International Journal of Financial, Administrative, and Economic Sciences (IJFAES)
Vol. (2), No. (2)

IJFAES

المجلة الدولية للعلوم المالية والإدارية والاقتصادية
الإصدار (2)، العدد (2)

June 2023

# "Identifying E-services of E-government Projects for Achieving Citizen Satisfaction"

**Wafaa Desoki Wahbi**
M.Sc. of Management, Faculty of Business, Alexandria University, Egypt
waffwahbbi@hotmail.com

**Sabryia Ali Elqenawi**
Ph.D. of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt
sabryiaaeq@alexu.edu.eg

**Abstract:**

E-government has many benefits at the level of individuals and governments. Most governments began to exploit information and communication technologies to provide information and services to their citizens. Despite the great role and the many benefits of e-government applications, many studies indicate that a large proportion of initiatives to implement E-government around the world failed in achieving the promised goals. Most e-government projects in developing countries have partially or totally failed. Adoption and use of e-services is still rather limited in most countries.

The growing attention paid by governments to invest in e-government projects and the lack of e-services usage refer to a gap between the real needs and expectations of users and the actual use and satisfaction. Responding to the high rates of failure and low rates of usage, the researchers were motivated to find new ways to reduce this gap and maximize the impact of investment in such projects. The aim of this paper

is to propose a novel approach for eliciting users' needs and requirements for e-government projects.

**Keywords:**

Electronic Services, E-government, Citizen Satisfaction.

## 1- Introduction

Governments exploit information and communication technologies to provide services to their citizens. E-government has many benefits at the level of individuals and governments. Some of these benefits, to name a few, are delivering information and services to citizens faster and easier; alleviating problems caused by dealing with limited experience or unskilled employees; broadening the participation of citizens in decision-making process; treating all citizens equally without any recommendation; upgrading the level of governmental performance; making the governments more transparent; simplifying government procedures which encourage business development; enhance interaction between three main actors in the society (government, citizens, and business) and helping trust building between governments and citizens.

Despite the great role and the many benefits of e-government applications, many studies indicate that a large proportion of initiatives to implement E-government around the world failed in achieving the promised goals [2]. Most e-government projects in developing countries have partially or totally failed. The success and failure rates are for government projects in developing countries as the following: 35% are totally failed; 50% are partially failed; and 15% are successful. These failures come with a high price. Among these potential costs of e-government project failure are financial costs, beneficiary costs, and political costs. Financial costs refer to the money invested in equipment, consultants, training, and so on. Beneficiary

costs refer to the loss of benefits (such as simplifying procedures, upgrading performance, reducing cost, increasing productivity, and earning the trust of citizens) that a successful e-government project would have brought. Political costs refer to the loss of face and loss of image for all parties involved in the failure whether they are individuals, organizations, and/or countries [14].

According to the United Nations survey 2014, Africa exhibits a regional digital divide with most internet activities, and its progress in the e-government development index, compared to the previous studies, is relatively slow. Countries in the region need to focus on building human capital, including ICT literacy, and bridging infrastructure gaps. Adoption and use of e-services is still rather limited in most countries. In Egypt, e-government uptake is very low. Only 11.3% of Egyptian households are aware of the existence of e-services. Only 2% of these households are using these services [28]. The growing attention paid by governments to invest in e-government projects and the lack of e-services usage refer to a gap between the real needs and expectations of users and the actual use and satisfaction.

Some issues that deepen and increase this gap are as follows:

- E-government projects have been often guided by supply-side factors and the demand side of the equation has received little attention.
- Technological possibilities rather than real user needs have determined the design of online services.
- Addressing the real needs of all diverse and highly heterogeneous user groups is problematic.
- E-government applications offer a wide variety of information and services which makes the content very complicated, and users need to be guided during the service process.

- There is no motivation for designers to focus on some considerations like attractiveness, visual appearance, and customization of the applications because no competition, and users are obliged to use the services.

E-government applications need more attention and effort to ensure the quality and success of these applications, to achieve a return on investment, and not be a waste of resources. Understanding and identifying the real needs and expectations of the citizens are crucial to fully capturing e-government benefits, improving service delivery to citizens in all segments of society, and raising governmental work effectiveness. So, the main concern of this work is to propose a novel approach for requirements elicitation in e-government projects.

## 2- Background

In this section, we introduce a theoretical background of e-governments, requirements engineering, and agile methodologies.

### 2-1 E-government

E-government refers to the use of information and communication technologies, particularly the Internet, as a tool to achieve better government [2]. There are four types of e-government. They are government to citizen (G2C); government to business (G2B); government to employees (G2E); and government to government (G2G) [22]. G2C provides information and services to citizens online.

