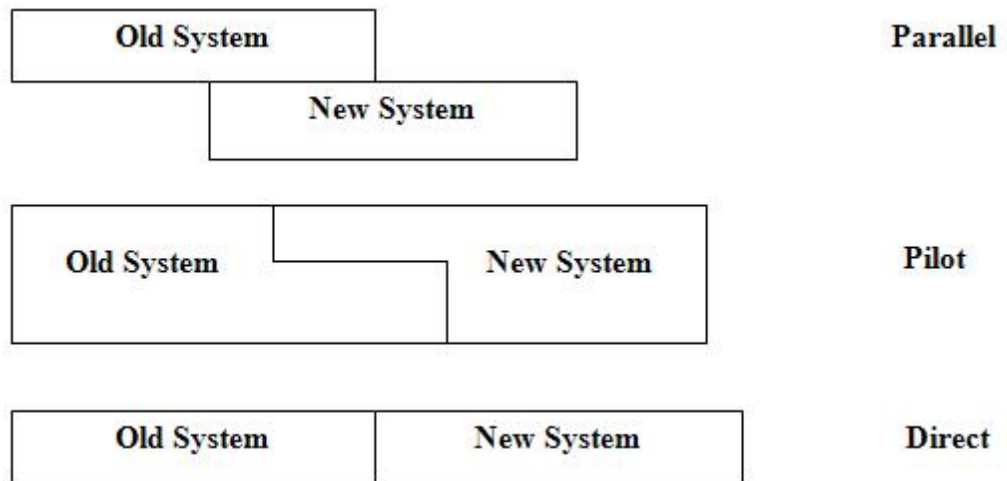


Figure 10: The three mentioned forms of conversion to a new system (O'Brien *et al.* 2008, 404)

### 3.2.6 Review and maintenance

The final stage of the system development process occurs when the system is running and in continuous use. There will be a review of the system to ensure that it does conform to the requirements set out at the feasibility study stage, and the costs have not exceeded those predicted (Avison *et al.* 2003, 31). This process includes a periodic review or audit of a system, the audit occurs in addition to continually monitoring a new system for potential problems or necessary changes (O'Brien *et al.* 2008, 406). It is customary to produce an evaluation report to gain an improvement in the way other systems are developed through the process of organizational learning (Avison *et al.* 2003, 31). Shortcomings will be identified, if these are easily rectified, then changes will be made during normal maintenance (Curtis *et al.* 2005, 410).

In order to ensure that the system operates, to modify the system so that it continues to meet changing business needs (Stair *et al.* 2009, 498), and to ensure the continued efficient running of the system, some changes are usually necessary and some staff will be dispatched to maintenance work. Some changes will be made due to changes in the organization or its environment, some to technological advances, and some to “extras” added to the system at an agreed period following

operational running. Besides, some maintenance work is also related to the correction of errors found during system operational. (Avison *et al.* 2003, 31)

If shortcomings are difficult to be rectified, it may require more drastic surgery (Curtis *et al.* 2005, 410). In some ways, once the maintenance stage begins, the SDLC starts over again (O'Brien *et al.* 2008, 406). According to Avison *et al.* (2003, 31), the operational system can be replaced by a new one when changes in the application area are no longer appropriate. The SDLC then ends and the feasibility of a replacement system is considered with the view to a new life cycle.

### 3.3 The strengths and weaknesses of the SDLC

SDLC has long been used as early as 1960s to methodologically solve problems on computer use and people management, in short, the information technology system. The survival of this method to these days proved it to have some strength. However, traditional SDLC has some weak points too. (Lewis 2008, 63)

#### 3.3.1 The strengths of the SDLC

First of the positive points of traditional SDLC is that it has been well-tested and well-tried (Lewis 2008, 63). Besides, communication is enhanced because the SDLC imposes a consistent set of documentation standards (Davis & Yen 1998, 3). The use of documentation standards helps to ensure that the completed specifications are communicated to the systems development staff, the users in the department, and the computer operations staff. It also ensures that these people are trained to use the system. Moreover, the education of users on subjects such as the general use of computers helps to quell fears about the effects of computers. (Avison *et al.* 2003, 34)

Afterward, the steps in the SDLC enhance management control, providing a framework for scheduling, budgeting, and project management (Davis *et al.* 1998, 3). Therefore unexpected high costs as well as lower expected benefits are avoided and missed changeover dates (deadline dates for the systems to be operational) are prevented to some extent (Avison *et al.* 2003, 34).

Last but not least, the traditional SDLC also allows the progress to be reviewed by the technologists and the users at the conclusion of each stage (Lewis 2008, 63) so that significant errors are more likely to be detected in the early stage of the life cycle (Davis *et al.* 1998, 3).

### 3.3.2 The weaknesses of the SDLC

Though there are many strengths of the traditional SDLC, there are always criticisms such as that the system developed may not accurately reflect the current business environment. A research carried out by the Cutter Consortium found that traditional SDLC methodologies “fall short in the new e-business environment. They are unable to keep up with fast-paced, ever-changing e-business projects”. (Cutter, 2000) The elapsed time between the initial proposal and the completion can be quite lengthy (often one or more years). Especially it often requires specifications be “frozen” as work progresses from one step to the next although user requirements do change over time. (Davis *et al.* 1998, 3)

Meanwhile, the traditional SDLC fails to satisfy the needs of the management, the system design of this method is also very plain and is often un-ambitious (Lewis 2008, 64). According to Avison *et al.* (2003, 35), systems developed by the SDLC approach often succeed in the operational processing level, but always ignore the middle and top management level. What information the management levels need is related to making decisions such as where to locate a new factory, which products to stop selling, what sales or production targets to aim for and how sales can be increased. Despite some information providing summary or exception reports, the majority of the computer is used only for routines and repetitive tasks. Avison *et al.* (2003, 35) continue pointing out that computers are being used to help solve low-level operational tasks rather than meeting corporate objectives, in that case, such systems are limited in their scope and un-ambitious. Figure 11 describes how the SDLC fails to meet the needs of management.

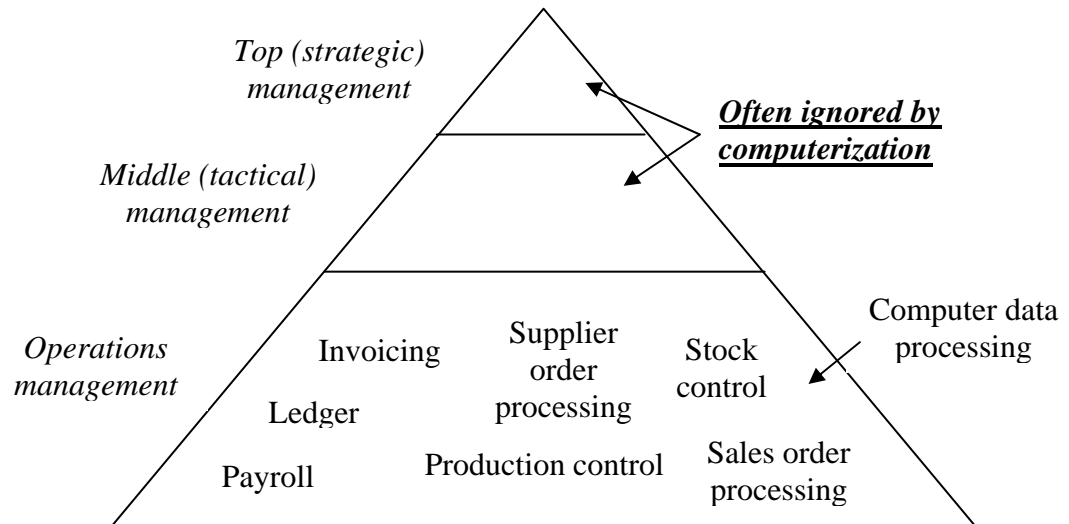


Figure 11: Failure to meet all the needs of management (Avison *et al.* 2003, 35)

Also, the SDLC is concerned as emphasising on “hard” thinking. Some people become so bogged down in the mechanics of following the steps and completing the exit criteria that they fail to solve the real problem (There is a fine line between discipline and rigidity) (Davis *et al.* 1998, 3). The real situations encountered are often ambiguous, issue-laden, messy and problematical.

However, the SDLC approach assumes that there are “facts” which only need to be investigated and identified; it assumes that there can be a “best solution” identified which will “solve the problem”; it assumes that this “ideal solution” can be easily engineered by following a step-by-step methodology; and it assumes that the techniques offered will analyse and design all that needs to be done. (Avison *et al.* 2003, 38)

In addition, other weak points of the traditional SDLC are its being unstable, inflexible, dissatisfaction of users, documentation problems, need of control, application backlog, incomplete systems, problems with some approaches, and maintenance workload. (Lewis 2008, 64)

## 4 A METHOD TO DEVELOP APPLICATIONS ON SOCIAL NETWORKING SITES

### 4.1 Introduction to the application development life cycle

Social networking sites application development life cycle (SNSADLC) is a methodology for guiding developers to build third-party applications for those burning social networking sites, such as the most popular sites Facebook and MySpace. It is generated originally from the traditional SDLC and turned into a special method that only orients to third-party application developers. The aim of putting SNSADLC forward is to provide project development staff with general instructions and rules to work to.

