ISSN (Print): 0974-6846
Indian Journal of Science and Technology, *Vol 11(31), DOI: 10.17485/ijst/2018/v11i31/128780, August 2018* *ISSN (Online): 0974-5645*

A Proposed Framework for Enhancing Story Points in Agile Software Projects

Nisma Gaffar[1], Hanan Moussa[2], Amr Kamel[3] and Galal H. Galal-Edeen[4]

[1]Faculty of Computers and Information, Cairo University Cairo, Egypt; nisma.algand@gmail.com

[2]Information Systems Faculty of Computers and Information, Cairo University Cairo, Egypt; h.moussa@fci-cu.edu.eg

[3]Computer Science Faculty of Computers and Information, Cairo University Cairo, Egypt; a.kamel@fci-cu.edu.eg

[4]Information Systems Faculty of Computers and Information, Cairo University Cairo, Egypt; galal@fci-cu.edu.eg

Abstract

Objective: we aim to enhance the agile sizing units to lessen the subjectivity of estimations and the dependency of the personal experience by introducing factors that would provide a clear basis and estimation guidance for size estimation, especially Epics estimation yet maintaining the agility of size estimation. **Methods/Analysis**: we surveyed a number of approaches that used to estimate software size in traditional software development. However, these size approaches have limitations and may not be suitable for agile software projects. In agile projects, size is estimated based on the experience of the team by Story Point (SP). SP is the common sizing unit that is assigned based on the relative size of the User Story. However, Story Point as it stands is subjective and is not defined in a standard way, and is ill-suited to large projects. **Findings**: in this research, we propose a framework and introduce a new sizing unit for estimating Epics called Enhanced Story Point sizing unit (ESP). Enhanced Story Point is calculated based on three factors affecting size estimation especially in the early phase of the software projects i.e. requirements phase. These factors are Uncertainty, Functionality, and Complexity of project requirements. We applied our framework on three case studies that used an agile process in order to manage their work and each team used Story Points unit as a unit of the project size. We evaluated our results by using two measures; Effort Variance and Magnitude Relative Error (MRE) for each Epics. Then we compared the results before and after using our framework i.e. when using SP and ESP. **Improvement**: an empirical evaluation demonstrates that our proposed work outperforms traditional estimation by Story Point in effort variance, especially in case of low-experience in estimation using existing Story Points of the team.

Keywords: Complexity, Epic, Estimation size, Story Points, Uncertainty, User Stories

1. Introduction

Estimating software projects is one of the most important and essential software development activities and at the same time, is one of the challenges and complex processes[1]. An accurate estimation can contribute to increasing efficiency and improve the performance of the project[2]. The most important decisions are taken at the early stages of software development projects. In these early stages, requirements are the main source for estimation. Estimating project basic metrics such as size, effort, cost, and changes, all depend on requirements[3].

Therefore, problems in requirements lead to a project inaccurate estimation. Many factors and attributes may

*Author for correspondence

affect project estimation such as uncertainty of the requirements, complexity, requirements quality and others. Requirements uncertainty is one of the root causes of changing requirements. These uncertainties are increasing in fast-changing markets[4]. Uncertainty requirements is natural, especially when the size estimate is being made early in the development process. Here, the clients and developers are still negotiating, trying to determine what needed[5]. Thus, many attributes can cause uncertainty in requirements. Five attributes, which consider encompassing many others attributes are discussed[4].

Knowing and measuring the complexity of the requirements may also affect estimation value. Software complexity is a well-known concept within the software engineering domain. The complexity of requirement can be measured by many factors such as Input-Output Complexity (IOC), Functional Requirements (RE), Non-Functional Requirements (NFR), Personal Complexity Attributes (PCA), Product Complexity (PC), Design Constraint Impose (DCI), Interface Complexity (IFC), User/Location Complexity (LCO) and System Feature Complexity (SFC)[6].

Estimating the effort, cost, and duration is an important estimation metrics, which may affect the project success[7]. These metrics can be estimated by defining the size of the project which can be used as an input attribute.

To estimate the size of a software project, various estimation approaches can be used. We classified these approaches into three approaches; Metrics-based approach such as Line of Code, Functionality-based approach such as Functional Points (FP), Use Case Points (UCP) and others. This approach required more details and attributes to define and estimate the size of project, which is may not available in the early phase of development i.e. requirements elicitation. Third approach is Relative-based approach, which is simple to calculate, however it depends on estimator's experience. This approach is based on the relative comparison between projects or requirements such as Expert-Judgment, Analogy, and Story Points (SP). Story Points is a relative sizing unit, which used to estimate the size of agile software projects.

Agile requirements may less formal and not always explicitly denoted as requirements engineering, also the requirements are not always documented[8]. Requirements in agile software projects are written in different terms based on the level of details: Epics, User Stories, and others. Epic is a high-level requirement that is mean the functionality is not clear and need to be identified. The Epic is defined as 'a big User Story', which should be broken down into smaller User Stories. However, splitting stories is not a simple task in agile projects[9]. User Story is a brief statement that describes the functionality or a piece of work the system needs to do for the user[10].