An example of this type is Singapore's eCitizen portal (www.ecitizen.gov.sg). Singaporeans can access about 1600 e-services. Part of this e-services, 1300 services are completely transacted online without having to know which government agency is responsible for a particular service. eCitizen has three main areas: topics, e-Services, and highlights. Topics provide the key information grouped by areas of interest such as sports, transport, health, housing, and so forth. E-Services is a

directory of all services offered. Highlights provide latest news and upcoming events. eCitizen provides a cross agency and citizen centric information and services (figure 1).

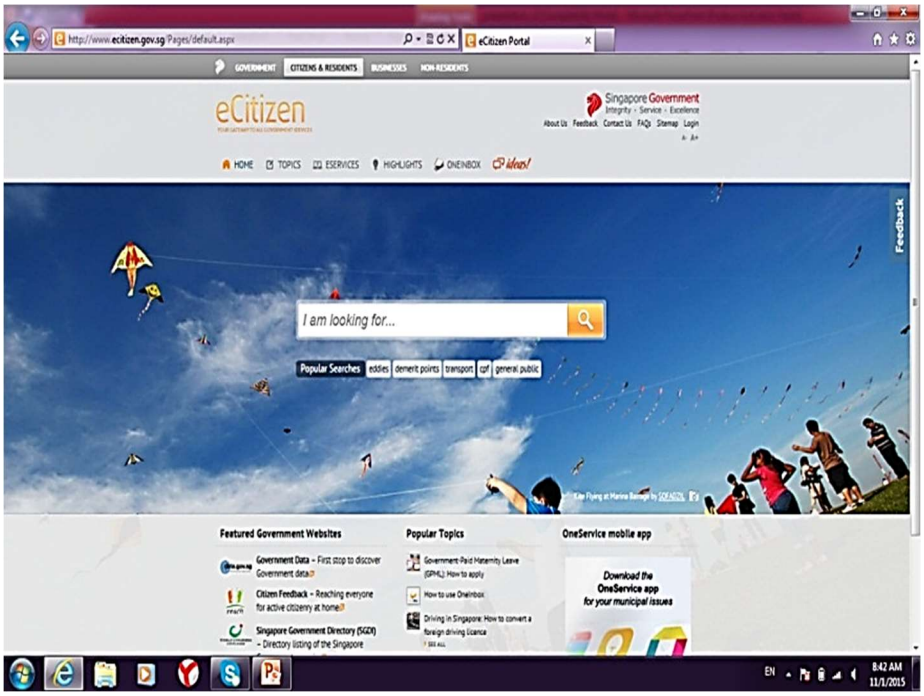


Figure (1): Singapore's e-Citizen Portal

G2B includes various services exchanged between government and business such as publishing policies, rules, and regulations; registering business; updating corporate information; tax payment, get financial help; and requesting answers to specific questions. An example of this type is shown in Figure (2) as the UK government provides online financial help for business including grants, loans, funding for small and medium-sized business, and business guide.

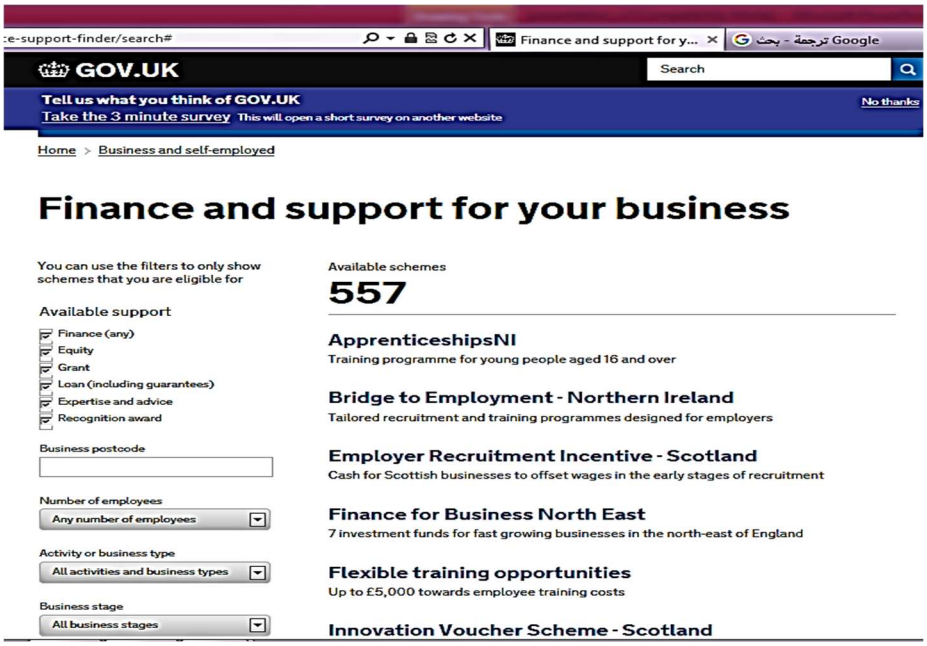

Figure (2): Gov.uk for Business

G2E provides information and services to government employees such as provision of human resource training; a view of payroll and tax information; and receive email notifications when new information related to employees is available. G2G aims to improve communication and data sharing between government agencies, and departments, and between different governments. An example of a successful G2G project is the Northeast Gang Information System. It is used by states in the Northeast to share information about street gangs [25].

