

**This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.**

**Author(s):** Halme, Erika; Vakkuri, Ville; Kultanen, Joni; Jantunen, Marianna; Kemell, Kai-Kristian; Rousi, Rebekah; Abrahamsson, Pekka

**Title:** How to Write Ethical User Stories? : Impacts of the ECCOLA Method

**Year:** 2021

**Version:** Published version

**Copyright:** © The Author(s) 2021

**Rights:** CC BY 4.0

**Rights url:** <https://creativecommons.org/licenses/by/4.0/>

**Please cite the original version:**

Halme, E., Vakkuri, V., Kultanen, J., Jantunen, M., Kemell, K.-K., Rousi, R., & Abrahamsson, P. (2021). How to Write Ethical User Stories? : Impacts of the ECCOLA Method. In P. Gregory, C. Lassenius, X. Wang, & P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming : 22nd International Conference on Agile Software Development, XP 2021, Virtual Event, June 14–18, 2021, Proceedings* (pp. 36-52). Springer. *Lecture Notes in Business Information Processing*, 419. [https://doi.org/10.1007/978-3-030-78098-2\\_3](https://doi.org/10.1007/978-3-030-78098-2_3)



# How to Write Ethical User Stories? Impacts of the ECCOLA Method

Erika Halme<sup>(✉)</sup> , Ville Vakkuri , Joni Kultanen , Marianna Jantunen ,  
Kai-Kristian Kemell , Rebekah Rousi , and Pekka Abrahamsson 

University of Jyväskylä, PL35 Jyväskylä, Finland

{erika.a.halme, ville.vakkuri, joni.m.kultanen, marianna.s.p.jantunen,  
kai-kristian.o.kemell, rebekah.rous, pekka.abrahamsson}@jyu.fi

**Abstract.** Artificial Intelligence (AI) systems are increasing in significance within software services. Unfortunately, these systems are not flawless. Their faults, failures and other systemic issues have emphasized the urgency for consideration of ethical standards and practices in AI engineering. Despite the growing number of studies in AI ethics, comparatively little attention has been placed on how ethical issues can be mitigated in software engineering (SE) practice. Currently understanding is lacking regarding the provision of useful tools that can help companies transform high-level ethical guidelines for AI ethics into the actual workflow of developers. In this paper, we explore the idea of using user stories to transform abstract ethical requirements into tangible outcomes in Agile software development. We tested this idea by studying master's level student projects (15 teams) developing web applications for a real industrial client over the course of five iterations. These projects resulted in 250+ user stories that were analyzed for the purposes of this paper. The teams were divided into two groups: half of the teams worked using the ECCOLA method for AI ethics in SE, while the other half, a control group, was used to compare the effectiveness of ECCOLA. Both teams were tasked with writing user stories to formulate customer needs into system requirements. Based on the data, we discuss the effectiveness of ECCOLA, and Primary Empirical Contributions (PECs) from formulating ethical user stories in Agile development.

**Keywords:** User story · Agile development · Ethics · Artificial Intelligence

## 1 Introduction

During recent years, the role of ethics has been emphasized in the context of Artificial Intelligence (AI) and Autonomous Systems (AS). In the field of Software Engineering (SE) however, few tools or methods are available for systematically incorporating ethics into development. Furthermore, AI ethics has seldom been studied from the perspective of practical application in SE. Ethically aligned AI/AS development principles and guidelines exist [1], yet as recent research demonstrates [2], there are still major challenges in translating these to practice.

© The Author(s) 2021

P. Gregory et al. (Eds.): XP 2021, LNBIP 419, pp. 36–52, 2021.

[https://doi.org/10.1007/978-3-030-78098-2\\_3](https://doi.org/10.1007/978-3-030-78098-2_3)

Overall, AI ethics currently seems to be an area with a prominent gap between research and practice [2]. While we now have some degree of consensus on what AI ethics is and what ethical principles and issues are important to consider in AI development [1], translating these principles into concrete action is challenging [2,3]. Organizations and developers seem to struggle with turning ethical guidelines into tangible requirements.

We have attempted to tackle this issue by proposing a method for implementing AI ethics in SE. The method is called ECCOLA.

The ECCOLA method has been iteratively developed and validated. This current paper reports on one of these iterative validations. Additionally, we wish to better understand how AI ethics should be practically applied to design and development. Another goal of the paper in the context of ethics, is to examine *user stories* and further knowledge of how to write ethical user stories in terms of translating ethical principles into tangible engineering requirements. ECCOLA will be discussed in greater detail in the next section.