Story Points is a subjective unit for estimating how big 'size' of User Story[11]. Most agile teams depend on experts' subjective assessment, which may lead to inaccuracy and inconsistencies between estimates since metrics based on Story Points cannot be used to compare teams, projects, departments or organizations[12]. Agile teams used several techniques to assign SP into User Story such as planning poker[13–16], Team Estimation Game (TEG)[17], T-shirt technique[18] and others techniques[13].

In case of estimating Epic, no many research papers have been discussed how to estimate the Epics. However, according to teams' opinions we found that: some teams use a planning poker technique with its large number 40 or 80[19]. Other teams use 20, 40 or 100[20]. The terms "infinity" or "too big" are also used[20]. Some teams use the Low (L), Medium (M) and High (H)[21,22]. These opinions do not base on a clear rule or criteria to guide the estimation of Epics. Most teams are not recommending estimating Epics and they ask from manager to split the Epic into User Stories before the estimation.

In order to improve estimating size in agile software projects and enhance Story Points sizing unit, several research papers have been introduced over the years. We classified these endeavors into three types of enhancement. First type of enhancement is using Story Points with some modifications[23–26]. A case study[26] was introduced to estimate size by using Story Point. This study proved that the Story Point technique could increase the efficiency of the team. A framework has proposed to enhance the effort estimation by using Sprint Points instead of Story Points[24]. However, Sprint Points are calculated based on the initial Story Points, then adjusting these SP by applying Sprint Points approach. The authors in[23] proposed a Story Point approach with various optimization techniques, which used to improve the estimation accuracy. In[25] proposed a Story Point approach to estimate the effort in agile software projects. However, this work is not supported by any case study or any data analysis to evaluate the performance of this approach.

Second type is enhancement by using Functional Size Measures (FSM) such as Function Points (FP)[9,27,28], COSMIC[29] and Use Case Points (UCP)[30] and others. The authors in[9] examines whether Function Points would be compatible with Story Points in agile projects. In[27] proposed a study that used Function Points (FP) in agile projects and tried to correlate them with Story Points. The result was as follows: FP is not used to estimate how long it will take to implement a story or complete a release. FP correlates poorly with estimated and actual task hours. Using FP is not as accurate to predict how long it will take to complete a project already underway.Function Points sizing unit was used in addition to Story Point to estimate the effort of a project development[28].

In[29] used the COSMIC method to quantify the functional size of each User Story in order to enhance the estimation. They counted the four sub-types (Entry (E), eXit (X), Read (R) and Write (W) and the sum of it is returned as the number of COSMIC Function Points (CFP). Use Case Points technique also used to estimate User Stories[30]. They proposed to assign Story Points based on complexity and time. The complexity of the User Stories was categorized into three levels easy, moderate and complex based on the Use Case Points attributes. However, the time factor is not discussed in detail. This research was not evaluated by case study.

In[31] introduced a comparative study between Function Points, Case Points and Story Points. They concluded that Story Points are useful for estimating with historical SP data, easy to calculate, difficult to validate, low objectivity and high subjectivity.

Third type of enhancement is using adjustment factors. The authors in[32] proposed a model that incorporates eight factors affecting the cost, size, and duration of the agile project. Those factors are identified on the grade of low, medium and large. The sum of these grades is called as Unadjusted Value (UV). The project is decomposed into User Stories and a Story Point count is assigned to each story, and then calculated the New Story Point (NSP) by adding the UV to the Story Point that was assigned before. Another study[33] addresses factors that directly impact the agile estimation process, but these factors are proposed without any specific values or weights assigned to them.

However, the enhancement using first type is still human-dependent estimation, which based on the personal experience. Estimating using second type, will add more complexity of the calculation and in the same time, the calculation steps itself may take more time to complete. And according to the third type, more time was spent in the estimation phase and still, the initial SP is based on the experience of the estimator.

Estimating by Functionality-based approach, have a set formula and a clear basis to calculate the size of the requirements of the project. If the agile sizing is lacking a clear basis, thus the idea of using the basis of objective sizing units that is used in the traditional approaches as a reference in agile projects sizing is worth considering. However, estimators in functionality-based approach spend a considerable time in estimating the size of the project. In addition, the exercise of estimation is not simple[34]. This complexity and long time needed to estimate the size of the requirements does not suite the simplicity and the agility that characterize the agile manifesto and the agile principles[35,36]. Thus, there is a need to enhance the agile sizing units to lessen the subjectivity of estimators and the dependency of the personal experience of estimators by introducing factors that would provide a clear basis for size estimation yet maintaining the agility of size estimation.

2. Proposed Framework

Our overall research goal is to build an estimation framework to estimate the Epics (big User Stories) and produces an Enhanced-Story-Point (ESP) as a sizing unit of Epic size. Thus, our proposed framework can be used when an Epic has just been created. ESP is based on three factors that may affect the estimation in the early phase of life cycle. These three factors are Uncertainty, Functionality, and Complexity. Each factor has some attributes and is introduced by a number of steps to get ESP.