SNSADLC is broken down into six steps which cover feasibility study, end-user investigation, developing specification, application analysis and design, application implementation and application publication, review and maintenance. The outline at each of the six steps of SNSADLC is listed as below:

- Feasibility study
  - Identify the opportunity
  - Evaluate OETHL Feasibility
- End-User investigation
  - Determine the scope of end-user
  - Understand end-user data
- Developing specification
  - Define the structure and content
  - Establish development timeline
- Application analysis and design
  - Draw a DFD and an ER Diagram
  - Prototype the User Interface
- Application implementation
  - Build the application
  - Integrate the application with SNS



- Application publication, review and maintenance
  - Publish the application on a third-party server
  - Improve the application

In this study, a series of simple actions of developing an application for a social networking site would be recorded under each SNSADLC step. The actions note the whole process of building the application “Crazy Photo” on the Facebook platform. The application “Crazy Photo” is completely designed and implemented by the author in order to better describe the proposed SNSADLC; therefore the copyright of “Crazy Photo” is reserved.

## 4.2 Social networking sites application development life cycle

### 4.2.1 Feasibility study

With opening their API (Application Program Interface) to third-party developers, social networking sites have become the world’s largest application platforms with hundreds of millions monthly users. According to a site comparison among the three biggest SNSs (Compete 2009), as of August 2009, there were 122 million visitors surfing Facebook, 56 million using MySpace and 24 million visiting Twitter.

From a business point of view, such a big amount of users stands for uncountable challenges and opportunities. Dramatically, those application developers turn into entrepreneurs on the battle of social networking site platforms. Thus, the first-task for the potential “entrepreneur” is to identify, explore and then select an attractive business opportunity. Kumar (2008, 24) defines opportunity as an attractive and excellent project idea which an entrepreneur search for and accepts such idea as a basis for his investment decision. Back to the developers’ side, an opportunity might imply a good idea for developing a certain application which has a potential user demand and is deserve to invest money or time to achieve it.

The OETHL feasibility study stands for feasibility studies in five different fields with regard to operational, economic, technical, human factors and legal respectively (O’Brien *et al.* 2008, 379). We now describe each of the five

categories in turn. The operational feasibility assessment focuses on the degree to which the proposed application matches with the social environment such as multinational culture and friend network. The purpose of the economic feasibility assessment is to determine the extent to which the proposed application will provide positive economic benefits to the organization involving the identification of all expected benefits and costs of the application. The assessment of technical feasibility focuses on gaining an understanding of the present technical resources of the developers and their applicability to the expected needs of the proposed application. The human factors feasibility assessment focuses on the most important components of a successful application implementation: the end users. In this category, we assess how sociable the proposed application will be. The legal feasibility of a proposed application includes a thorough analysis of any potential legal ramifications resulting from the construction and implementation of the new application.

***Action 1: Crazy Photo on Facebook: Feasibility study***

According to Facebook Statistics (2009), there are some major reasons to choose Facebook as the action platform to developing a photo skimming application:

- It currently has the biggest user amount, more than 300 million active users.
- More than 2 billion photos are uploaded to the site each month.
- Every month, more than 70% of Facebook users engage with Platform applications.
- Average user has 130 friends on the site.

The idea is then raised up to develop a photo flicking application due to the fact that billions of photos are viewed by millions of people every month, and about two thirds of people are fans of application users on Facebook.

On the other hand, the existing photo application on Facebook is clear enough but seems lack of new modern elements, especially after the appearance of Apple's image reflection which looks like the new drop shadow (see figure 12).

Nowadays, reflections are found in iChat, iTunes, iPhoto and everywhere on Apple's marketing site (Jason 2006).

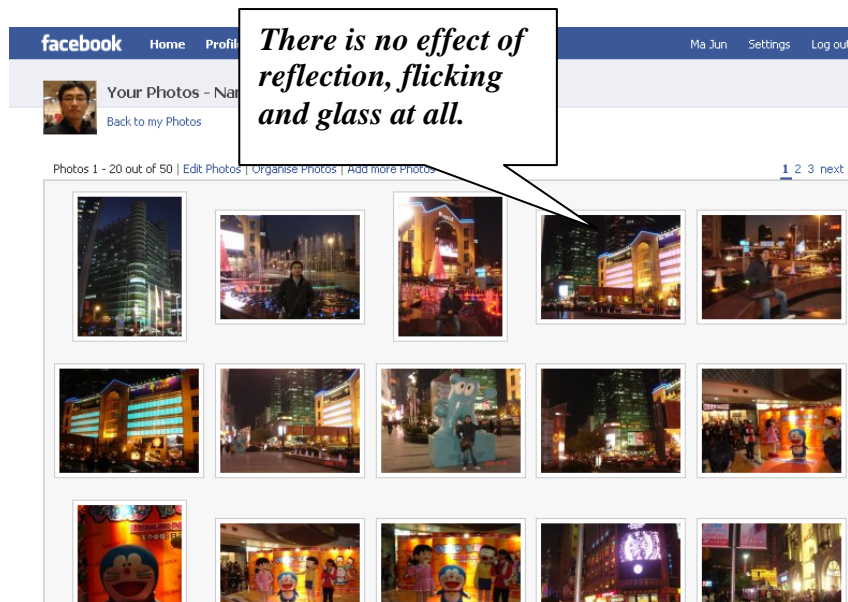


Figure 12: The existing photo application lacks of modern elements

In a word, the opportunity to develop a photo flicking reflection application in order to give users a new way to view their photos is identified. Next, the OETHL feasibility will be measured as shown in figure 13.

<p><b>Operational Feasibility</b></p> <ul style="list-style-type: none"> <li>• Give users an alternative to view their own photos and their friends' photos.</li> <li>• Give users a modern reflection, flicking and glassy experience.</li> </ul>	<p><b>Economic Feasibility</b></p> <ul style="list-style-type: none"> <li>• Use the social networks photo resources, save own server space costs, even for free.</li> </ul>
<p><b>Technical Feasibility</b></p> <ul style="list-style-type: none"> <li>• The third party server (www.hostoi.com) is free for limited usage and reliable:</li> </ul>	<p><b>Human Factors Feasibility</b></p> <ul style="list-style-type: none"> <li>• Photo viewing is a key activity for social networks users.</li> <li>• Young people like new elements such as reflection and flicking.</li> <li>• The developer is capable of web 2.0 techniques.</li> </ul>
<p><b>Legal Feasibility</b></p>	
<ul style="list-style-type: none"> <li>• No patent or copyright violations.</li> <li>• No privacy policy violations.</li> </ul>	

Figure 13: The OETHL feasibility study of the application in action

#### 4.2.2 End-User investigation

When we mention end-user investigation, it carries two layers of meaning. With the first sense, we mean that there should be a firm determination of who the users of a product are, what level of expertise they have, what they are likely to assume about the systems and the environment in which they are operating (Faulkner 2000, 22). It is pretty tricky to investigate all users' age, nationalities and levels of expertise, etc, for there are millions of users activate on social networking sites. However, there is one feature of most SNS users that they use SNSs to display their identity and social network publicly and make new connections (Farkas 2007, 109). In addition, considering the huge amount of users, it is very possible they use social applications at both homes and work places. But that is not a matter which place they choose. The matter is that leisure-related applications are favoured by people. Dr Brent Coker, from the Department of Management and Marketing, the University of Melbourne, says that people who do surf the Internet for fun at work are more productive than those who do not (The University of Melbourne News 2009).

Another sense of end-user investigation is to understand user data. Almost all social networking sites store rich profiles in order to allow them to share information with friends. Those rich profiles usually contain such as name, gender, nationality, education background and picture, etc. According to Facebook Statistics (2009), 97% of users have full name, 85% of users have uploaded a picture and 58% of users have specified education history. By understanding these data, applications can be created with a more personalized and social experience. For example:

- Localize the application or content to the user's geography.
- Feature content shared or created by the user's friends.
- Identify other users in the same geography or industry and feature content or trends of those users.
- Create tailored or different experiences based on genders.
- Suggest a user to invite his friends to the application based on the friends' interests.
- Feature key information from a user's stream that is related to the application (such as featuring mentions of popular movies on a movies site).

### ***Action 2: Crazy Photo on Facebook: End-User investigation***

In accordance with Facebook Statistics (2009), the largest demographic concentration of Facebook is the college crowd of 18 – 24 year olds (40.8%) who are thought as pursuer of fashion, newness and coolness. They are likely to be in their puberty while a person looks the most handsome or beautiful; they are likely to enjoy sightseeing vigorously everywhere. Therefore they are keen on sharing private photos with friends or anyone in their networks. On the other hand, perhaps they are struggling with their college study life or under heavy stress of career as new employees. Thus they need a more casual way to view pictures to release pressure.

To differentiate from the existing photo application (see figure 12), it is necessary to explore how Facebook stores billions of users' pictures. Basically, all the photos in Facebook database have three ID (identification) attributes: subject ID, album ID and photo ID.

Subject ID is ID of the user who is tagged (similar to keywords or labels added to a photo to make it easier to be traced later) by other users in the photos. To be tagged by others does not require the user himself to upload any photos, instead, photos that are uploaded by anyone containing the tagged user will have that user's subject ID.