Writing user stories is a practice commonly used to help define requirements during development, especially in Agile software development. Thus, we felt that ethical user stories could be one way of making (AI) ethics a part of the workflow of developers. To study user stories in the context of (AI) ethics, we conducted an empirical study of 15 projects. These projects were split into two, with half of the project groups using the ECCOLA method to guide the user story writing process, and the other, the control group, writing user stories without ECCOLA yet with another set of non-ethically oriented cards (“placebos”). The main research question of the current study is: “How can Non-Functional ethically-oriented User Stories be written with the assistance of the ECCOLA method?”

## 2 Background

### 2.1 Implementing Ethics into Software Development

Research seems to point to both challenges and benefits in applying ethics within Agile methods. Miller and Larson [4], on human values in Agile software development, highlighted the importance of developers acquiring an awareness of and skill in performing ethical analysis. This was in order to be able to evaluate development methods on a more sophisticated level. Yet, developers may experience difficulties in articulating ideas about human values, due to their technical language orientation [4]. While comparing Agile Principles with software ethics, Judy [5] concluded that the conversation of ethical dilemmas is largely absent from the Agile context. Particularly in instances where ethical issues do not directly affect business value or teams. Miller and Larson [4] call for tools of ethical analysis; they propose that parties involved in software development need intellectual skills and a vocabulary that will help them understand and communicate competing human duties, values and consequences.

Agile practices are “designed to navigate essential complexity” [5]. Their growing rate of adoption is based on an inclination towards harnessing values and culture in development processes and practices [5]. At the same time, Miller

and Larson [4] propose that through deontological analysis, the Agile Manifesto itself can be seen to place emphasis on human values. According to Judy [5], the Agile community serves as a “vital resource” for peers with shared values.

It would seem that ethical building blocks exist in the Agile methodology itself, but applying ethical analysis tools could further improve the situation through clarifying ethical targets and what they mean in action, even in the absence of “standard” methods.

To address the unique challenges posed by information technology (IT), concepts such as Information Ethics, and further, Computer Ethics, have emerged. The discussion around guidelines and codes of conduct for ethical considerations, as well as initiatives to promote ethical software development, progress as technology evolves. For example, the *ACM Code of Ethics and Professional Conduct* for ethical software development dates back to 1992. It was subsequently updated to better suit the advancement of technology in 2018<sup>1</sup>. This ACM Code of Ethics, as an example of an acknowledged resource of computer ethics, presents principles of responsibility for all who “use computing technology in an impactful way”. It considers ethical principles such as prioritizing human well-being, trustworthiness, fairness and privacy.

The ethical principles of computer ethics proceeded into the evolving discussion of autonomous, intelligent technologies. Debates and discussion regarding AI ethics has produced a widely recognized understanding of AI ethics guidelines, that consist of partially the same principles as those in computer ethics. For example, a study of the guidelines [1] identified a “global convergence emerging around five ethical principles”, namely: transparency, justice and fairness, non-maleficence, responsibility and privacy.

When discussing ethics in IT, Value-Sensitive Design (VSD) is also worthy to mention. Having emerged from the Human-Computer Interaction (HCI) community in the 1990s, it is “a theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner throughout the design process” [6]. It has been utilized in various domains and tools including a ToolKit (for envisioning practices), consisting of 32 cards for envisioning the use case scenario themed in stakeholders, time, values, and pervasiveness [7].

While codes for ethical conduct in SE exist, an issue across domains of software development is in that these codes do not carry over into practice. As suggested by [8], any number of guidelines, policies, and procedures to encourage ethical behavior cannot guarantee their implementation. They state that, “credible results and a strong discipline of empirical software engineering are based on mutual trust that everyone will behave ethically” [8]. However, this trust has not proven to be sufficient in facilitating ethical thinking. For example, McNamara et al. [9] replicated a prior behavior ethics study, and found out that explicitly instructing participants to consider the ACM Code of Ethics in relation to the impacts of their software development decision-making had no influence on actual ethical decision-making itself. In the field of AI ethics, [2]

<sup>1</sup> <https://www.acm.org/articles/bulletins/2018/july/new-code-of-ethics-released>.

discovered a gap between research and practice regarding the ways in which AI ethics are implemented. While on the one hand, AI ethics are discussed in academic circles, on the other hand, discussions had not carried over to industrial application.

## 2.2 ECCOLA Method and It's Application

Inspired by the challenges of implementing ethics in AI development, the ECCOLA method [10] used in the current study, was developed with the intention to provide developers with “an actionable tool for implementing [AI] ethics”. The method considers topics of AI ethics created in reflection of AI ethics principles from relevant literature while aiming to make them more practical and applicable for development. The ECCOLA method is a deck of 21 cards, with eight (8) themes and one to six (1–6) topics in each theme (see Table 1). Developers can utilize the ECCOLA cards to implement the various ethical consideration prompts in software development by using the questions provided on the cards. Each card consists of one topic like the theme transparency considers topics under Communication and Explainability, while Accountability considers topics such as Auditability and Ability to Redress. One additional card, called the Game sheet, explains how the method is used in practice. The cards are split into three sections to motivate what to do while providing a practical example. The cards also contain a note-making space to make it even more practical in real life development work.