Figure 1 shows our proposed framework that we have designed for enhancing Story Point. The proposed framework provides detailed steps for estimating the size of an agile project and provides basis for the estimation unlike the existing agile estimation techniques. Our proposed framework has four main steps: First, for each Epic, rate the uncertainty attributes. Second, for each Epic, list the Functional Requirements/User Story then count the functionality. Third, for each Functional Requirement/User Story, count the complexity. Fourth, calculate the Enhanced Story Points (ESP).

A brief description of our framework can be summarized as follows: for each Epic, three main steps are performed; first, rate the uncertainty attributes between team members and compute the uncertainty value. Uncertainty has five attributes that consider the

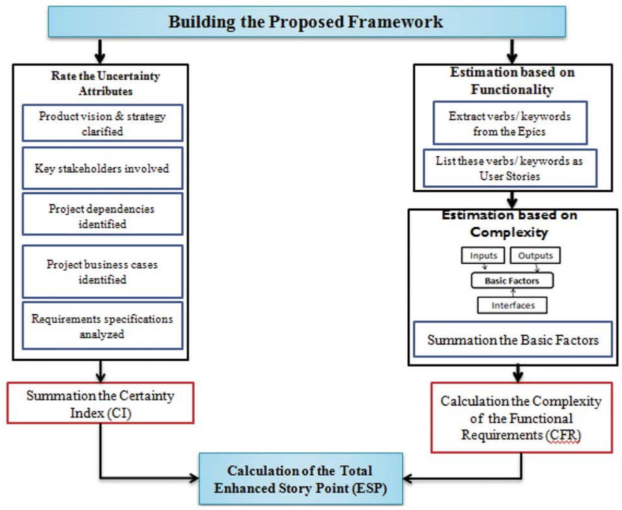

Figure 1. Our proposed framework.

reasons for requirements uncertainty. Second, extract keywords from the Epic by using textual analysis approach then list it as functional requirements or User Stories. Third, for each Functional Requirement/User Story, we calculate the basic factors of the complexity. These basic factors: input, output, and interface. In the end, we calculate ESP as a summation of the uncertainty value with the complexity of the functional requirements. We will discuss these four steps in detail in the following section.

Step 1: Rate the Uncertainty Attributes

We use five attributes of the requirement uncertainty to measure the uncertainty in the each Epic. In this step, we define five values i.e., 1-5 for each attribute, 1 is less uncertain and 5 are more uncertain. For each Epic, team members follow some steps to calculate the uncertainty value as follows:

1. Each member of the team chooses one value based on his/her expectations and experience
- For example, the attribute "Product vision & strategy clarified"; the developer rates it as '2' and tester said that he thinks it is '3, another member gives it '3'.
2. All the team members will discuss to agree on one value for each attribute.
- For example, the attribute "Product vision and strategy clarified", after discussion all team members reach value 3. The Delphi technique[37] can be used for this step.
3. Sum all the index as Certainty Index (CI)

$$CI = \sum_{1}^{5} Index \tag{1}$$

Step 2: Count the Functional Requirement (FR)

Epics are written with little details of functionality so the functional requirements may not be clear for the team especially in the first meeting. Therefore, we apply a Textual Analysis approach to extract some keywords (verbs), which indicates functionality. Textual analysis is a technique that is used to extract information systematically from unstructured data[36,37]. Two steps are followed to count the FR:

- Extract some keywords (verbs) from the Epic.
- List these keywords (verbs) as Function Requirements/ User Stories.

Then, count how many FR in each functional requirement or User Story as follows.

$$FR = No.of\ \ Functionality \tag{2}$$

Step 3: Calculate the Basic Factors of Complexity

For each FR/US, we will count the Basic Factor of the complexity i.e., Input, Output, and Interface.

$$BF = No.of\ Inputs + No.of\ Outpu + No.of\ Interfaces \tag{3}$$

Then, calculate the complexity of the functional requirements as:

$$CFR = \sum_{1}^{n} (FR * BF) \tag{4}$$

Where n is the number of functional requirements.

Step 4: Calculate the Enhanced Story Point (ESP)

ESP is the calculation of uncertainty index and complexity of functional requirement.

$$ESP = CI + CFR \tag{5}$$

3. Case Studies

The case study investigates whether a development team that applied the Enhanced Story Point (ESP) achieved improved performance of estimation compared to when they use Story Point (SP). The reminding part of this section describes the projects and how data were collected.

3.1 Case studies and Teams Studied

We will apply our framework on three case studies. First case study is Lead Management Project, one project of a company in Egypt. This project consists of three Epics. This company has a medium level of experience (3-5 years). We named it 'Case Study 1'. The second case study is a reference example introduced by Chon[11], this case study is a learning case that discussed a game development. Cohn, who introduced this case study have a high level of experience. We named it 'Case Study 2'. The final case study is a graduation project. This project is Visualization Dashboard and web application for a company. This case study consists of four Epics and the team members have no or low level of experience to deal with agile projects. We named it 'Case Study 3'.

3.2 Data Collection

The data used in this study is collected as 'estimated size by Story Points', 'estimated effort hours' and 'actual effort hours' per Epic. Our dataset consists of three case studies with 12 Epics and 88 User Stories. Each case study has a different scenario and different numbers of the team members. They counted the Story Points and estimated effort based on their experience.