E-government is seen as an evolutionary phenomenon. Its maturity can be expressed using sequential steps, for instance, from immature to mature with improved quality and advanced features such as a one-stop shop, personalization, online payment, and interoperability. E-government maturity model is a set of stages that determine the maturity of e-government applications. Many models have been proposed to assess

e-services provided to citizens. The most famous models are the Layne & Lee model [19]; Andersen & Henriksen model [3]; and the United Nations model [28]. The authors in [12] presented a comparison between 25 e-government maturity models. The findings show that models' stage numbers vary from 2 to 6 stages. Almost all the models contain a stage related to the availability of the portal on the web. Different names are used while they focus on the same aspects (Emerging information, Publishing, Online presence, and Cataloguing); A stage where citizens can interact with the government (Interaction); A stage where the citizens can transact with governments (Transaction); and a stage that covers advanced features such as information sharing between agencies (Integration). Moving towards the higher stages, with transactional and connected services, can be difficult. Such moving requires robust data protection systems and secure data sharing across government agencies [28].

There are many challenges facing the implementation of e-government applications. Some of these challenges are poor suitability of existing traditional software development methodologies to the dynamic nature of e-government projects [24]; lack of skills in the management; poor infrastructure; legal or policy barriers; digital divide; e-literacy; lack of protection of citizens' privacy; lack of transparency; the absence of trust between the governments and the individuals; lack of education and marketing programs which are needed to inform the people about the values of e-government services; the produced applications are unsecure; inaccessible; and incompatible [2]. IT designers understand technology but not the reality of government. Public officials and politicians understand the reality of government but not the technology [14].

## 2-2 Requirements Engineering

Developing successful e-government applications requires special attention to the requirements engineering process due to the number and diversity of stakeholders involved [26] and due to the complexity and variety of questionable requirements such as usability, accessibility, security, navigability, download speed, and interoperability. Requirements Engineering (RE) is defined to be the branch of software engineering concerned with identifying the functions of, and constraints on, software systems [31]. The RE process is the first stage in the development of a software system, and it feeds into the broader software development process. The literature contains more than one view of what activities constitute the RE process. The three major activities common to all other views are requirements elicitation, requirements documentation/specification, and requirements validation.

Requirements elicitation is the first activity in the RE process through which the requirements of a system are discovered and elaborated. Activities for requirements elicitation can be divided into five types: understanding the application domain; identifying the sources of requirements; analyzing the stakeholders; selecting the approaches, and tools to use; and eliciting the requirements from stakeholders and other sources. There is a variety of available approaches and tools for performing requirements elicitation. The most known techniques are interview, domain analysis, ethnography, and prototyping.

Requirements specification is the activity that results in producing the requirements specification/document, which is the output of the RE process. Requirements specification/document clearly and accurately describes each of the essential requirements. It should be understandable by all the system stakeholders and should generally contain a description of the problem under consideration, and a model of what is needed. There is a wide variety of ways for expressing a requirements

specification, ranging from informal natural language to more formal graphical and mathematical notations.

Formal methods provide a more rigorous representation based on mathematics and can be used to conduct mathematical proofs of consistency of specification and correctness of implementation. Formal methods are particularly suitable for critical systems i.e., ones in which potential financial or human loss would be catastrophic, and the cost of applying mathematically rigorous methods can be justified. The most common formal methods for the formal definition of functionality are Z, VDM, LOTOS, and the B-Method. Formal methods are slowly becoming more important. If their scope can be broadened to address wider system issues, they will become more useful.

Requirements validation is the activity through which possible problems in the requirements specification are detected before the specification is used for development. Good software requirements specification documents should be unambiguous, unrestrictive, complete, correct, Readable, modifiable, and organized. Although an ideal requirements specification is unachievable, a good and effective RE method should address as many as possible of these properties. The most known requirements validation techniques are Prototyping, Functional Test Design, and Reviews and Inspections.

The Standish and Gartner groups have, several times, conducted surveys that point out that project failures are perceived as caused by a lack of proper attention to requirements processes. The CHAOS report published in 1995 shows that almost half of the canceled projects failed due to a lack of requirements for engineering [9].

It became known that a large proportion of e-government application requirements are volatile. They tend to change quickly. The traditional RE approach, which focuses on collecting all requirements early, is no longer appropriate for the dynamic

context. Hence the need for a different RE approach appeared. Agile methodologies and practices emerged as an attempt to embrace higher rates of change more formally and explicitly in software requirements and customer expectations [30].