On the other, album ID is ID of the album where the photos belong to. Knowing album ID helps find information such as the album owner, name, cover photo and the amount of photos inside, etc.

Moreover, photo ID is ID of each photo in the photo list. It is the key attribute in the photo application since it is related to where the photo is located in, who owns it, when it is uploaded, and what comment it has got, etc.

Broadly speaking, to understand the three attributes will result in ability to gather a wide range of related data, which is essential to develop applications. On this basis, we can easily using SQL to retrieve photos in a variety of ways from Facebook that finally benefits the end users.

### 4.2.3 Developing specification

In this stage of the SNSADLC, the work is to pre-design the look and feel of the application and to draft a timetable for developing.

To begin with, imagine when we go to a bookstore and open a book about cooking, we aim to find recipes and information on how to prepare them. Likewise, when the user tries to use the application on social networking sites, they expect to find content that relates to the theme of the social application.

The types of application on SNS are almost all dynamic driven, which is decided particularly by its platform of social communication. To better communicate with the users, therefore, there are a couple of issues needed to be taken into consideration when defining the application content:

- What information is available and how to get more from the users
- How can users get information from the application
- What users might expect from the application that other applications have
- What kind of layout should the application have
- Does it require displaying by multimedia

To clarify all these questions is not easy, just as Rognerud (2008, 49) says, building web content and structure correctly should be of primary concern (see Figure 14).

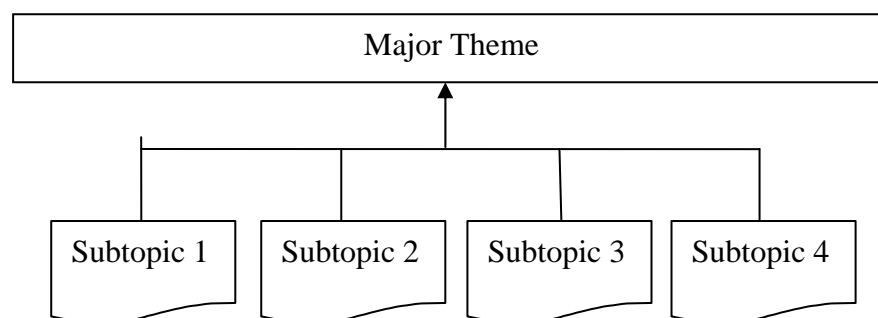


Figure 14: The Silo Approach to Web applications (Rognerud 2008, 50)

The silo approach is thought as the easiest way to successfully manage the content and structure. It forms a logical structure of the web application, and creates related and supporting pages of the application. What is noteworthy is that, in this approach, related information is grouped into distinct sections within the application.

Moreover, Rognerud (2008, 50) indicates that content of the web related application should be organized for relevance, eye appeal and navigation. The navigational bar must be placed at the top of the application and keep the look consistent on each page.

Once the rough content and structure of the application have been defined, it is necessary to establish a development timeline as well. A fixed date for analysis, design, implementation and maintenance activities might drive the schedule for going live (make the application available on the SNS). Zimmerman (2008, 58) argues that without a target date, an application will never be finished. A realistic timeline allocates a quarter of the time to analysis and design, half of the development time to actual programming and fill the application with content, and another quarter time to testing and revisions.

For applications on social networking sites, the development time is unthinkable to be two years, unacceptable to be six months either (Avison *et al.* 2003, 104), approximately three months is a realistic minimum for most cases (Zimmerman 2008, 58). One real example is Playfish who spends three-six months in developing high quality social applications on SNS (Playfish Case Study 2009).

### ***Action 3: Crazy Photo on Facebook: Developing specification***

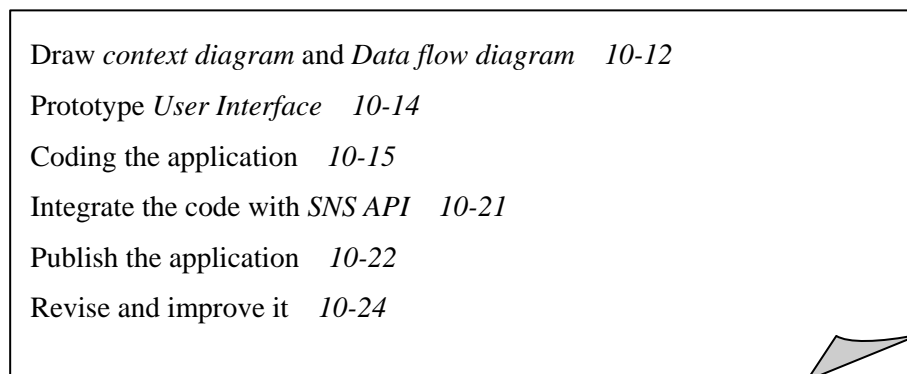
As we discussed in the previous actions, we are developing a photo flicking reflection application, therefore our core content on the main page is to be a photo viewer, no matter whose or what photos the users view, he does not change, what change are just the photos with reflection and flicking effects. A cool horizontal dragging bar inside the viewer is indispensable to allow users to move photos easily and quickly. Meanwhile inside the viewer, the default photos are set to be those photos the users were tagged.

Besides the photo viewer which is a core element, the users ought to have rights to search friends' by typing their full or just parts of their names. In this case, at least a textbox and a button are needed for getting input data from users. Thus normally, the page and the photo viewer would be refreshed to any friends the users want.

To those people who have multiple albums on Facebook, we also need to leave space for listing them out. This is what the existing photo application has and the users are likely to expect from our application. The listing items are all available from the album ID as we mentioned in the previous action including album's title, cover and amount of photos. In addition, to some enthusiasts who own many albums (e.g. over 20), the author believes it a good design to hide most of them by default which are expandable if needed. This idea is used to replace the existing photo application's paging technique.

Moreover, in order to strengthen the utility of social networking resources, the author encourages users to introduce the applications to many of their friends. Consequently a subpage is needed for inviting friends. It is the users' turn to decide whether to invite friends or not since it is not the main expectation the user might have when using the application.

Concerning the size of the actions, it is undoubtedly not realistic to take three months long to develop as usual, we then decide to use two weeks long to complete the remaining work of the actions of which three days for analysis and design work, one week for implementation and the rest for publishing and maintenance. The real timetable is shown in figure 15:



Draw <i>context diagram</i> and <i>Data flow diagram</i>	10-12
Prototype <i>User Interface</i>	10-14
Coding the application	10-15
Integrate the code with <i>SNS API</i>	10-21
Publish the application	10-22
Revise and improve it	10-24

Figure 15: The timetable of developing the application



#### 4.2.4 Application analysis and design

In the process of analyzing and designing a computer-based system or application, data flow diagrams (DFDs) and entity-relationship diagrams (ERDs) are frequently used as important integrated techniques of the system development methodologies (Avison *et al.* 2003, 205). We first look at the DFD method. It conveys how information data flows through systems and how those data are transformed in the process. There are three principal reasons for using DFDs according to Le Vie, Jr. (2000).

- DFDs are easier to be understood by technical and nontechnical audiences.
- DFDs can provide a high level system overview, complete with boundaries and connections to other system.
- DFDs can provide a detailed representation of system components.

DFDs consist of four basic components that illustrate how data flows in a system: entity, process, data store and data flow. The graphical nature of the DFD generates a more concise document that helps users to understand it since it is argued that picture can convey meaning more quickly than traditional textual methods (Avison *et al.* 2003, 205).

Now we turn to the entity-relationship diagram technique. An ERD shows what data are being used in the process or program, and how the files are related (Le Vie, Jr. 2000). It assumes the real world is represented by a set of entities and relationships among them (Avison *et al.* 2003, 178).

Entity-relationship diagram is primarily intended for the database design process, it does not show any program functions, nor data flow (Le Vie, Jr. 2000). An ERD is usually comprised of entities, attributes associated with entities and relationships among entities which are likely to be one-to-one, one-to-many or many-to-many. Figure 9 is an example of an ERD represents the relationship of one-to-many.

User interface design is frequently a prototyping process which will most likely help the application design work. User interface design process concentrates on the design of attractive and efficient forms of user input and output. It produces detailed design specifications for information products such as display screens,

interactive user/computer dialogues (including the sequence or flow of dialogue), audio responses, forms, documents, and reports. (O'Brien *et al.* 2008, 386)

Nevertheless, Faulkner (2000, 106) argues that the prototype of the user interface should be seen merely as a means of getting the design right. The most essential is the user interface needs to be designed from a programming point of view and must be properly represented at all stages, otherwise it will become impossible to maintain.

#### ***Action 4: Crazy Photo on Facebook: Analysis and Design***

At this point, we will use and only use data flow diagram technique to help analyze the application. Entity-relationship diagram is not necessary since the small application in action does not require a database design. We are clear that the “crazy photo” application displays photos including tagged photos and album photos based on user’s selections in a modern way.

These activities are all within the boundary of the application. The data flow diagrams are drawn in the level 0 (the top level, also known as context diagram, see figure 16) and the level 1 (see figure 17).

