2013 Joint Conference of the 23nd International Workshop on Software Measurement (IWSM) and the Eighth International Conference on Software Process and Product Measurement (Mensura)

Improving the User Story Agile Technique Using the INVEST Criteria

Luigi Buglione
Industry & Service Dept.
ETS / Engineering Ingegneria Informatica S.p.A. (IT)
Rome, Italy
luigi.buglione@eng.it

Alain Abran
Dept. of Software Engineering & Information Technologies
Ecole de Technologie Supérieure (ETS)
Montréal, QC, Canada
alain.abran@etsmtl.ca

***Abstract*—Although the Agile Software Development (ADS) approach has been around for the last 15 years, it is only recently that attention has moved towards Agile Software Management (ASM) for tackling some of the management-related weaknesses, such as estimating on the basis of User Story points. This paper presents an application of the INVEST criteria (Independent – Negotiable – Valuable – Estimable – Small – Testable) for improving the measurement technique of User Stories, introducing sizing units and a technique to negotiate requirements. It includes a discussion on an approach to balancing the six criteria used to evaluate a set of User Stories in a Sprint.**

***Keywords*—Estimation; Requirements; INVEST; Agile; User Story; Story Points.**

I. INTRODUCTION

The Agile approach ("'Agile") has become popular over the last 15 years, but, while simple in concept, sometimes applying it remains challenging. This paper investigates a single aspect of Agile, consisting of some of the quantitative techniques of the approach, and their use for benchmarking and estimation purposes.

In the Agile community, the typical estimation approach is based on effort-related concepts such as story points and velocity, among others, derived from requirements documented using the User Story technique. From the measurement-related issues identified in [1], the following stand out: (a) product sizing units; (b) historical data; and (c) requirements. This paper focuses on the third issue, requirements, and proposes a restructuring of the User Story technique through the introduction of the set of criteria compiled on the INVEST Grid [7] (Independent, Negotiable, Valuable, Estimable, Small, and Testable) for evaluating, improving and managing user stories.

The purpose of the focus on estimation is, from the business perspective, to reduce the estimation error as much as possible. This error is often a consequence of incomplete requirements, requirements provided late (*scope creep*), or not having the right level of detail for expressing what is known in the Functional Size Measurement (FSM) community as the *elementary process concept* (the elementary process being the smallest unit of activity that is meaningful to the user). Another purpose is to reduce the subjectivity in story point estimates, which are typically derived and managed in an experiential way (e.g. simplified Delphi) without reference to the organization's historical data for comparison.

II. THE THREE MAIN ISSUES – BACKGROUND

This section clarifies three User Story issues: sizing units, historical data, and requirements. As discussed in [1], these are three of the main issues, identified through root-cause analysis, which, if solved, could improve estimates by diminishing the number of estimation errors by reducing variability.

A. Sizing Units

The first issue is related to the lack of an explicit measurement unit for sizing Agile requirements. Story points (SP), as currently structured, represent the effort estimated by the Agile team that is required to develop a User Story (US). Since they are established locally, they are directly impacted by local conditions (related to the team, the technology, and the context, for example). To summarize, the SP technique estimates effort, but does not size the software itself using measurement rules that are independent of the development context.

The classic cause and effect chain in production-development processes has explicit, and distinct, quantities, each with their own distinct measurement units: the quantity (Q) of objects and services to be produced influences the effort (E) required to produce this quantity of output, which in turn influences the costs: Q → T → C (Quantity → Time (Effort, Duration) → Costs) [8][1]. In estimation models, the quantity is the input to the estimation models, and the Effort and Costs are the output variables: the inputs have their own size units, the output has a different size unit (effort, in this case), while it is the estimation function that translates the input units into the output units. In the SP approach, there is no such direct and explicit measurement of the inputs with their corresponding size units. SP skip this explicit measurement of the inputs to jump directly to the estimation of its outputs in terms of effort units. This process lacks transparency and objectivity. For example, for effort, there is no objective traceability to independent sizing of the estimates. The SP approach is a typical example of an over-simplification in measurement and software estimation: the transfer function from the inputs to the

[1] See Figure 1 for the way to split user requirements into FUR, NFR, and other requirements.