## 2-3 Agile Methodologies

Agile is a group of software development processes that are iterative, incremental, self-organizing, easy to learn, and welcome to changes at any point within the whole development life cycle. Agile methods are based on iterative development processes. Each iteration is a self-contained and mini-project. It covers requirements engineering, design, coding, and testing in the same as the traditional phases of any plan-driven method. Each iteration leads to an iteration release [30].

Agile methods accommodate a list of principles and a set of practices. Principles are essential truths that do not change over time. On the other hand, practices are applications of principles to a specific situation. So, the development team who hope to be agile must first decide whether they are consistent with agile principles, and then, select from the practices that support these principles [30]. Examples of agile principles are the highest priority is to satisfy the customer through early and continuous delivery of valuable software; Welcoming changing requirements; Building projects around motivated individuals; and face-to-face conversation; Working software is the primary measure of progress; and self-organizing teams [5]. Examples of practices are Acceptance testing; executable documentation; Continuous integration; energized work; Informative Workspace; Pair programming; Planning Poker; Short Iterations; Short Releases; and stand-up meetings. It is worth mentioning that some of the practices are common only in agile methods, and others are used in both agile and plan-driven methods [30].

By using agile methods, individuals and interactions are considered more significant than processes and tools, working software is selected over comprehensive

documentation, customer collaboration is appreciated more than contract negotiation, and the development process becomes responsive to a change more than following a plan [5]. The most widely used agile methods are Extreme Programming (XP) [13], and Scrum [27].

XP is a lightweight methodology for small-to-medium-sized teams developing software in the face of vague or rapidly changing requirements [4]. It focuses on building quality into the product as it is built [30]. XP is based on four values [7] and thirteen practices [30]. The four values are Communication, Simplicity, Feedback, and Courage. XP practices include sit-together, whole team, informative workspace, energized work, pair programming, stories, short iteration, quarterly release, slack, ten-minute build, continuous integration, acceptance- and unit-test-driven development, and incremental design. Several case studies have confirmed that XP improved quality, productivity, team morale, and customer satisfaction [30].

The Scrum development method was proposed in 1995 by Ken Schwaber [27], it has a very clear project management stress. Scrum is predicated on the concept that software development is not a cleanly defined process, but a series of ‘black boxes’ with complex input/output transformations. The Scrum process starts with the creation of the Product Backlog including the prioritized features required by the customer. The next phase of Scrum focuses upon a series of 30-day Scrum Sprints. During each Sprint the Scrum team will complete a working set of features that have been selected from the overall Product Backlog. Short meetings are held by the Scrum team on each day of the Scrum Sprint. Each daily meeting allows the team to monitor project status and discuss problems and issues. The conclusion of each 30-day Sprint involves the software demonstration of the product features that have been finished during that Sprint [7].

## 3- Related Work

The following are some important papers related to e-government projects, requirements engineering, and agile methodologies:

United Nations produces an e-government survey every two years as a tool for decision-makers to identify their areas of strength and challenges in e-government and to guide e-government policies and strategies. In the 2014 e-government survey, E-government Development Index (EGDI) was used to assess e-government development status. EGDI is a composite measure of three important dimensions of e-government, namely: the provision of online service, telecommunication connectivity, and human capacity. The world median of EGDI is 0.4712. The highest regional EGDI is Europe (0.6936), followed by Americas (0.5074), Asia (0.4951), Oceania (0.4086), and finally Africa (0.2661). Table (1) shows the regional e-government leaders.

Table (1): Regional e-government Leaders

| Region | Country | Global Rank |
|---|---|---|
| Africa | Tunisia | 75 |
| | Mauritius | 76 |
| Americas | United states of America | 7 |
| | Canada | 11 |
| Asia | Republic of Korea | 1 |
| | Singapore | 3 |
| Europe | France | 4 |
| | Netherlands | 5 |
| Oceania | Australia | 2 |
| | New Zealand | 9 |

Africa was led by Tunisia and Mauritius. Americas was led by the United States of America. Asia was led by the Republic of Korea. Europe was led by France. Oceania was led by Australia. Even though Mauritius is one of the small islands developing states, with a small land and population, the government of Manutius has put effort into developing its online portal. EGDI of Mauritius improved from 93 in 2012 to 76 in 2014 [28].