3.3 Measures

We use two measures to assess our result as follows:

$$\textit{Effort Variance}\,(\mathrm{EV}) = \frac{\textit{Actual Effort} - \textit{Estimated Effort}}{\textit{Estimated Effort}} \tag{6}$$

$$\textit{Magnitude of}\ \text{Relative Error}\,(\textit{MRE}) = \frac{\textit{Actual Effort} - \textit{Estimated Effort}}{\textit{Actual Effort}} \tag{7}$$

4. Results and Analysis

After we analyzed the data that was estimated by Story Points based on the experience of the team, we calculated the Effort Variance between estimated and actual effort (EV_SP) and Magnitude of Relative Error (MRE_SP) to show the estimation error in case of estimating by SP. Then, we applied our framework and estimated the same Epics using Enhanced Story Point (ESP) which discussed in Section 2. In order to evaluate our work and compare/analyze the results, we should know the estimated effort in case of using our proposed framework i.e. by using ESP (Estimated Effort (ESP)). Then, we calculated the Effort Variance between estimated and actual effort (EV_ESP) and Magnitude of Relative Error (MRE_ESP). Finally, we compared the estimated effort variance based on ESP with that based on SP. We will start with case study 3 to provide a simple example of our estimation steps. The analysis of the result as follows:

Case Study 3

As we noticed above, this case consists of four Epics. We will apply our four main steps of our framework and calculate the ESP for each one of those Epics. We discuss our steps with an Epic example in details as follows:

For Epic 1: **As a user, I can create and update account information in OGV web application.**

Step 1: Rate the Uncertainty Attributes

The first step is determining the uncertainty in the Epic with five attributes of requirement uncertainty. Here in this epic, the team members accepted on these indices of uncertainty:

- Product vision and strategy clarified = 2
- Key stakeholders involved =2
- Project dependencies identified = 1
- Project business cases identified = 3
- Requirements specifications analyzed = 1

Then, sum all these indices is our Certainty Index (CI) based on Equation (1)

$$\mathrm{CI} = 2 + 2 + 1 + 3 + 1 = 9$$

Step 2: Calculate the Functional Requirement (FR)

The second step is extracting keywords from the Epic in order to know the functional requirements and list it as User Stories.
For the same Epic 1:

Epic 1: As a user, I can create and update account information in OGV web application.
Two key verbs/keywords are extracted from Epic 1:
Create and Update Account information. From these two verbs, we create five functional requirements based on functional decomposition techniques. Then we wrote it as User Stories as follows:

- As a user, I can sign up for a new account in OGV web application.
- As a user, I can login to access the account.
- As a user, I can log out from the account.

- As a user, I can update account information.
- As a user, I can change user password.

Step 3: **Calculate the Basic Factors of Complexity**

The third step is to count the basic factors of the complexity. For each User Story, we count the input, output, and interface. The first User Story in this Epic **"As a user, I can sign up for a new account in OGV web application"**. The user enters his/ her information to create the account; this can be counted as **one input**. The website homepage which appears when a user signs up can be counted as **one output**. The user information will be stored in personal file information and this can be counted as **one interface file**. Then we sum the entire basic factor as Equation (3)

Basic Factors (BF) = 3

Then, compute the Complexity of Functional Requirement as shown in Equation (2) and (4)

FR = 1 and BF = 3

(CFR) = 1 * 3 = 3

We repeat these steps until all User Stories of the Epic are finished. **Table 1** shows the Functionality and Complexity factors of the first Epic in case study one.

Step 4: Calculate the Total Enhanced Story Point

Last step we sum the Complexity of Functional Requirement (CFR) with Certainty Index (CI) as our Enhanced Story Point as shown in equation (5)

ESP = 9 + 11 = 20

In order to compare Effort Variance before and after using ESP, we should calculate the estimated effort in case of using ESP. This can be calculated as follows:

The Story Points (SP) of Epic 1 is 13 and the estimated effort is 163 hours. From this, we can calculate the Estimated Effort (ESP) as:

$$Estimated\ Effort\ (ESP) = \frac{ESP * Estimated\ Effort\ (SP)}{Story\ Points} \quad (8)$$

Estimated Effort (ESP) = (20*163) /13 = 251 hours

We calculated the Effort Variance (EV) and Magnitude of Relative Error (MRE) for ESP to evaluate our result. The result shows that for an Epic 1 when we estimate effort based on ESP, the Effort Variance decreased from 0.61 (61%) to 0.04 (4%). In addition, the MRE also decreased from 0.37 to 0.04. This means that the estimation based on our proposed ESP is closer to the actual which shows that the factors we used has affected positively the estimation value and the result is better than estimation based on Story Point. Figure 2 shows the estimated and actual effort of SP and ESP of the four Epic in case study 3.