978-0-7695-5078-7/13 $26.00 © 2013 IEEE
DOI 10.1109/IWSM-Mensura.2013.18

outputs is hidden and not described; the inputs themselves are not objectively measured; and the estimation is based on judgment calls without traceability to objectively quantified inputs using international standards, such as Function Points for sizing the Functional User Requirements (FUR).

B. Historical Data

The absence of explicit and objectively measured sizing units means that adequate comparisons across software functions of various sizes cannot be made and prevents the design of well-structured estimation models. Without adequate sizing units, it is impossible to leverage, for benchmarking or estimation purposes, the content of international repositories based on ISO-recognized functional sizing measurement methods. Currently, the best known publicly available repositories of software projects are the ISBSG repository (www.isbsg.org) and the PROMISE repository of software projects (http://promisedata.org).

C. Requirements

Last, but not least, Functional User Requirements (FUR) represent the ISO's mandatory base concept for establishing a software sizing unit at the functional level – see ISO/IEC 14143-1 [16]. However, FUR are only one side of the story. Most of the time, they are complemented by NFR (Non-Functional Requirements)[2], which have many different facets [2], referred to in ISO language as "technical and quality requirements" (e.g. ISO 25010:2011, and its predecessor ISO 9126-1:2001). A typical US, for instance, describes an FUR while its quantification into effort implicitly includes the related effort for its corresponding set of undocumented NFR [5].

Another over simplification is, from our viewpoint, that a requirement is not typically split into sub requirements, but managed as a unit. There is most often an NFR side to an FUR. In fact, it is often possible to further decompose NFR into FUR to be allocated to other software in the same project [10][13][14][3]. In addition, core NFR are not decomposable into FUR. This underscores the usefulness of clear definitions which are shared definitions by the two contracting parties (customers and developers), in order to avoid misconceptions in the early phases of the project life cycle (i.e. the requirements elicitation phase) which might lead to litigation. This issue will be treated in a later paper. The scope of this work is limited to finding a more practical way to deal with US, with a view to improving the quality of requirements right from the start of a project.

III. THE INVEST GRID

In this section, we introduce the use of the INVEST Grid in Agile requirement management. A requirement in an Agile context should be: (a) Independent; (b) Negotiable; (c) Valuable; (d) Estimable; (e) Small; and (f) Testable. These criteria are represented by the acronym INVEST. Ideally, until the two parties (customer and provider) have agreed that the INVEST criteria have been met in a US, that US should not be transferred to a Sprint (iteration in the Scrum language). This is because, at this point, the six INVEST criteria constitute a qualitative evaluation target. For instance, each US must be analyzed to determine if it is sufficiently independent of the other US to be moved to another Sprint without impacting software deployment. If the client has a high enough level of authority to negotiate a change in a US before that US becomes a full requirement, the provider should allow the change, but without falling into the scope creep problem, or running into a time constraint with respect to the agreed schedule for modifying US, etc.[4].

The question is, to what extent should the criteria be met? How independent, or small, or testable, etc. should a US be? The first improvement to make would be to introduce a threshold (or tolerance level) for each of the six criteria. For instance, a US could be considered sufficiently independent if it does not need to be split into more sub requirements. This improvement would help to reduce subjectivity in the use of this set of evaluation criteria.

An example of the use of the set of INVEST criteria was developed in preliminary work for evaluating a set of measures and KPI in a BPM (Business Process Model) for an organization [11], This grid uses a rating scale of 0 to 3, based on Cohn's definition in a US context (e.g. "Small" means that a US should be sufficiently granular, but not defined at too high a level. For instance:

- If a US is rated 0 based on the Small criterion, it means that it is very large and cannot be completed in a Sprint;
- If a US is rated 1 based on the Small criterion, it means that the customer and the provider agreed during the analysis that it can be completed in a Sprint, along with other US, but it is too small to create an overhead based on the Testability criterion.