In [26], a literature review is conducted to explore the various existing classifications of e-government stakeholders and their required benefits. This review informs a proposal for a typology of stakeholder roles, and for a typology of stakeholder benefits. The proposed typology of e-government stakeholders' roles consists of twelve roles. They are people as service users, people as citizens, businesses, small-to-medium-sized enterprises, public administrators (employees), other government agencies, non-profit organizations, politicians, e-government project managers, design and IT developers, and researchers and evaluators. The typology of stakeholder benefits is a list of stakeholder interests such as accountability, confidentiality, democracy, ease of use, economic growth, employment and jobs, integration among government units, justice, quality of service, and security, among others. Both typologies are used to construct a stakeholder benefits analysis tool (SBAT), which can be used to map stakeholder roles to stakeholder benefits. For example, the highest-ranked benefits associated with citizens are transparency, openness, and trustworthiness. The highest-ranked benefits associated with businesses are economic growth and productivity. The highest-ranked benefits related to public administrators are empowering employees and reducing the admin burden.

The authors in [20, 21] proposed seven criteria that need to be addressed by a successful Web engineering process. These criteria are short development life-cycle times; delivery of bespoke solutions and different business models; multidisciplinary

development teams; small development teams working in parallel on similar tasks; business analysis and evaluation with end-users; Requirements capture and rigorous testing; and Maintenance of WBAs. These criteria are then used to assess the suitability of a representative sample of well-known software engineering processes for Web engineering. The results showed that support for short development life-cycle times is better addressed with agile methods; neither traditional nor agile processes suggest support for business process reengineering; The support for multidisciplinary development teams is limited; There is a need for stronger focus on Analysis, Evaluation, Requirements, and Testing phases if traditional software engineering processes are to become more suitable for Web engineering; and the support for maintenance seems to increase the closer processes get to the agile community. The proposed seven criteria were also used to motivate the definition of the Agile Web Engineering (AWE) process. AWE is a lightweight process based on the principles given in the Agile Manifesto. it is differentiated from other agile and plan-driven processes by specific application to Web engineering and its particular focus on three criteria for a Web engineering process, namely, 'Different business models', 'Small development teams working in paralle1' and 'Analysis and Evaluation'. AWE has an iterative and incremental process life-cycle that explicitly focuses on the phases required in Web engineering.

In [17], the authors investigated the adherence degree of agile RE practices in traditional software development organizations. An approach is proposed for the purpose of investigation. Such an approach constitutes seven steps. The first step is to identify the list of agile RE practices that their adherence degree will be measured. The second step is to design a questionnaire by formalizing the agile RE practices in the form of questions. The third step is to choose the projects that will be subjected to the study. The fourth step is to conduct semi-structured interviews with the participants. The fifth step is to review the requirements documents of the projects

to get more information about project requirements. The sixth step is to analyze the information gathered to identify the adherence degree of each agile RE practice in each single project. The seventh step is to aggregate all the results of each agile RE practice in all projects in one table. The mean of all adherence degree of an agile RE practice in all projects is calculated and then agile RE practices are ranked. Such a proposed approach is applied on five different projects, from four different organizations, to measure the adherence degree of a set of common agile RE practices. These common practices are face-to-face communication, iterative RE, extreme prioritization, constant planning, prototyping, review & test, and test-driven development. The results show that traditional software development organizations are applying agile RE practices to different degrees without their awareness. Iterative RE has ranked first as the most applied agile RE practice. The least applied agile RE practice is test-driven development.

The researchers in [18], conducted a systematic literature review on agile RE practices and challenges to illustrate how traditional RE issues are resolved by agile RE practices. The review identified five challenges traceable to traditional RE that were overcome by agile RE. These challenges are communication issues, over-scoping, requirements validation, requirements documentation, and rare customer involvement. Communication issues occur when there is a lapse in providing required information to relevant people. This challenge is overcome by frequent face-to-face meetings, collocated teams, onsite customers, and cross-functional teams. Over-scoping is defined as setting the scope of the project that is too large for the resources provided. In agile methods, developers receive a list of features that are constantly prioritized by the customer. Thus, the chance of having to repeat allocation is reduced. Agile methods deal with requirements validation by requirements prioritization and prototyping. The customer continues with prioritization of requirements in every iteration. Thus, the less important requirement

remains on hold. Prototyping helps in providing the customer with a blueprint of the product, and therefore, helps in validating the requirements. Requirements documentation means agile methods rely on tacit knowledge and frequent face-to-face communication in place of documentation for explicitly supporting knowledge sharing. Rare customer involvement is where customer involvement plays an important part in agile methods. In traditional software development, customers obtain the product after development. In contrast with agile methods, customers remain aware of the progress that is made throughout the development lifecycle.

A 4-dimensional analytical tool (4-DAT) has been presented for the purpose of analysis and comparison of agile methods. 4-DAT helps in selecting a more suitable agile method for a particular project. 4-DAT examines the agile methods from four dimensions. The first dimension is method scope characterization. It is a set of eight scope items such as project size and team size. The second dimension is the agility characterization. It is a set of five agility features that can be used to measure the degree of agility in the method quantitatively. Examples of these features are flexibility and speed. The third dimension is the agile values characterization. It is a set of six values that can be used to examine the support of agile values in different practices of agile methods. An example of these agile values is individual interactions over processes and tools. The fourth dimension is the software process characterization. It is a set of four components of the software process. They are the development process, project management process, software configuration control process, and process management process [23].