M R E is high in Epic 2 compared with Epic 1 that is because in Epic 2 when team members discussed the uncertainty attributes, they assigned low values compared with other Epics; more tasks with low uncertainty on this Epic make our ESP and Estimated Effort (ESP) far from the actual effort. To avoid this problem in the future, we may introduce confidence factor for the estimate. However, the result when we use ESP is better than when using SP. In Epic 3, the result shows that the effort variance also decreased from 0.60 (60%) to 0.16 (16%) and MRE in this Epic is 0.14, which shows that using ESP is better than using SP.

About Epic 4, Estimated Effort (ESP) is more than the actual effort and the effort variance was a negative value, this because the value of the uncertainty here is higher than other Epics. In the early stage of this project i.e. in the requirement stage, the team was not having enough knowledge about alert management and what it should do. The basic information they had is just that we can add a simple alert to the system. When we compare the estimated

Table 1. Basic factors of the complexity for Epic 1

Epic	User stories	Input	Output	Interface	CFR
Epic 1	As a user, I can sign up for a new account in OGV web application	1	1	1	1(1+1+1)=3
	As a user, I can login to access the account	1	1	-	1(1+1+0)= 2
	As a user, I can log out from the account	1	1	-	1(1+1+0)= 2
	As a user, I can update account information	1	-	1	1(1+0+1)=2
	As a user, I can change user password	1	-	1	1(1+0+1)= 2
Total					**11**

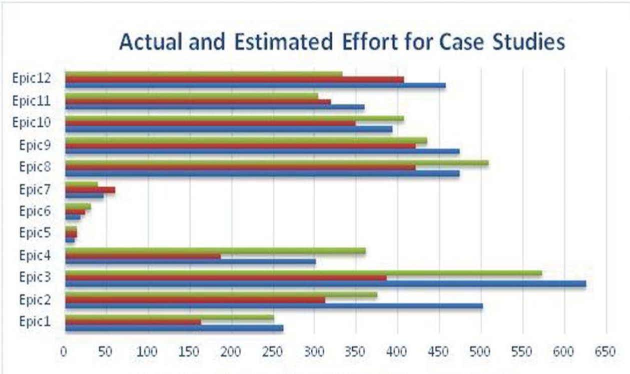

Figure 2. Actual effort and estimated effort for the case studies.

effort based on ESP with the estimated effort based on SP, we note that the ESP is better than SP since the MRE was decreased from 0.36 to 0.15 for SP and ESP respectively.

Case Study 1

In this case study, we have three Epics with an absolute average effort variance 21%. After we applied our steps to get ESP, the effort variance was varying from Epic to another. The result shows that for an Epic 5 when we estimate by ESP, the effort variance decreased from -0.21 (-21%) to -0.05 (-5%), which seem to be better than estimated by SP. In addition, the MRE also decreased from 0.25 to 0.08. The MRE is high in Epic 6 compared with others Epics that is because in Epic 6 we counted the number of interfaces based on this rule "each functional requirement or User Story has one interface". (It is worth noting that the estimate could have been closer to the actual if this functional requirement was counted as one interface for all of the six User Stories. This shows that our framework could not be automated and will depend on a specialist to count the basic factors of complexity. For an Epic 7, the result when using ESP is somewhat better than estimated by SP since the MRE is decreased from 0.28 to 0.21 for SP and ESP respectively as shown in Figure 2.

Case Study 2

After we applied our framework and calculated the ESP, the result was as follows: for the first Epic (Epic 8), our estimated points (ESP) are greater than SP, that's because the basic function of each game means more actions i.e. start, save, restore and others that the user clicks it. All these actions can be counted as inputs and at the other hand, the action that results from each click is counted as an output. Thus, the estimated effort in case of using ESP is more than the actual effort. However, in this Epic, our (MRE) has decreased from 0.13 to 0.07. For Epic 9, when we estimated using ESP, the Estimated Effort (ESP) effort is greater than existing Estimated Effort (SP) "i.e., when estimated by SP" and close to the actual effort. Thus, the effort variance is decreased from 0.13 (13%) to 0.09 (9%) and in addition, the MRE has decreased from 0.11 to 0.08.

Epic 10 is related to the appearance of the game, only two inputs that the user can do: the first input is when the user selects background music and the second input is when choosing the background board' style. The rest of functional requirements can be counted as outputs or interface files. However, according to the first discussion between the team members, this Epic has lots of uncertainty. For thus, our ESP hers is more than SP and the Estimated Effort (ESP) closer to the actual effort.

Finally, for Epic 11, and Epic 12, when we estimated using ESP, the effort variance and the MRE are higher than when estimated by SP especially in Epic 12. The reason for that is that in our framework, we do not count processes i.e., for these Epics (Epic 11 and Epic 12), one action can be done by the user, which is selecting the level of the strength and the output is the game board with the selected level. However, the process is more difficult because there are many possible ways to win. Moreover, many spaces on the board take more effort and time from the programmers and testers to do it. **Figure 2** displays the estimated and actual effort of SP and ESP of the five Epics in case study 2.

The Effort Variance (EV) and (MRE) for the three case studies with twelve Epics is displayed in Table 2.