The INVEST grid in Appendix A can be used as a template when the customer and provider meet to evaluate a US.

IV. THE INVEST PROCESS

The set of concepts and activities described in the previous sections is represented as an activity flow in Figure 1, and is referred to here as the INVEST process.

Before entering into a session for evaluating a set of US for new Sprints, the customer and provider can, from their experience, set acceptable thresholds, for each of the six criteria. Taken together, these thresholds can be referred to as a *profile*. If some of the thresholds are not met by the end of the evaluation session for one or more US (and so do not coincide with the established profiles – step #5), the distance remaining, in terms of effort, will constitute the improvement axis on which to start work to determine what should be done and how much effort may be needed to meet the thresholds, and so continue to the next evaluation stage (returning to step #1 or #2, depending on what needs to be modified and who should be

[2] Browse SEVOCAB [4] to access updated ISO definitions.

[3] These last two references focus on the opportunity to provide a broader view of systems, and not strictly of software entities, and also combine the software and the hardware views.

[4] See [6] for the full explanation about how to use the 'basic' INVEST criteria.

responsible for the modifications, as indicated in the process flow).

Working with the grid requires that a single US be viewed in a comprehensive way, as part of a set of requirements to be analyzed and allocated, from a project management perspective, to a Sprint iteration. For example, if a US is rated 0-1 for the Small criterion (and so not expressed at the elementary process level, according to the definition proposed in Appendix A), it may need to be split into two or more sub requirements. In contrast, if a US is rated 3 for the Independent criterion, it can be transferred to a Sprint on its own, with no overlapping issues with other project management or planning, tasks.

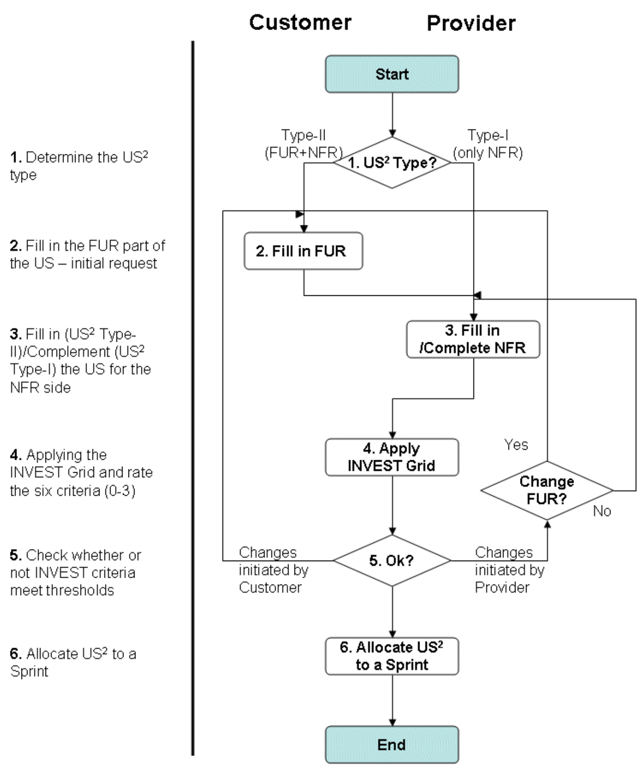

Fig. 1. The INVEST process flow

Of course, Appendix A is an instantiation of this INVEST approach and not a way to analyze a US exhaustively. An Agile team can take the lessons learned, and customize the definitions and the way to split the definitions of maturity (0, 1, 2, and 3), as preferred.