A framework for developing e-government applications has been proposed in [24]. The framework adopted an agile software product management (SPM) approach for requirements refinery [29]. The agile SPM approach is built upon the assumption that to develop software with a high probability of success, a functioning team of product managers is required to work cooperatively with the development team to

supply approved and well-defined requirements. The agile SPM approach is Scrum-inspired. It adopted Scrum principles to improve the ability to handle large amounts of complex requirements. The framework presented in [24], integrated agile SPM, XP, Scrum [27], and 4-DAT [23]. The framework began with requirements refining and effort estimation. Agile requirements refinery includes four stages: the vision, the theme, the concept, and the definition. Vision is the main idea of the project. A product manager converts it to a set of themes to define the business problem and the functionality. To ensure the technical feasibility of the themes, they are reviewed by the development team. Then each theme is refined into a set of concepts, which includes the set of solution stories. The solution stories constitute the set of backlogs to be achieved. Each backlog is examined based on the task's scope and the first dimension in 4-DAT to choose the more suitable agile methods (XP or Scrum) for that task.

The authors in [10, 11] conducted a literature survey to investigate to what extent agile methods support the identification and modeling of non-functional requirements (NFRs). The results showed that agile methods have not adequately identified and modeled NFRs such as security and scalability. Non-functional Requirements Modeling for Agile Processes (NORMAP) has been proposed to identify and classify NFRs, treat NFRs as first-class artifacts, and link them to functional requirements (FRs). A newly proposed Agile Requirement taxonomy was classified as Agile Use Case (AUC), which represents a FR, Agile Loose Case (ALC), which represents a NFR, or Agile Choose Case (ACC), which represents the potential solutions of NFRs. These three artifacts, AUCs, ALCs, and ACCs, form the three fundamental building blocks of the NORMAP Methodology. The term "Loose" was derived from the notion that NFRs are "soft" goals. Such goals are usually hidden and difficult to identify. NORMAP Methodology can be used manually or semi-automatically. The manual framework is called Non-functional Requirements

Modeling for Agile Manual (NORMANUAL) practices. The semi-automatic tool is called The Non-functional Requirements Modeling with Agile Automatic (NORMATIC). NORMATIC uses Natural Language Processing for NFRs identification. It supports linking artifacts in three different configurations: AUC-to-ALC, ALC-to-ACC, and ALC-to-ALC. In the first case, a FR is linked to an NFR. The second case links an NFR to a potential solution. In the third case, an NFR is positively or negatively impacting another NFR. NORMATIC applies different color-coding schemes to improve agility in a visual environment.

## 4- The Proposed Approach for E-services elicitation

In response to the high rates of failure and low rates of usage, we are motivated to find new ways to reduce the gap between the real needs and expectations of the users and the actual use and satisfaction. Reducing this gap will increase usage rates of e-government applications and maximize the impacts of investment in such projects.

So, our main objective is to propose a novel approach for eliciting user needs and requirements for e-government applications. As depicted in Figure (3), constructing the proposed approach contains the following steps:

1. Elaborating requirements elicitation approach for e-government projects.
2. Applying the proposed approach to many e-government applications
3. Evaluating the proposed approach and measuring the actual use and satisfaction
4. Finalizing the approach