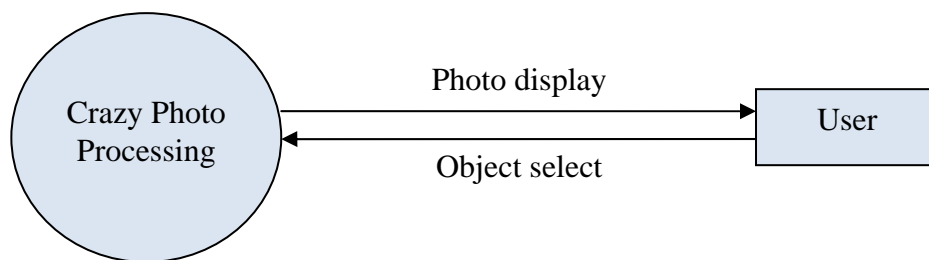


Figure 16: Level 0 DFD of “crazy photo”, also known as context diagram

In the level 0 DFD, the circle is the process box containing all the activities happening within the application boundary; the rectangular is the entity while the arrows are data flow connecting both process and entity.

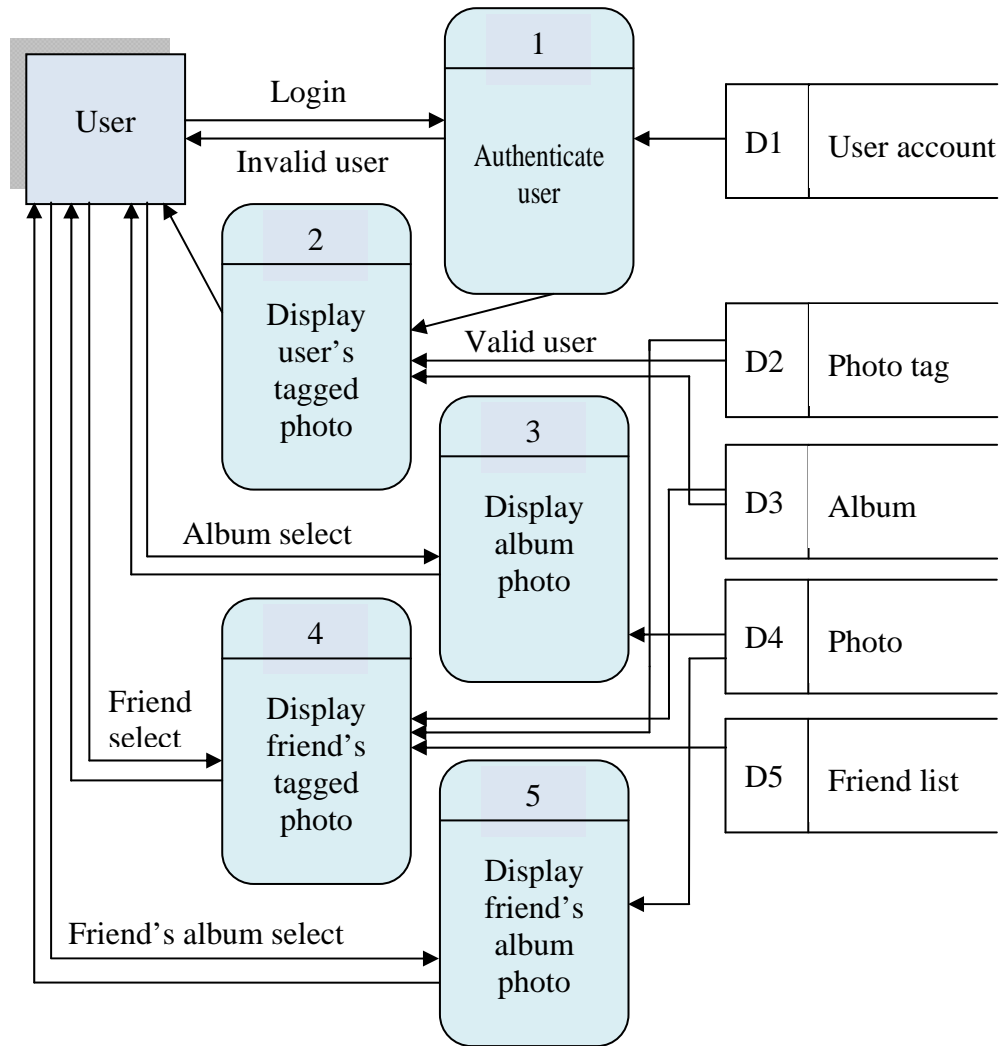


Figure 17: Level 1 DFD of “crazy photo”

The Level 1 DFD provides a high-level view of the application that identifies the major processes and data stores. Figure 17 illustrates the process and data store by using a rectangular with a rounded corner and a pair of parallels with one end closed.

Based on the analysis and design work of data flow diagramming, together with the previous action steps, the user interface of the “crazy photo” can be drawn by simple illustration tools of word processing. Figure 18 depicts the attractive user interface of the application comprised of user input and output.

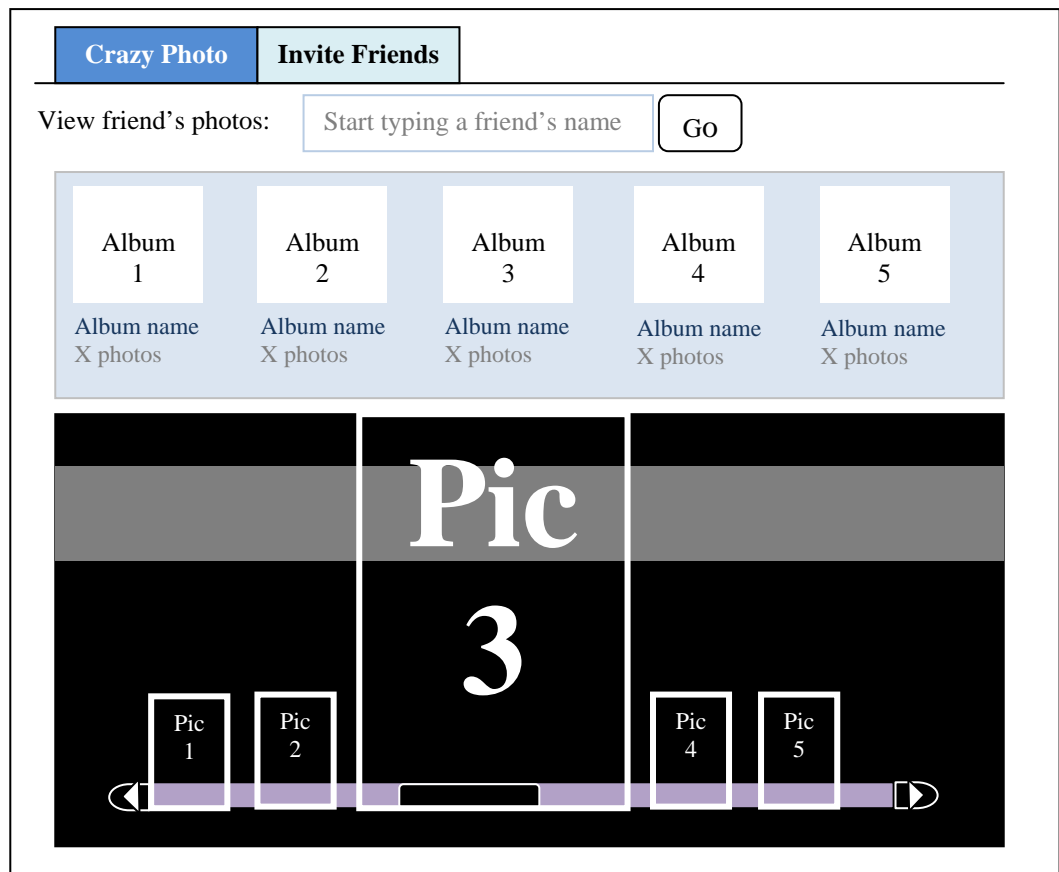


Figure 18: “crazy photo” user interface prototype

The black background is the main photo displaying area with a horizontal navigation bar. The photos will be displayed in a flickering and water-reflected way. Above the photo is the pretty light blue area where lists all the users' albums with their names and number of photos.

If the user wishes to view any friend's photo, he can input the name of the friend into the textbox which is a bit upper than the album area and click the “go” button to trigger the page to redirect. The two tabs on the top can link the user to either the home page “crazy photo” or the invitation page if he wishes to tell more friends about this application.

#### 4.2.5 Application implementation

The application implementation phase aims to transform the logical and physical design into completed application. Almost all the coding work before launching

the application is done in the phase. Therefore programming language skills are required. HTML (Hypertext Markup Language), XML (eXtensible Markup Language), and Java are three programming languages which are important tools for building multimedia Web pages, Web sites, and Web-based applications (O'Brien et al. 2008, 141).

The coding work can be done much more easily and smoothly if two parts of the previous study were carried out properly.

- Human Factors Feasibility (O'Brien *et al.* 2008, 381). The part of feasibility study in the very early development stage ensures the developers are capable of the potential required techniques. If the developers are predicted as lacking of such techniques or having techniques in another possible area, the applications development might go towards that area so that a huge amount of resources are avoided of being wasted.
- User interface designing is based on programming point of view (Faulkner 2000, 106). The designing work is not radical interface drawing. It should adequately cooperate with coding work. If user interface designing and application programming are separated into completely isolated sectors, there is rarely any possibility to succeed in the development.