V. Some Lessons from Experience

The preliminary application of the INVEST grid in an industrial context has led to some observations for planning improvement actions in organizations. The lessons learned include the following:

- Definitions – In their first INVEST sessions, the customer and the provider may need to realign the way they define the working items and the NFR activities linked to the more 'visible' FUR. This may primarily impact the Small criterion.
- Level of granularity – In FSM, the verb "Manage" is typically split into 4-5 elementary processes (the so-called CRUD or CRUDL processes – Create/Read/Update/Delete/List). The risk here is the inconsistent application of the US granularity level, which has a direct impact on the Small criterion and indirectly affects the Independent criterion, in terms of a better allocation of stories to Sprints, and the Testable criterion, as a result of the test coverage.
- Estimation by experience/analogy – Most practitioners in the Agile context tend not use any structured measure (such as functional size measures), but only to determine the time (effort/duration) as the final value for scheduling a US. In addition, because NFR-related activities are 'hidden', SPs do not allow direct comparison, and a sizing measure has been recommended instead.
- Limited attention to proper test coverage against requirements – This is one of the issues that practitioners deal with less often, because little residual effort and budget typically remain at that moment of the project life cycle. The Testable criterion stresses the need to reduce the Cost of Non Quality (CONQ) from the start, instead of focusing only of the more visible costs that may be incurred prior to the first deployment.
- APM (Agile Project Management) concepts –, The application of these concepts, for example the Burndown chart and regular velocity tracking during Burndown hours, can be leveraged by the introduction of one or more sizing measures. The INVEST Grid can help organizations rethink the Requirements Elicitation phase, which is often under evaluated, even though it is crucial for determining the right quantities to produce and the production schedule of a project.

VI. Discussion & Future Work

Agile methods use the immature and poorly structured User Story technique to manage a project. Improvements to functional size measures can help improve both requirements management and the sizing process. Our proposed improvement is the INVEST Grid, designed for running a mix of qualitative and quantitative planning games for each story to be discussed and, following acceptance, to be assigned to the appropriate Sprint/iteration.

Further work is required to refine and tailor definitions across the four levels for the six INVEST criteria, and to obtain empirical feedback from various industry domains.

References

[1] Buglione L. & Abran A., Improving Estimations in Agile Projects: Issues and avenues, 4th Software Measurement European Forum (SMEF 2007), Rome (Italy), May 9-11, 2007, ISBN 9-788870-909425, pp. 265-274, URL: http://goo.gl/Y25tz

[2] Daneva M., Buglione L., Hermann A., Software Architects' Experiences of Quality Requirements: What we know and what we don't know, REFSQ'13, 19th Int. Working Conference on Requirements Engineering, Essen (Germany), April 8-11, 2013, pp. 1-17, URL: http://goo.gl/LbBlE

[3] ISO/IEC, IS 14143-1:2007, Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts.

[4] SEVOCAB, Software/System Engineering Vocabulary, IEEE/ISO/PMI, URL: www.computer.org/sevocab

[5] Buglione L. Agile-4-FSM. Improving estimates by [means of] a 4 piece puzzle, Webinar, IFPUG Agile Interest Group, May 17, 2012, URL: http://goo.gl/wtXWt

[6] Cohn M. User Stories Applied: In Agile Software Development, Chapter 2: Writing Stories, Addison Wesley, 2004, URL: http://goo.gl/zN2jg

[7] Buglione L. Meglio Agili o Veloci? Alcune riflessioni sulle stime nei progetti XP, XPM.it, Febbraio 2007, URL: www.xpm.it

[8] Buglione L. The Next Frontier: Measuring and Evaluating Non-Functional Productivity, Metric Views, IFPUG Newsletter, Vol. 6, Issue No. 2, August 2012, pp. 11-14, URL: http://goo.gl/yMgdC

[9] DSDM Consortium, *DSDM Public Version 4.2*, 2007, URL: http://www.dsdm.org

[10] M. Kassab, O. Ormandjieva, M. Daneva, A. Abran. Non-Functional Requirements Size Measurement Method (NFSM) with COSMIC-FFP. Lecture Notes in Computer Science (LNCS) 4895 – 0168, Springer-Verlag Berlin Heidelberg, pp. 168-182, 2008.