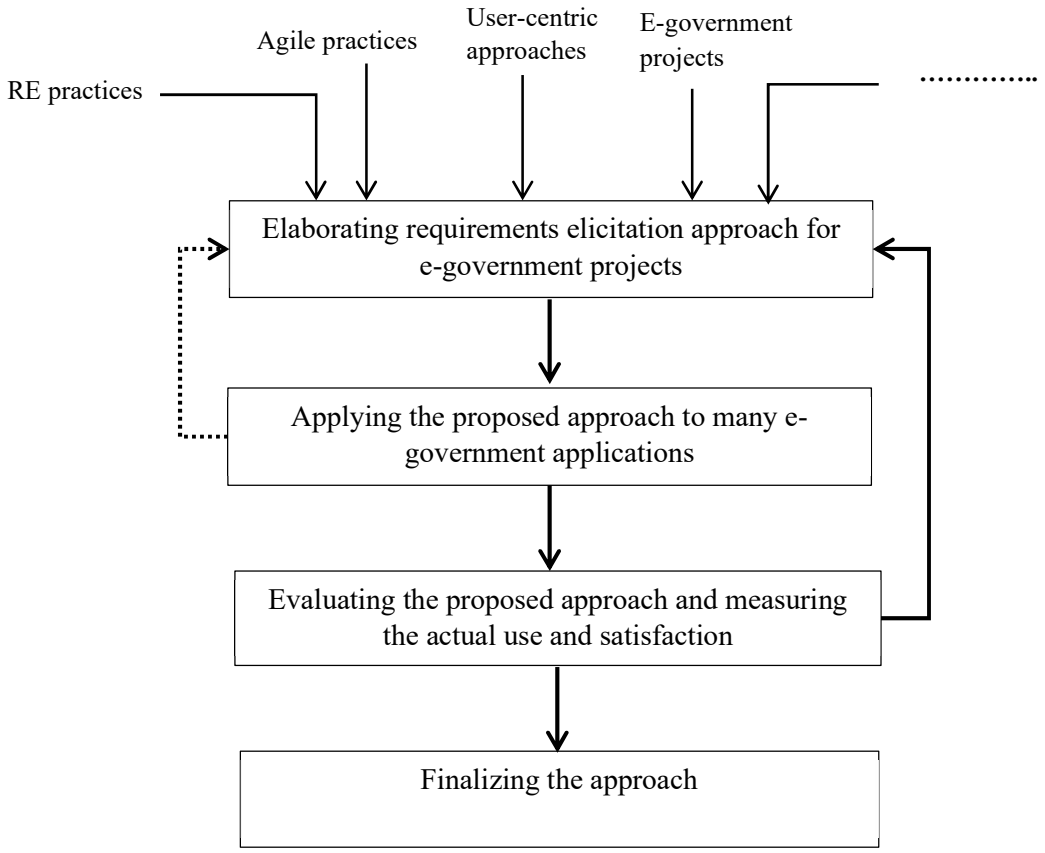


Figure (3): Requirements Elicitation Approach

**(1) Elaborating requirements elicitation approach for e-government projects:**

To elaborate on the requirements elicitation approach, the researchers will mainly focus on requirements engineering practices, agile practices, user-centric approaches, and e-government projects. The dotted box in Figure (3) refers to the possibility of adding other paradigms, such as fuzzy logic, in the elaboration process.

Firstly, "Requirements" is the central and core word that frequently recurs when discussing the causes of success and failure of software projects. Centering on requirements will lead to accurate and realistic identification of the needs of citizens; prevent the inclusion of unneeded features that don't use; increase citizen satisfaction; and save money because potential problems can be fixed early. Secondly, large-scale agile software development remains an active research area. We aim to take advantage of the success of agile methods in a lot of small and medium projects and extend them to large projects like e-government. Thirdly, many researchers noticed that there is a lack of data and investigations into the demand side of e-government services. Technological possibilities are the most used determinant for the design of online services rather than real citizens' needs. Nowadays, there is a tendency to progress to more user-centric e-government approaches and in this paper, the researchers attempt to be in line with this tendency and bring user-centric approaches into practice.

**(2) Applying the proposed approach to many e-government applications:**

Elaborating requirements elicitation approach from the mentioned sources is not sufficient. Specific attention should be given to practical experience. So, it is vital to apply the proposed approach in many e-government applications for the sake of identifying the needs and expectations of users to discover any weak points and make the required modifications.

**(3) Evaluating the proposed approach and measuring the actual use and satisfaction:**

To evaluate and determine the true value of the proposed approach, the researchers will measure the actual use and satisfaction level regarding e-government applications developed by the proposed approach and compare the results with other applications developed by a traditional method.

**(4) Finalizing the approach:**

Applying and evaluating the proposed approach will inform the researchers whether they have missed some important users' requirements and/or focused on some requirements, which are not the primary concerns of the citizens. So, we can return to the first step and re-elaborate and finalize the approach.

## 5- Conclusion

Understanding and detecting the real needs and expectations of the citizens are crucial to fully capturing e-government benefits, improving service delivery to citizens in all segments of society, broadening citizen participation in the decision-making process, and upgrading the level of government performance. The development process of e-government applications is facing many challenges. Many indicators refer to a gap between the real needs and expectations of the users and the actual use and satisfaction. Reducing this gap will increase the usage rate of e-government applications and maximize the impact of investment. This paper aims to propose an enhanced approach for requirements elicitation in e-government projects. It is a demand-oriented approach focused on the requirements elicitation phase and based on requirements engineering practices, agile practices, e-government projects, and user-centric approaches.

## References

[1] Hisham M.E. Abdelsalam, Hatem A. Elkadi, and Sara Gamal, "Setback and Remedy of Local E-Government Projects: A Case Study from Egypt", ICEGOV2010, 2010.

[2] Tamer Almarabeh and Amer Nizar Abuali, "A General Framework for E-Government: Definition Maturity Challenges, Opportunities, and Success", European Journal of Scientific Research, 39 (1), 29-42, 2010

[3] Kim Viborg Andersen and Helle Zinner Henriksen, "E-government Maturity Models: Extension of the Layne and Lee Model", Government Information Quartery, 23, 236-248, 2006.