Once the application is coded and technically working, it is still a semi-finished product: it is far away from social networking sites since actually they are not connected. Therefore the rest of the work in this phase is to introduce API of SNSs into the application. SNSs usually would send application developers API keys. Then the integration jobs can be achieved by leaving the API keys into both application administration pages on SNS and the applications' codes.

According to Facebook Developers (2009), to introduce API of SNSs into the application has three benefits besides its connection function.

- It enables the developers to extract information very easily about each individual authenticated and authorized through the applications, and their friends.

- It enables the developers to encourage their users to invite other friends of the SNS to continue using the applications.
- It enables application users to share statuses, photos, videos, and even entire streams of data with their friends. Especially, the shared contents are often provided by the applications that individuals are able to share with their social graph.

### ***Action 5: Crazy Photo on Facebook: Implementation***

The coding work has started from 15 October 2009 based on the drawn schedule. There are several files of Facebook client library which need to be downloaded into the code. To begin with, we choose FBML (Facebook Markup Language) as the frame of our canvas page. FBML is the markup language developed by Facebook and it is an evolved subset of HTML with some elements removed while others which have been added are specific to Facebook. It enables building full Facebook Platform applications that deeply integrate into a user's Facebook experience (Facebook Developers 2009).

Meanwhile, we also use PHP (a widely – used scripting language that is especially suited for Web development and can be embedded into HTML), JavaScript and CSS (Cascading Style Sheet) to be associated with FBML in order to better develop the application. By using basic HTML and CSS elements, we manage to build the application interface based on the prototype designed (see source code in figure 24). The logical part is not complicated as in different use cases, we simply accordingly retrieve different data from different data stores which were analyzed in previous step (see source code in figure 25).

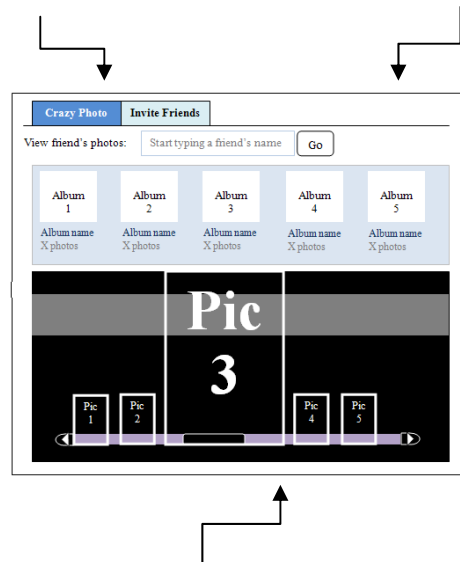
The functional part is the most challenging therefore we spend most of the time on figuring it out. The potential technical barriers such as dragging bar, water-reflection and flicking are easily leaped over, while some adjustment work in order to fit for Facebook platform costs much time, for instance, interface width and length, flicking speed, photo size and reflection size.

In the last stage of implementation, we need to connect the application with Facebook by integrating the Facebook API into the application. We install the Facebook Developer application and enter all the required information to get the

API key so that the application can communicate with the API (see source code in figure 26). The whole process of connecting between Facebook API and the application is illustrated in figure 19.

1. Set up the application,  
install the client library

3. Write callback script to  
execute the application



2. Copy the API key and root into the application

Figure 19: The three steps to connect Facebook and the application

#### 4.2.6 Application publication, review and maintenance

The last step of SNSADLC is to finalize the applications so that they are ready to be used by the users. Different from the traditional SDLC, the publication is specially added in the last stage of the life cycle in order to highlight its importance and speciality. Although the applications are developed for the SNS platform, it does not offer any bidding servers to run the application. Therefore the developers in some case need to look for reliable third-party servers to host the implemented applications. There are a large number of good servers. However there are certain criteria for a good server of SNS applications. The considerable factors of choosing a server are the reliability, cost and the most important, whether it is compatible with the SNSs or not.

Once being put on the good server, the application needs a thorough testing before it is formally published to the end users. Scenarios are a good way for testing

since they enable users to look at the features and functions offered by a system so that it can reveal problem areas and errors (Faulkner 2000, 169).

After the application is published for some time, a review of the application and maintenance is necessary to be conducted to ensure the application does conform to the expectations laid out in the feasibility study and the continued efficient running of the application (Avison *et al.* 2003, 31). During maintenance, a series of methods can be used for gathering information from users including interview, questionnaire, observation, survey, and discussion with them (Faulkner 2000, 30). These processes might lead to an improvement of the application which advance its technology, decrease its errors and increase its sociality.

### ***Action 6: Crazy Photo on Facebook: Publication, Review and Maintenance***

Once the application is implemented, it is almost ready for use. At the same time, all the code files are still lying in the development console. Therefore, we need to find a third-party server to host them. The first server “Freehostia” ([www.freehostia.com](http://www.freehostia.com)) offers 250 MB free disk space and performs properly for normal web pages. However, when the source code files are moved onto it, they just do not work. The problem occurs might due to “Freehostia” does not support Facebook API. We then have to start looking for a new server.

The second web server we find is “Webhost” ([www.000webhost.com](http://www.000webhost.com)). Compared with “Freehostia”, it also offers free service and it is more reliable and feature-rich. 1500 MB disk space and 100 GB bandwidth make it pretty ideal for free web hosting. Shortly afterwards, we move all the application source code files onto “Webhost”, excitingly, “crazy photo” succeeded in performing for the first time on Facebook (see figure 20)!

Although “crazy photo” is successfully moved onto the third-party server, it needs some scenario testing before performing publicly. The scenarios then are designed according to different use cases:

#### ***Scenario 1 for the “crazy photo”***

Jack is a netizen who has many photos uploaded on 24 albums of Facebook, he installs “crazy photo” and watches all of his tagged photos on the home page. He



also clicks the “expand” link to view all of his 24 albums. He then chooses any albums among them. After that, he feels the application quite interesting and decides to share it with his friends. He clicks “invite friends” link, thereby he finally invites three friends.

### *Scenario 2 for the “crazy photo”*

Peter is one of the three people who receive invitation of using “crazy photo”. He is a newbie of Facebook therefore he has no photos there. He installs “crazy photo” but there has no tagged photos and displayed albums. He sees “view friends’ photos” tag and types his friend’s name “Jack” in. After clicking “Go” button, he finds Jack’s tagged photos and five of Jack’s albums above them. Furthermore, all of the 24 albums are shown after he clicks “Expand” link. He chooses one of the albums and starts watching all the photos in the chosen album.



Figure 20: “crazy photo” on Facebook

Based on the scenarios, the application is thoroughly tested. Finally the small fatal problems are solved and system functions are improved until it is completely ready to use and formally promoted to end-users.

The success exceeded all expectations within only two days after publishing. As shown in figure 21, there are altogether 11 users who used “crazy photo”. Among them, most people think it is worth using. Based on a simple survey conducted online (<http://crazyphotosatisfaction.surveyconsole.com>), almost all of the users feel comfortable and easy when using “crazy photo”. Through the survey, user H suggests the photo flicking speed could be slowed down a little bit in order to be less dizzy. Meanwhile, user Q comments “crazy photo” as a cool application since it animates the photos without using Flash which requires heavy loads for computers. Moreover, user N hopes that the definitions of the photos can be as good as the original photos.



Figure 21: “crazy photo” application info

At last, in the end of the development life cycle of “crazy photo”, what is going to be is a long-term, iterated maintenance according to various reviews from all end-users. There is possibility that one day, the maintenance and changing work is too heavy to be controlled and the “crazy photo” is no longer favoured by users. Then the SNSADLC of “crazy photo” is ended. At that time there might be a need to develop another application and it will enter a new SNSADLC as matter of course.

## 5 CONCLUSION

### 5.1 Evaluation of the SNSADLC methodology

To evaluate methodologies are always thought as a difficult task since different writers of methodologies might have various criticism views. According to Avison *et al.* (2003, 555), this chapter introduces a simple set of views to evaluate the SNSADLC methodology (see figure 22). The evaluation is represented by a subjective view rather than a statement of “facts” about methodologies.

1. Philosophy
  - a) Paradigm
  - b) Objectives
  - c) Domain
  - d) Target
2. Model
3. Techniques and tools
4. Scope
5. Outputs
6. Practice
  - a) Background
  - b) User base
  - c) Participants
7. Product

Figure 22: The seven elements of evaluating methodologies (Avison *et al.* 2003, 556)

#### 5.1.1 Philosophy

The first element concerns the identification of the philosophy of the methodology. In the following paragraphs, we will examine the four sub-elements to philosophy in turn.

*a) Paradigm*

As to SNSADLC, it is the belief in the integration of the social networking sites and traditional system development life cycle. The work system is proposed based on the analysis of both parties in terms of their characteristics and weaknesses. Therefore it inherits the traditional SDLC model and considers the features of SNSs as important elements. As a result, the underlying paradigm for SNSADLC is the systems paradigm.

*b) Objectives*

It is clear that SNSADLC aims to develop computerized information systems. It is different to those general problem-solving methodologies. To some extent, building computerized information systems belongs to wide-ranged problem-solving methodologies. The bounded action which develops an application on SNSs illustrates that the main focus of SNSADLC clearly embodies an assumption that a computerized system is to be constructed.