We compare the results of the three case studies by calculating Average Effort Variance when estimated by SP (Avg_EV(SP)) and when estimated by ESP (Avg_EV(ESP)) for each case study as shown in Table 3. We concluded that:

The enhancement for Case Study 3 (Low-Experience) is the highest one, where the Avg_EV (SP) was 0.61 (61%) vs Avg_EV (ESP) was 0.09 (9%). Our view point of the reason of that is summarized as follows: Case Study 3 is a graduation project so the team members did not have previous experience in agile approach; they assigned their estimated points based on the tasks they are familiar with for each Epic, not on complexity. They did not follow any defined rules to estimate. As the result of that, their Avg_EV(SP) is high (61%). When we applied our proposed framework and discussed with the team members our rules and steps, especially the uncertainty attributes and how can be effective or may

Table 2. Effort variance and MRE for both SP and ESP for each Epic

Case Study Id	Epic Id	Using Story Points (SP)			Using Enhanced Story Points (ESP)		
		SP	EV_SP	MRE_SP	ESP	EV_ESP	MRE_ESP
Case study 3	Epic 1	13	0.61	0.38	20	0.04	0.04
	Epic 2	25	0.60	0.37	30	0.33	0.25
	Epic 3	31	0.62	0.39	44	0.14	0.12
	Epic 4	21	0.61	0.41	28	-0.13	0.15
Case study 1	Epic 5	16	-0.20	0.25	14	-0.07	0.08
	Epic 6	24	-0.21	0.26	31	-0.34	0.47
	Epic 7	64	-0.22	0.28	39	0.27	0.21
Case study 2	Epic 8	29	0.13	0.11	35	-0.07	0.07
	Epic 9	29	0.13	0.11	30	0.09	0.08
	Epic 10	24	0.12	0.11	28	-0.03	0.04
	Epic 11	22	0.13	0.11	21	0.18	0.15
	Epic 12	28	0.13	0.11	20	0.37	0.27

have a positive effect on the estimated value, the result was improved and the estimated by ESP was better than that estimated by SP. The results show that estimation based on ESP is better with an Avg_EV(ESP) is 9%. In addition, the MRE is decreased from 0.36 to 0.09.

In the Case Study 1 (Medium-Experience), results when we applied ESP are better than their estimation using SP; the Average Variance was decreased from -21% to -5%. This is because the team has good experience with agile projects and SP. Therefore, the enhancement that our framework introduced is not as high as the case study 3.

Finally, for Case Study 2 (High-Experience), our enhancement is low, compared with Case Studies 1 and 3. This case study is a reference example introduced by Cohn, which discussed and explained how to estimate by SP in details. Team members have a high experience to define all the rules of estimating User Stories with SP. In addition, an agile coach joined their meetings and supported the team with their SP estimation and User Story understandings. Discussions between the team members and the agile coach increased the success of the whole agile project management. All those reasons made the estimated points (SP) much closer to the actuals. The Average Effort Variance in their estimation points (Avg_EV(SP)) was 0.13 (13%) while the Average Effort Variance of our estimated ESP (Avg_EV(ESP)) is 0.11 (11%) as shown in **Table 3.** **Figure 3** shows the average effort variance for each case study.

Table3. Average effort variance for the three case studies

Case Study_id	Avg_EV(SP)	Avg_EV(ESP)	Enhancement
1	-0.21 (-21%)	-0.05 (-5%)	0.16 (16%)
2	0.13 (13%)	0.11 (11%)	0.02 (2%)
3	0.61 (61%)	0.09 (9%)	0.52 (52%)

When we calculated our enhancement percentage in each case study, the results seem a positive enhancement as shown in **Figure 4.**

5. Experience Impact

As we mentioned above, estimating size in agile projects is based on the personal experience of the estimator(s), which means that if the estimator(s) has good experience with agile methods and Story Points estimation; there is a higher probability of having a more accurate estimation. Our framework is meant to provide guidance to estimator(s) to ease out the impact of the personal experience. While analyzing the case studies, we have observed that the experience parameter has an impact on the estimation results. Thus, according to our observation, we classified the estimator(s) of each case study into three types as fol-

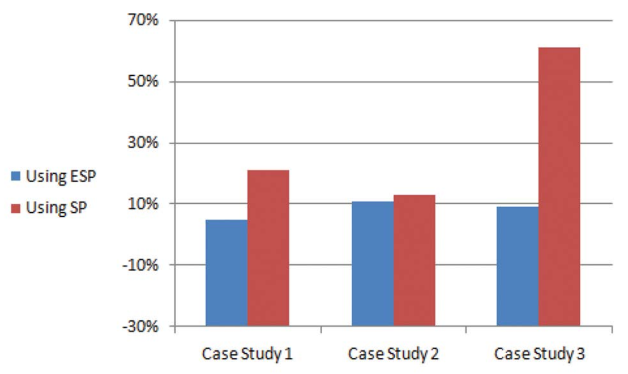

Figure 3. Decreased the average effort variance for each case study when using ESP.