[11] Buglione L., Abran A. Assessment of Measurement Indicators in SPI Frameworks, in "Investigations in Software Measurement," 13th International Workshop on Software Measurement (IWSM2003), September 23-25, 2003, Montréal (Canada), Shaker Verlag, ISBN 3-8322-1880-7, pp. 287-309.

[12] Crosby P. Quality is Free, McGraw-Hill, 1979, ISBN 0-07-014512-1.

[13] Khalid T. Al-Sarayreh, Alain Abran, Juan J. Cuadrado-Gallego, “A Standards-based Model of System Maintainability Requirements,” Journal of Software: Evolution and Process, Wiley Online Library – Wiley & Sons, March 2012, Online ISSN: 2047-7481 DOI: 10.1002/smr.1553, Print version: May 2013.

[14] Abran A, Al-Sarayreh, K.T., Cuadrado-Gallego J. J. “A Standards-based Reference Framework for System Portability Requirements”, Computer Standards and Interfaces Journal, Elsevier, Vol. 35, 2013, pp. 380-395. http://dx.doi.org/10.1016/j.csi.2012.11.003

[15] Berardi E., Buglione L., Santillo L., Symons C., Trudel S. Guideline for the use of COSMIC FSM to manage Agile projects, v1.0, COSMIC, Technical Report, September 2011, URL: http://goo.gl/XWqBO

[16] ISO/IEC IS 14143-1:2007, Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts, International Organization for Standardization, 2007

Annex A: INVEST Grid

INVEST	Description	0 *Poor /Absent*	1 **Fair**	2 *Good*	3 *Excellent*
I – Independent	*User Stories should be as independent as possible*	The start of construction of a US is tied to the completion of at least one other US	The completion of a US hinders the start of construction of at least one other US	The US can contain any constraint, but its release can be constrained by the completion of at least one other US	The US is fully independent, and it can be realized and released with any constraint
N – Negotiable	*User Stories should be "open", reporting any relevant details as much as possible*	The US contains enough detail to be a technical specification (Design phase), leaving no room to negotiate any element	The US is written with enough detail to be a functional specification (Analysis phase), leaving no room to negotiate any element	The US is written with informative content defining a User Requirement in a consolidated manner, yet shared between Customer and Provider	The US is written with the informative content typical of a high-level need, allowing feedback between customer and provider
V – Valuable	*User Stories should provide value to end users in terms of the solution*	The functional part (F) of the US does not contain all the functionalities requested by the customer	The functional (F) part of the US expresses mostly qualitative (Q) and technical (T) requirements about the system, and needs to be more developed in terms of functional requirements	The functional (F) part of the US expresses mostly the functional requirements requested by the Customer, but also includes qualitative (Q) and technical (T) requirements	The functional (F) part of the US correctly expresses only the functional requirements requested by the customer
E – Estimable	*Each User Story must be able to be estimated in terms of relative size and effort*	The US shows only its functional (F) part, filled in by the customer, but without sufficient detail to allow the provider to fill in the Q/T parts	The US shows only its functional (F) part, filled in by the customer, but validated with the provider	The US has been completed by the provider with respect to Q/T issues, but still needs to be validated jointly with the customer	All the useful parts of the US (F/Q/T) are shown, allowing the effort need to size and estimate it, and validated by both parts
S – Small	*Each User Story should be sufficiently granular, and not defined at too high a level*	The US is very large, and cannot be completed within a Sprint	The US is very large, and can be completed within a Sprint, but cannot accommodate the creation/delivery of other US	The size of the US is such that it can be completed within a Sprint, jointly with other US, but it is too small to create overhead about the Testing phase	The size of the US is such that it can be completed within a Sprint, jointly with other US, ensuring an appropriate balance between development and testing activities
T – Testable	*Each User Story must be formulated in an effort to stress useful details for creating tests*	The US does not include tips about Acceptance Tests	The US includes a formal indication of Acceptance Tests, but yet to be completed	The US includes an indication of Acceptance Tests which are complete, but yet to be validated	The US includes an indication of completed and validated Acceptance Tests