*c) Domain*

As classified as a specific problem-solving methodology, the domain of SNSADLC is to seek what aspects the methodology addresses. It begins by assuming that a methodology is needed for developing applications on social networking sites. The main contents are then expatiated concerned with the assumption.

*d) Target*

The SNSADLC methodology aims at building applications in a particular environment, that is, the social networking sites. Although it is claimed to be general purpose, the bound action to develop an application is taken in a limited type and only on Facebook among numerous SNSs due to the study time and resources. In addition, the main technique suggested in the methodology is data flow diagramming, which is thought as not particularly suitable for web-base systems (Avison *et al.* 2003, 566).

### 5.1.2 Model

The second element of the evaluation framework relates to the model that SNSADLC methodology uses. The primary process model used is the data flow diagram. In the action of developing, the DFD is subdivided into two levels of which the higher level indicates the main contexts and the lower level describes the data flows within the system boundaries in detail.

Another process model suggested in SNSADLC is entity-relationship diagram. It plays a less significant role in the methodology compared with DFD. Because ERD is mainly used to clarify the structure of databases, it is not useful when there is no need for designing databases. For this reason, it is not drawn in the design of the testing application.

### 5.1.3 Techniques and tools

The third element of the evaluation framework is what techniques and tools that the SNSADLC methodology employs. The methodology assumes all the relevant techniques are capable, which are regarded as fundamental to it such as timetable and scenarios. This makes the methodology does not rely on particular techniques to a great extent and therefore they can be easily replaced or substituted once the more advanced techniques are available.

On the other hand, some tools are recommended to some degree in the SNSADLC methodology. These tools range from prototyping, code generation, server choosing, and so on. Most tools are not implied as specific brand-name tools. There is an exception of an appropriate third-party server “Webhost” for developing applications on Facebook is named, which is thought as practical and particular.

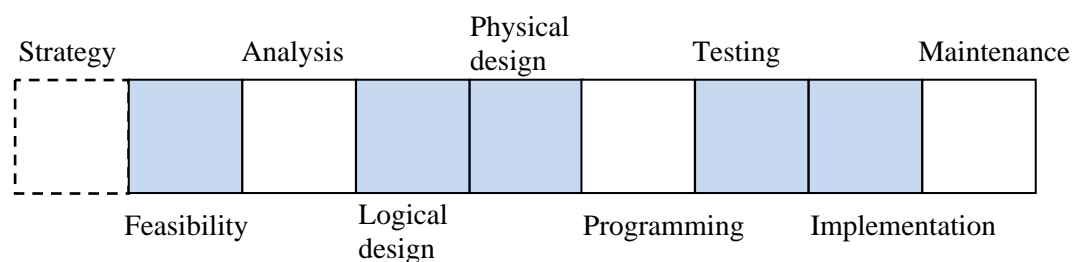


Figure 23: Scope of SNSADLC methodology

#### 5.1.4 Scope

The fourth element of the evaluation framework concerns scope. We focus on life cycle stages that SNSADLC methodology addresses. According to Avison *et al.* (2003, 568), 9 of the most common stages that appear in other methodologies are used to analyze its scope including strategy, feasibility, analysis, logical design, physical design, programming, testing, implementation, and maintenance.

As can be seen in figure 23, different rectangles indicate the results of the analysis of scope. Shaded rectangles represent that the methodology covers the stage in some details including the provision of specific techniques and supporting tools. Whereas, unshaded rectangles illustrate that the areas are only addressed in the methodology without enough contents in detail and depth. Those areas are offered less guidance thus developers can interpret and perform in their own ways. Besides, the rectangle with broken lines addressing overall information systems strategy, purpose and planning is not properly mentioned in the methodology.

#### 5.1.5 Outputs

The fifth element of the evaluation framework is outputs. The outputs stand for what SNSADLC methodology delivers at the end of each life cycle stage. Avison *et al.* (2003, 570) states that the outputs specify how detailed the rules are processed rather than what rules are produced.

In each stage of the SNSADLC methodology, two main issues are discussed. Some deliverables are talked in detail although there are not many of them. For example, OETHL feasibility and end-user scope determination are specified in terms of the outputs as well as user interface prototype. Consequently, the developing actions in those areas are processed more successfully since such identifications of the areas contribute a lot to understanding the methodology.

#### 5.1.6 Practice

The one coming before the last element of the evaluation framework is the practise of the SNSADLC methodology. The practise refers to three facets of the

methodology containing background, user base, and participants (Avison *et al.* 2003, 562).

First of all, the SNSADLC methodology has an academic background. The purpose of the research is to find a better way to guide developing applications on SNSs. It is obviously not sophisticated and inadequate at this moment therefore it is not commercial-oriented.

Next, the user base of the SNSADLC methodology is broad. Actually anyone among millions of SNS users can become a developer of SNS applications. Therefore these developers group a large potential user base of the methodology.

In the end, as there are a huge amount of potential participants of the methodology, it is not necessary for them to command such as professional analysis skills. They range from professional software developers, experts to even the students or fans of this area. The success of the experimental application “crazy photo” proves that under the step-by-step guidance, it is not difficult to develop at least small-sized applications on SNSs with some IT knowledge including designing and programming.

#### 5.1.7 Product

The final element of the evaluation framework is product. Unlike the commercial aimed methodology, the SNSADLC is an academic research. Therefore it does not require any cost for purchasing it. Based on the thesis, the actual length of the methodology document is 19 pages with one real and succeeded example by using it as a guideline.

Generally speaking, through reading the whole procedure of the methodology, the users or developers might be illuminated so that new idea is likely to be generated for developing particular applications on SNSs. Moreover, to developers, the diagrams such as shown in figure 16 and 17 as well as the code example in the appendix 1 are as first hand information of how to develop applications in the SNS platform environment, which might solve design or programming troubles in some cases.

## 5.2 Further development of the SNSADLC methodology

Strictly speaking, the SNSADLC is not sophisticated at present. As there is no standard methodology for web development (Avison *et al.* 2003, 545), especially for social networking development, we hope SNSADLC methodology can become a standard to fill up the gap. For this reason, there is still a big room for improvement. Under this subsection, we discuss the possible improvements for the SNSADLC methodology to develop one step further.

To begin with, we believe there is a need for more process models to be introduced into the methodology. In SNSADLC methodology, as process models, only data flow diagram and entity relationship diagram exist. Although they are enough for small-scale applications, there is potential shortage for large-scale applications. Therefore, beside the primary process models, more models are needed in this case to better analyze data and processes. The possible process models that we find existing in other methodologies are structure diagram, rich picture and socio-technical model, etc (Avison *et al.* 2003, 566).

Next, some facets in the methodology need to be addressed with more contents in both detail and depth including strategy, analysis, programming and maintenance. In order to make it a standard guideline for developers, these contents should form a unique rule instead of being interpreted in different ways by the developers.

Additionally, based on the recommended tools in the SNSADLC methodology, more tools can be added including brand-name tools, for instance, drawing tool Microsoft Visio in the stage of application system analysis. If the application is also database bounded, good database management tools should be suggested such as Microsoft SQL Server. Also, during implementation stage, some sophisticated programming tools ought to be suggested such as Microsoft Visual Studio and Macromedia Dreamweaver.

In the end, it is likely that the SNSADLC methodology can be utilized for commercial use provided that more process models and tools are available. Since more and more organizations are attempting to invest money on purchasing outsourced packages (Avison *et al.* 2003, 546), more and more applications will be developed externally. This could be seen as a potential opportunity for SNSADLC to become a commercial methodology.



## 6 SUMMARY

The study starts with an introduction of social networking sites (SNSs), the burning topic of the Internet nowadays. Different from the old ways of web interaction, by adopting Stanley Milgram's "six degrees of connection" theory, SNS stresses on displaying the identities and networks publicly and thus SNS users can accordingly make new friends. Under the introduction is a history review of the SNSs which can be traced back to as early as early-to-mid 1960s to late 1970s. The SNSs experienced different bellwethers such as Friendster, MySpace to the new giant Facebook which owns millions of active users. The dramatic increasing popularity of SNSs is due to some evolutionary features. Thereby, the appearance of social applications increased versatility of SNS.

The applications on SNSs differentiate from the traditional applications on the Internet which may require local friends or new friends online to play together. The users of social applications can easily share their identities and experiences within their networks. Meanwhile, social applications have pros and cons at the same time such as both good news and bad news may spread through the whole networks within seconds. Nevertheless, sharing is a key feature of the social application and in order to strengthen this feature, more and more SNSs have opened up their API to common users to encourage their own applications development.

The study then continues to review theoretical literatures of the SDLC in Chapter three. The SDLC plays a vital role in information systems development as a general approach. One of the typical features of the SDLC is that no matter what variant it is, it always starts from feasibility study to the review stage.

The traditional SDLC is comprised of six stages containing feasibility study, system investigation, systems analysis, systems design, implementation, review and maintenance. All of these six stages are deeply discussed in the thesis and as a result, strengths and weaknesses of the SDLC are concluded in order to better understand it. For example, providing a framework for scheduling, budgeting, and project management to enhance management control is thought as the good point of the SDLC. On the other hand, unsuitability for the fast-changing Internet environment is one of the drawbacks of the SDLC.