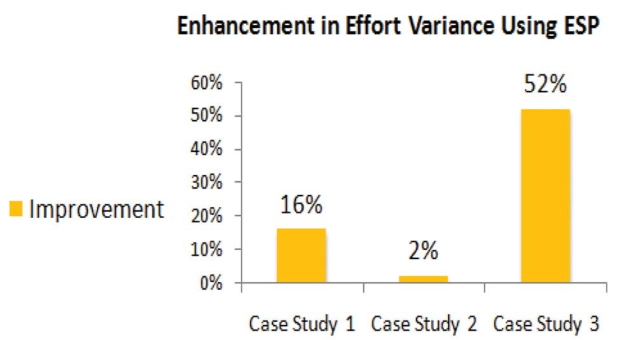

Figure 4. Enhancement in average effort variance using ESP in the three case studies.

lows: The estimator of the first case study (Case Study 3) is within the 'Low-Experience' level, while the estimator of the second one (Case Study 1) is within the 'Medium-Experience' level and the estimator(s) of the third study is within the 'High-Experience' level. **Figure 5** displays the change (increase/decrease) enhancement using ESP according to level of team experience in the estimation phase that achieved by using our proposed framework. Notice that ESP results are higher in case of project teams with Low level of experience.

6. Conclusion and Future Work

It is difficult to estimate the size of the project accurately in the early phase of the project because in the early phase, there is uncertainty about the project scope and the functionalities are still not clear. Moreover, there are inherent requirements changes, which increase the uncertainty of requirements and functionalities.

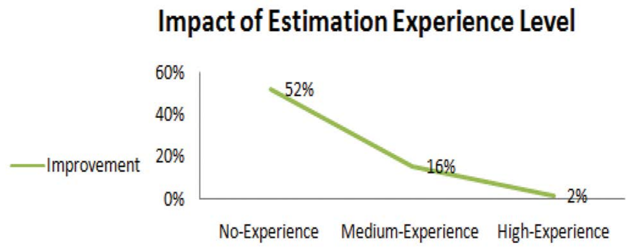

Figure 5. Impact of estimation experience level.

Estimating the size of the agile software projects is based on tame experience. Story Points (SP) is a unit measure of size used to assign a relative value to User Stories. However, if the team does not have the experience to estimate by SP or to deal with agile projects, the estimation may not be accurate.

In this study, we presented Enhanced Story Point (ESP) to enhance size estimation on agile software projects and help the team members to reach an accurate estimation especially when no-experience cases. ESP is based on three factors that may affect estimation accuracy; these factors are Uncertainty, Functionality, and Complexity. We applied the framework on three case studies with 12 Epic and 88 User Story. These three case studies have used an agile process in order to manage their work and assigned SP based on their experience and on the knowledge of the project requirements (User Stories). For each Epic, we calculated the Effort Variance and Magnitude Relative Error (MRE) in case of estimation by Story Points (SP) and the result is varying from case to another according to the level of each estimation team's experience. After that, we applied our framework and calculated the Enhanced Story Point (ESP) as a unit to measure the size of each case study.

We classified these three case studies into three levels: Low-Experience, Medium-Experience, and High-Experience level. The result shows that estimation based on ESP resulted in improvements in case of Low-Experience (Case Study 3) level is higher than Medium-Experience (Case Study 1) and High-Experience (Case Study 2) level, while the improvements in 'High-Experience' level is lowest one. The results are promising, however, more case studies are needed to be studied assessed and evaluated. In a future work, more Epics and case studies will be applied to validate our findings.