[4] Kent Beck, " eXtreme Programming Explained: Embrance Change", Addison-Wesley, 2000

[5] Kent Beck et al. "The Agile Manifesto", [online]. http://www.agilealliance.org/the-alliance/the-agile-manifesto/ , 2001.

[6] Akemi Takeoka Chatfield and Omar Alhujran, "A Cross-Country Comparative Analysis of E-Government Service Delivery Among Arab Countries", Information Technology for Development, 15 (3), 151-170, 2009.

[7] Peter Clutterbuck, Terry Rowlands and Owen Seamons, "A Case Study of SME Web Application Development Effectiveness Via Agile Methods" The Electronic Journal Information Systems Evaluation, 12 (1), 13-26, 2009.

[8] Nagy Ramadan Darwish, "Improving the Quality of Applying eXtremeProgramming (XP) Aproach", IJCSIS International Journal of Computer Science and Information Security, 9 (11), 2011.

[9] Armin Eberlein and Julio C.S.D.P. Leite, "Agile Requirements Definition: A View from Requirements Engineering" In: Proceedings of International Workshop on Time-Constrained Requirements, 2002.

[10] Weam M. Farid, "The Normap Methodology: Lightweight Engineering of Non-Functional Requirements for Agile Processes", In: 19th Asia-Pacific Software Engineering Conference, IEEE, 2012.

[11] Weam M. Farid and Frank J. Mitropoulos, "NORMATIC: A Visual Tool for Modeling Non-Functional Requirements in Agile Processes", In: Southeastcon, IEEE, 2012.

[12] Abdoullah Fath-Allah, Laila Cheikhi, Rafa E. Al-Qutaish and Ali Idri "E-government Maturity Models: A Comparative Study", International Journal of Software Engineering & Applications (IJSEA), 5 (3), 2014.

[13] Rostislav Fojtik, " eXtreme Programming in Development of Specific Software", Procedia Computer Science, 3, 1464-1468, 2011.

[14] Richard Heeks, "most e-government for development projects fail: how can risks be reduced?", Institute of Development Policy and Management (IDPM), Univerity of Manchester, 2003.

[15] Richard Heeks and Savita Bailur, "Analyzing E-Government Research: Perspectives, Philsophies, Theories, Methods, and Practice", Government Information Quarterly, 24, 243-265, 2007.

[16] Elizabeth Hull, Ken Jackson and Jeremy Dick, "Requirements Engineering", 3nd ed., Springer, 2011.

[17] Mennatallah H. Ibrahim and Nagy Ramadan Darwish, "Investigation of Adherence Degree of Agile Requirements Engineering Practices in Non-Agile Software Development

Organizations", (IJACSA) International Journal of Advanced Computer Science and Applications, 6 (1), 2015.

[18] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva and Shahaboddin Shamshirband, "A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges", Computers in Human Behavior, 2014.

[19] Karen Layne and Jungwoo Lee, "Developing Fully Functional E-government: A Four Stage Model", Government Information Quarterly, 18, 122-136, 2001.

[20] Andrew Gregory McDonald, "The Agile Web Engineering (AWE) Process", PhD Thesis, 2004.

[21] Andrew Gregory McDonald and R. Welland, "Agile Web Engineering (AWE) Process: Multidiscplinary Stakeholders and Team Communication", In: ICWE'03 Proceedings of the 2003 International Conference on Web Engineering, , 515-518, 2003.

[22] P.J.Pascual, "e-government", UNDP-APDIP, 2003.

[23] A. Qumer and B. Henderson-Sellers, "Measuring Agility and Adoptability of Agile Methods: A 4-Dimensional Analytical Tool", In: IADIS International Coneference Applied Computing , , 503-507, 2006.

[24] Atef Tayh Raslan and Fatma El-Licy, "An Agile-Based Framework for Development of E-Government Projects", IJICIS, 12 (1), 2012.

[25] M. Rouse, "government to government" [online].
http://whatis.techtarget.com/definition/government-to-government-G2G, 2010.

[26] Jennifer Rowley, "E-Government Stakeholders-Who are They and What do They Want?", International Journal of Information Management, 31, 53-62, 2011.

[27] Ken Schwaber, "Scrum Development Process", In: 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA), 1995.

[28] United Nations, "2014 e-government for the future we want", New York, 2014.

[29] K. Vlaanderen, S.Jansen, S.Brinkkemper, and E.Jaspers, "The Agile Requirements Refinery: Applying Scrum Principles to Software Product Management", Information and Software Technology, 53 (1), 58-70, 2011.

[30] Laurie Williams, "Agile Software Development Methodologies and Practices", Advances in Computers, 80, 1-44, 2010.

[31] Pamela Zave, "Classification of Research Efforts in Requirements Engineering", ACM Computing Surveys, 29 (4), 315-321, 1997.