In chapter Four, a methodology of developing applications on SNSs is proposed based on the previous study and analysis. The methodology is comprised of six stages of social networking sites application development life cycle which is specialized in guiding developing social applications. The six stages include feasibility study, end-user investigation, developing specification, application analysis and design, application implementation, application publication, review and maintenance.

Under text description of each stage, a series of simple actions of developing an application “crazy photo” for a SNS Facebook is taken in order to examine the methodology in practical operation.

The conclusion part analyzes and evaluates the methodology, states the limitation of the study and gives the future possible development. The evaluation subjectively discusses the methodology in terms of philosophy, model, techniques and tools, scope, outputs, practice and product. Based on the discussion, the main problems of the study concerning the uncertainty of the user level and the scope of related experiments are pointed out. At the end of the study, further development is suggested relating to the issues of enriching the present process models and tools in the methodology in order to turn the academic methodology towards a commercial one.

## GLOSSARIES

API	Application Program Interface, a set of routines, protocols, and tools for building software applications.
DFD	Indicates Data Flow Diagram, a graphical representation of the "flow" of data through an information system.
End User	The final or ultimate user of a computer system; usually implies an individual with a relatively low level of computer expertise.
ERD	Entity-Relationship Diagram, a specialized graphic that illustrates the interrelationships between entities in a database.
OETHL	Refers to five terms as Operational, Economic, Technical, Human and Legal which are used to evaluate feasibility studies of information systems.
SDLC	System development life cycle.
SNSADLC	The abbreviation of Social networking sites' application development life cycle.
SQL	Stands for Structured Query Language which is a standard language for accessing and manipulating databases.
Third-party Application	The applications that developed and run by the third parties except social networking sites and users.
Web 2.0	Web 2.0 is commonly associated with web applications which facilitate interactive information sharing. A Web 2.0 site allows its users to interact with other users or to change website content. Examples of Web 2.0 include social networking sites, video-sharing sites, web applications, blogs, etc.

## REFERENCES

### **Published books**

- Avison, D. & Fitzgerald, G. 2003. Information Systems Development: methodologies, techniques and tools. 3<sup>rd</sup> International edition, McGraw-Hill, Berkshire
- Baloun, K. 2007. Inside Facebook: life, work and visions of greatness. Trafford publishing, Oxford
- Creswell, J. 2003. Research Design: Qualitative, Quantitative, and Mixed Method Approaches. 2<sup>nd</sup> edition, Sage Publications Ltd, London
- Curtis, G. & Cobham, D. 2005. Business Information Systems: analysis, design, and practice. 5<sup>th</sup> edition, Pearson Education, Essex
- Davis, W. & Yen, D. 1998. The Information System Consultant's Handbook: Systems Analysis and Design. CRC Press, Florida
- Doyle, S. 2001. Information Systems for You. 3<sup>rd</sup> edition, Nelson Thornes, Cheltenham
- Farkas, M. 2007. Social Software in Libraries: Building Collaboration, Communication, and Community Online. Information Today, Inc., New Jersey
- Faulkner, X. 2000. Usability Engineering. Macmillan Press Ltd., New York
- Feiler, J. 2008. How to Do Everything: Facebook Applications. McGraw-Hill, Berkshire
- Flick, U., Kardorff, E. & Steinke, I. 2004. A companion to Qualitative Research. Sage Publications Ltd, London
- Goddard, W. & Melville, S. 2004. Research Methodology: An Introduction. 2<sup>nd</sup> edition, Juta & Co. Ltd, Lansdowne
- Kerzner, H. 2009. Project Management: a Systems Approach to Planning, Scheduling, and Controlling. 10<sup>th</sup> edition, John Wiley and Sons, New Jersey

- Kumar, R. 2005. *Research Methodology: a Step-by-Step Guide for Beginners*. 2<sup>nd</sup> edition, Sage Publications Ltd, London
- Kumar, S. 2008. *Entrepreneurship Development*. New Age International, New Delhi
- Lewis, J. 2008. *SDLC 100 Success Secrets - Software Development Life Cycle (SDLC) 100 Most Asked Questions, SDLC Methodologies, Tools, Process and Business Models*. Lulu.com
- O'Brien, J. & Marakas, G. 2008. *Introduction to Information Systems*. 14<sup>th</sup> edition, McGraw-Hill, Berkshire
- Ponniah, P. 2007. *Data Modeling Fundamentals: a Practical Guide for IT Professionals*. Wiley-Interscience, New Jersey
- Rettberg, J. 2008. *Blogging- Digital Media and Society Series*. Polity Press, Cambridge
- Rob, P. & Coronel, C. 2007. *Database Systems: Design, Implementation, and Management*. 8<sup>th</sup> edition, Cengage Learning, Boston
- Rognerud, J. 2008. *Ultimate Guide to Search Engine Optimization: Drive Traffic, Boost Conversion Rates and Make Lots of Money*. Entrepreneur Press, Madison
- Stair, R. & Reynolds, G. 2008. *Fundamentals of Information Systems*. 5<sup>th</sup> edition, Cengage Learning, Boston
- Stair, R. & Reynolds, G. 2009. *Principles of Information Systems*. 9<sup>th</sup> edition, Cengage Learning, Boston
- Warner, J. 2008. *Web Sites Do-It-Yourself for Dummies*. Wiley Publishing, Inc., Hoboken
- Zimmerman, J. 2008. *Web Marketing for Dummies*. 2<sup>nd</sup> edition, Wiley Publishing, Inc., Hoboken

## Websites

Bernoff, J. 2008. Why Social Applications will Thrive in A Recession.  
<http://blogs.forrester.com/groundswell/2008/02/why-social-appl.html> (accessed 26 September 2009)

Boyd, D. & Ellison, N. 2007. Social Network Sites, Definition, History and Scholarship. <http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html> (accessed 25 September 2009)

Compete. 2009. Site Comparison of facebook.com, myspace.com, twitter.com  
<http://siteanalytics.compete.com/facebook.com+myspace.com+TWITTER.com/?metric=uv> (accessed 5 October 2009)

Cutter. 2000. Light Methodologies Best for E-business Projects. Cutter Consortium <http://cutter.com/consortium/research/2000/crb001003.html> (accessed 19 September 2009)

Davidson, P., Hedrich, R., Leavy, T., Sharp, W. & WA, Y. 2002. Information Systems Development Techniques and Their Application to the Hydrologic Database Derivation Application.  
<http://cadswe.colorado.edu/PDF/RiverWare/DavidsonLV2002.pdf> (accessed 12 October 2009)

Facebook Developers. 2009. Facebook Connect.  
[http://wiki.developers.facebook.com/index.php/Facebook\\_Connect](http://wiki.developers.facebook.com/index.php/Facebook_Connect) (accessed 16 October 2009)

Facebook Statistics. 2009. <http://www.facebook.com/press/info.php?statistics> (accessed 7 October 2009)

Folleth, C. 2002. What is SDLC and what is it used for.  
[http://www.essortment.com/hobbies/sdlcsoftwaresy\\_svkr.htm](http://www.essortment.com/hobbies/sdlcsoftwaresy_svkr.htm) (accessed 5 September 2009)

Jason. 2006. Reflections are the New Drop Shadows.  
[http://37signals.com/svn/archives2/reflections\\_are\\_the\\_new\\_drop\\_shadows.php](http://37signals.com/svn/archives2/reflections_are_the_new_drop_shadows.php) (accessed 8 October 2009)

Kharif, O. 2007. Social-Networking Sites Open Up.

[http://www.businessweek.com/technology/content/feb2007/tc20070213\\_172619.htm](http://www.businessweek.com/technology/content/feb2007/tc20070213_172619.htm) (accessed 28 September 2009)

Le Vie, Jr., D. 2000. Understand Data flow Diagrams.

<http://www.stc.org/confproceed/2000/PDFs/00098.pdf> (accessed 14 October 2009)

Office of Communications. 2008. Social Networking: A quantitative and qualitative research report into attitudes, behaviours and use.

[http://www.ofcom.org.uk/advice/media\\_literacy/medlitpub/medlitpubrss/socialnetworking/report.pdf](http://www.ofcom.org.uk/advice/media_literacy/medlitpub/medlitpubrss/socialnetworking/report.pdf) (accessed 24 September 2009)

O'Reilly Home. 2008. Social Games for Social Platforms: Unleashing Viral Fun.

<http://en.oreilly.com/gspwest2008/public/schedule/detail/2593> (accessed 25 September 2009)

Playfish Case Study. 2009. Facebook developers.

<http://developers.facebook.com/cases/?playfish> (accessed 26 September 2009)

Smithh, A. 2009. Facebook- A successful social networking platform for app developers. [http://www.facebookster.com/blog/facebook--a-successful-social-networking-platform-for-app-developers\\_90](http://www.facebookster.com/blog/facebook--a-successful-social-networking-platform-for-app-developers_90) (accessed 29 September 2009)

Social Application Tutorial (v0.9). 2009. OpenSocial.

[http://wiki.opensocial.org/index.php?title=Social\\_Application\\_Tutorial\\_%28v0.9%29](http://wiki.opensocial.org/index.php?title=Social_Application_Tutorial_%28v0.9%29) (accessed 25 September 2009)

Social Networking Websites Review. 2009. <http://social-networking-websites-review.toptenreviews.com/> (accessed 5 October 2009)

The University of Melbourne News. 2009. Freedom to surf: workers more productive if allowed to use the Internet for leisure.