7. References

1. Armario J, Gutiérrez JJ, Alba M, García-García J, Vitorio J, Cuaresma MJE. Project Estimation with NDT. 7th International Conference on Software Paradigm Trends; 2012. p. 1801–11.
2. Galal-Edeen GH, Kamel A, Moussa H. Applying an estimation framework in software projects a local experience. The 7th International Conference on Informatics and Systems (INFOS), IEEE; 2010. p. 1-9
3. Tawfik SM, Abd-Elghany MM, Green S. A software cost estimation model based on quality characteristics. The 19th International Conference on Advanced Information Systems Engineering. 2007. p. 1–7.
4. Ebert C, De Man J. Requirements uncertainty: Influencing factors and concrete improvements. Proceedings of the 27th International Conference on Software engineering. 2005. p. 553–60 https://doi.org/10.1145/1062455.1062554
5. Pfleeger SL, Wu F, Lewis R. Software cost estimation and sizing methods: Issues, and guidelines. Rand Corporation; 2005. p. 1–127.
6. Sharma A, Kushwaha DS. Complexity measure based on requirement engineering document and its validation. International Conference on Computer and Communication Technology (ICCCT). IEEE. 2010. p. 608–15. https://doi.org/10.1109/ICCCT.2010.5640472
7. Moussa H, Galal-Edeen G, Kamel A. Enhancing software sizing adjustment factors. Proceedings of the 4th International Conference on Intelligent Computing and Information Systems (ICICIS'09); 2009. p. 438–44.
8. Bjarnason E, Wnuk K, Regnell B. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. Proceedings of the 1st Workshop on Agile Requirements Engineering; 2011. p. 1–7. https://doi.org/10.1145/2068783.2068786
9. Santana C, Leoneo F, Vasconcelos A, Gusmão C. Using function points in agile projects. Agile Processes in Software Engineering and Extreme Programming; 2011. p. 176–91. https://doi.org/10.1007/978-3-642-20677-1_13
10. Beck K. Extreme programming explained: embrace change. 2nd ed. Addison-Wesley Professional; 2004. p. 1–224. PMCid:PMC2780070
11. Cohn M. Agile estimating and planning. Pearson Education; 2005.
12. Buglione L, Abran A, editors. Improving estimations in agile projects: Issues and avenues. Proceedings of the 4th Software Measurement European Forum (SMEF 2007); Rome (Italy). 2007. p. 265–74.
13. Choetkiertikul M, Dam HK, Tran T, Pham T, Ghose A, Menzies T. A deep learning model for estimating story points. IEEE Transactions on Software Engineering. 2016. p. 1–1.
14. Grenning J. Planning poker or how to avoid analysis paralysis while release planning. Hawthorn Woods: Renaissance Software Consulting. 2002; 3:94–8.
15. Haugen NC. An empirical study of using planning poker for user story estimation. IEEE Agile Conference; 2006. p. 9–34. https://doi.org/10.1109/AGILE.2006.16
16. Mahnič V, Hovelja T. On using planning poker for estimating user stories. Journal of Systems and Software. 2012;85(9):2086–95. https://doi.org/10.1016/j.jss.2012.04.005
17. Poženel M, Mahnič V. Studying agile software estimation techniques: The design of an empirical study with students. Global Journal of Engineering Education. 2016; 18(2):1–6.
18. Estimating with Tee Shirt Sizes. Available from: https://www.mountaingoatsoftware.com/blog/estimating-with-tee-shirt-sizes
19. Cohn M. User stories applied: For agile software development. 1st ed. Addison-Wesley Professional. 2004. p. 1–304. PMid:15238064 PMCid:PMC1360149
20. In Defense of Large Numbers. Available from: https://www.mountaingoatsoftware.com/blog/in-defense-of-large-numbers
21. Fibonacci Must Die. Available from: http://blog.daverooney.ca/2011/06/fibonacci-must-die.html
22. Agile Estimation Guidance. Available from: https://www.leadingagile.com/2016/08/agile-estimation-guidance
23. Satapathy SM, Panda A, Rath SK. Story point approach based agile software effort estimation using various svr kernel methods. The 26th International Conference on Software Engineering and Knowledge Engineering (SEKE 2014); Vancouver, Canada. 2014. p. 304–7.
24. Popli R, Chauhan N. A sprint-point based estimation technique in Scrum. IEEE International Conference on Information Systems and Computer Networks (ISCON); 2013. p. 98–103. https://doi.org/10.1109/ICISCON.2013.6524182
25. Coelho E, Basu A. Effort estimation in agile software development using story points. International Journal of Applied Information Systems (IJAIS). 2012; 3(7):1–4. https://doi.org/10.5120/ijais12-450574
26. Georgsson A. Introducing story points and user stories to perform estimations in a software development organisation. A case study at Swedbank IT (Unpublished master's thesis) Umeå University, Umeå, Sweden; 2011. p. 1–52.
27. Fuqua AM. Using function points in XP-considerations. International Conference on Extreme Programming and Agile Processes in Software Engineering; 2003. p. 340–2. https://doi.org/10.1007/3-540-44870-5_46

28. Kang S, Choi O, Baik J. Model-based dynamic cost estimation and tracking method for agile software development. IEEE/ACIS 9th International Conference on Computer and Information Science (ICIS); 2010. p. 743–8. https://doi.org/10.1109/ICIS.2010.126
29. Desharnais J-M, Buglione L, Kocatürk B. Using the COSMIC method to estimate Agile user stories. Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement; 2011. p. 68–73. https://doi.org/10.1145/2181101.2181117
30. Ali M, Shaikh ZA, Ali E. Estimation of project size using user stories; 2016.
31. Armentrout AW, Trujillo RM. Function points, use case points, story points: Observations from a case study. Cross Talk. 2013. p. 1–5.
32. Bhalerao S, Ingle M. Incorporating vital factors in agile estimation through algorithmic method. International Journal on Computational Science and Applications (IJCSA). 2009; 6(1):85–97.
33. Chandrasekaran RLS, Kanchana V. Multi-criteria approach for agile software cost estimation model. Proceedings of to Check; 2007. p. 1–8.
34. Kandpal M, Kandpal A. Critical analysis of traditional size estimation metrics for object oriented programming. International Journal of Computer Applications. 2012; 58(13):1–7. https://doi.org/10.5120/9345-3669
35. Angelis L, Stamelos I, Morisio M. Building a software cost estimation model based on categorical data. IEEE Software Metrics Symposium, Proceedings 7th International Software Metrics Symposium; 2001. p. 4–15.
36. Thangaratinam S, Redman CW. The delphi technique. The obstetrician and gynaecologist. 2005; 7(2):120–5. https://doi.org/10.1576/toag.7.2.120.27071
37. Hagal MA, Sallabi OM. A structured approach for extracting functional requirements from unclear customers. Naif Arab University for Security Sciences; 2011.