<http://uninews.unimelb.edu.au/news/5750/> (accessed 9 October 2009)

Tol, M. 2009. Web Site Development Life Cycle.

<http://www.merttol.com/articles/web/web-site-development-life-cycle.html> (accessed 11 September 2009)

U.S. House of Representatives. 1999. Systems Development Life-Cycle Policy.  
<http://www.house.gov/cao-opp/PDFSolicitations/SDLCPOL.pdf> (accessed 8  
September 2009)

Vaishnavi, V. & Kuechler, W. 2004. Design Research in Information Systems.  
<http://ais.affiniscap.com/displaycommon.cfm?an=1&subarticlenbr=279>  
(accessed 28 November 2009)

Wanjiku, E. 2009. Disadvantages of Social Media.  
<http://emmalifetips.blogspot.com/2009/09/disadvantages-of-social-media.html>  
(accessed 26 September 2009)



## LIST OF FIGURES

Figure 1: The relationship between a methodology and an application	2
Figure 2: Research process	5
Figure 3: Thesis structure	7
Figure 4: Identity on and social network on SNS	8
Figure 5: A Facebook profile	11
Figure 6: Data flow among users, SNS and social applications	14
Figure 7: Basic structure of SDLC	18
Figure 8: Terms of description in feasibility study	19
Figure 9: Relationship of employee and department	21
Figure 10: The three mentioned forms of conversion to a new system	24
Figure 11: Failure to meet all the needs of management	27
Figure 12: The existing photo application lacks of modern elements	31
Figure 13: The OETHL feasibility study of the application in action	31
Figure 14: The Silo Approach to Web applications	34
Figure 15: The timetable of developing the application	36
Figure 16: Level 0 DFD of “crazy photo”	38
Figure 17: Level 1 DFD of “crazy photo”	39
Figure 18: “crazy photo” user interface prototype	40
Figure 19: The three steps to connect Facebook and the application	43
Figure 20: “crazy photo” on Facebook	45
Figure 21: “crazy photo” application info	46
Figure 22: The seven elements of evaluating methodologies	47
Figure 23: Scope of SNSADLC methodology	49
Figure 24: Source code of the main page	62
Figure 25: Source code of the photo viewer	63
Figure 26: Source code of linking the application to Facebook API	63

## APPENDICES

Appendix 1 – Code

```

<div id="menu"><fb:tabs>
  <fb:tab-item href='<?php echo $baseUrl ?>index.php' title='Crazy Photo' <?php echo
    ($request == 'index' || $request == " || $request == 'junmagic') ? "selected='true'" : "" ?> />
  <fb:tab-item href='<?php echo $baseUrl ?>invite.php' title='Invite Friends' <?php echo
    ($request == 'invite') ? "selected='true'" : "" ?> /></fb:tabs>
</div><div id="album">
<?php
print "<form method='post' action='http://apps.facebook.com/junmagic^'>";
print "<br/>View friend's photos: ";
print "<fb:friend-selector idname='to' include_me='true'>";
print "<input value='Go' type='submit'>";
print "</form>";
$user = $facebook->require_login();
if (isset($_POST['to'])) {
  $ab = array($facebook->api_client->photos_getAlbums($_POST['to'],NULL));}
elseif (isset($_REQUEST['albumid'])) {
  $abu = array($facebook->api_client-
    >photos_getAlbums(NULL,$_REQUEST['albumid']));
  $ab = array($facebook->api_client->photos_getAlbums($abu[0][0]['owner'],NULL));}
else { $ab = array($facebook->api_client->photos_getAlbums($user,NULL));}
$countAlbum = count($ab[0]);
print "<div style='float: left;position:relative;border:solid 1px #bbbbbb;background-
  color:#f7f7f7;color: #8787b6;padding: 5px 2px;margin: 10px 0px;'>";
print "<div id='initialAlbum' style='display: block;float:left;'>";
for ($j=0;$j<$countAlbum;$j++){
  $photoAlbum = array($facebook->api_client->photos_get(NULL,$ab[0][$j]['aid'],NULL));
  print "<div style='float: left;padding: 2px 5px;width: 130px; height: 170px;'>";
  print "<div><a href='\"".$_SERVER['PHP_SELF']."'?albumid=".$ab[0][$j]['aid']."'><img
    src='\"".$photoAlbum[0][0]['src']."' border=0 /></a></div>";
  print "<div><a
    href='\"".$_SERVER['PHP_SELF']."'?albumid=".$ab[0][$j]['aid']."'>".$_ab[0][$j]['name'].
    "</a></div>";
  print "<div>".$_ab[0][$j]['size']." photos</div>";
  print "</div>";}
print "</div>";
if ($countAlbum>5){
  print "<div style='clear: both;margin: 0px 0px 5px 0px;position:relative;float:right;'><a
    href='\"#\" onclick='toggle(document.getElementById('singleAlbum'))'; return
    false;'><div id='text'>Click to see more</div></a></div>";}
print "</div>";
if (isset($_POST['to'])) {
  print "<fb:iframe
    src='http://majun.hostoi.com/junmagic/photo.php?to=".$_POST['to']."'
    width='750px' height='500px' scrolling='no' frameborder='0'>";}
elseif (isset($_REQUEST['albumid'])) {
  print "<fb:iframe
    src='http://majun.hostoi.com/junmagic/photo.php?albumid=".$_REQUEST['albumid']."'
    width='750px' height='500px' scrolling='no' frameborder='0'>";}
else { print "<fb:iframe src='http://majun.hostoi.com/junmagic/photo.php' width='750px'
  height='500px' scrolling='no' frameborder='0'>";}
print "</fb:iframe>";
?></div>

```

Figure 24: Source code of the main page

```

<div id="imageFlow">
  <div class="top"></div>
  <div class="bank">
    <?php
      require_once 'appinclude.php';
      $user = $facebook->require_login();
      if (isset($_POST['to'])) {
        $prints_id = (int)$_POST['to'];
      }
      else {
        if (isset($_GET['to'])) {
          $prints_id = (int)$_GET['to'];
        }
        elseif (isset($_REQUEST['albumid'])) {
          $prints_id = $_REQUEST['albumid'];
        }
        else {
          $prints_id = $user;
        }
      }
      if (isset($_REQUEST['albumid'])){
        $arr = array($facebook->api_client->photos_get(NULL,$prints_id,NULL));
      }
      else{
        $arr = array($facebook->api_client->photos_get($prints_id,NULL,NULL));
      }
      $count1 = count($arr);
      if ($arr[0] != NULL){
        for ($i=0;$i<$count1;$i++){
          $count2 = count($arr[0]);
          for ($k=0;$k<$count2;$k++){
            print "<a rel=\"". $arr[$i][$k]['src_big']. "\" href=\"". $arr[$i][$k]['link']. "\" title=\" \"
              target=\"_blank\">";
            if ($arr[$i][$k]['caption']!=""){
              $caption = $arr[$i][$k]['caption'];
              print $caption;
            }
            else{
              print " ";
            }
            print "</a>";
          }
        }
      }
    </div>
    <div class="text">
      <?php
        if ($arr[0] != NULL){
          print "<div class=\"title\">Loading</div>";
          print "<div class=\"legend\">Please wait...</div>";
        }
        else{
          print "<div class=\"title\">This user has no photos tagged by others so far.</div>";
        }
      ?>
    </div>
    <div class="scrollbar">  
    
    
    </div>
  </div>

```

Figure 25: Source code of the photo viewer

```

<?php
  $appapikey = '495e60adc4054d7d5b689139d6c2c72f';
  $appsecret = ' ';
  $facebook = new Facebook($appapikey, $appsecret);
  $user = $facebook->require_login();
  $appcallbackurl = 'http://majun.hostoi.com/junmagic/';
  try { if (!$facebook->api_client->users_isAppAdded()) {
    $facebook->redirect($facebook->get_add_url()); } } catch (Exception $ex) {
    $facebook->set_user(null, null);
    $facebook->redirect($appcallbackurl);}?>

```

Figure 26: Source code of linking the application to Facebook API

Appendix 2 – List of online survey questions and results (in italics)

1. Did you feel comfortable when using “crazy photo”?

A. Very comfortable. (1)

B. Comfortable. (8)

C. Uncomfortable. (0)

D. Not sure. (2)

2. Did you need any help when using “crazy photo”?

A. Never, it’s very easy for me to use. (10)

B. Sometimes, some functions I don’t understand. (1)

C. Always, I find it very difficult for me to use. (0)

D. Not sure. (0)

3. To improve “crazy photo”, can you give some comments or suggestions?

*User H: Good application! Despite I feel a bit faint when using it, perhaps the flicking speed could be slower.*

*User Q: I like the AJAX technique, it’s cool.*

*User N: I think the photos are not as clear as the original ones. Can you make them clearer?*

4. Please rate a grade of 1-5 what you thought of “crazy photo” (i.e., 5 would be very good, 1 would be very bad).

A. 5 (2)

B. 4 (7)

C. 3 (2)

D. 2 (0)

E. 1 (0)