**Table 1.** ECCOLA card themes

Card themes (8)	Card number (0–20)	Card amount (total 21)
Analyze	#0	1
Transparency	#1–6	6
Safety & Security	#7–9	3
Fairness	#10–11	2
Data	#12–13	2
Agency & Oversight	#14–15	2
Wellbeing	#16–17	2
Accountability	#18–20	3

ECCOLA is a modular, sprint-by-sprint process, where relevant cards are chosen in advance in order to make the method manageable and focused in the proceeding development work. This process results in a paper trail of ethical choices to be made during the development of software product.

In short, the three (3) phases of prepare, review and evaluate are repeated in every iteration during the development process. Decisions and card selection processes become easier and more productive when developers/users familiarize

themselves with the card themes and contents. The cards are to be sorted into three (3) piles before development. The first pile is for the planning stages of the project. The second one is for different parts of the development and the third pile, if needed during the project’s final phase. The project or product defines what cards are selected and utilized at different development stages. Tutorial sessions are held before the interested parties start to deploy the method. The sessions contain some exercises and an introduction to the method and AI ethics, if needed. In this sense, ECCOLA is, in Agile methodology, a continuum for ethical building blocks in the form of an analysis tool and this we will elaborate upon more in the coming paragraphs.

### 2.3 User Stories in Ethically Aligned Software Design

User stories in the field of SE and Agile software development connect the two sides of software project parties - business and development - in relation to information about customer requirements [11]. User stories are highly apt for Agile environments (as originated from the XP method) due to the fact that they can be utilized for planning iterations and within iterative development processes [11]. From the outset, Agile practices “focus on the development and delivery of only those features that are really useful to customer” [12]. These methods are applied in development projects with fast moving targets, where development teams and applied tools should adapt easily to changes [13]. As the name suggests, this provides software projects with manageable agility, particularly in terms of bringing value to customer needs. This value delivery is enabled through requirements engineering (RE) practices such as user stories [12].

User stories serve as mediators or boundary objects between users and the development team. In the user story process, the decision-making and idea of the software outcome is spread along the development project duration [11]. This simple yet unifying function offers the development team an effective tool to handle information just-in-time. In practice, user stories are handwritten cards or paper notes generated by the customer team. If the customer is not involved in the process the product owner - part of the development team - answers for the customer software requirement needs.

The user story card or template generally contains two sections that describe the requirements at a high level. This is formulated into three leading sentences: “As a <role>, I can <action>, so that <goal>.” This progresses with acceptance criteria that are utilized to evaluate the user story execution [14]. Based on Cohn’s [14] original developments, Dimitrijevic et al. [12] capsulize the user story process into seven steps: user stories gathering, user role modeling, acceptance testing, estimating and planning releases and iterations, as well as tracking and communicating. These seven areas of user story processing emphasize the unpretentious nature of what the process components should entail.

The user stories are classified according to functional and non-functional requirements. The functional requirements represent stories that are “comprehensible by both the developer as well as the customer team... and it’s a discrete

piece of functionality; that is, something a user would be likely to do in a single setting” [11]. The goal for requirements that are classified as non-functional requirements address the system needs, e.g. performance, availability, usability, security and capacity [11], which represent the system quality in general. Ethical requirements can be classified as non-functional requirements as they share similarities with quality requirements, for instance in terms of qualities such as security.

### 3 Research Framework

Sketching and prototype generation have been described as extensions of designer and developer cognition (see e.g., [15]). Likewise, for decades cards have been used as highly practical and effective tools for not only materializing thoughts but also representing how we mentally structure, categorise, and prioritise information [16].

Through utilizing cards in combination with light weight methods such as user stories in Agile processes, we may observe benefits from several perspectives: 1) concretizing the mental arrangements of information through arranging the cards; 2) physically re-ordering these cards to find better alternatives and smoother streams of logic; 3) direct information and guides for development; and 4) the ability to test user logic – in and of itself, and/or in light of the system and its re-design/re-development or improvement, and/or in relation to software developer logic while translating ideas generated from the cards into coherent and actionable stories (from scenario to program) [17].

In this study, we empirically evaluated the ECCOLA method. ECCOLA is a method for implementing AI ethics, which we have presented in an existing paper [10] and briefly above. The advantage of thinking tools such as cards and user stories – the types of tools utilised within this current study – for instance, are that they can be used repeatedly and iteratively throughout the design and development process. As their forms and functions also suggest, not only are these tools instruments for extending and validating thought, but they are also a means of engaging multiple minds – the input of several or many people – within the thought structuring, or cognitive development-action process. This facilitates and enables collective cognition through teams and developer-stakeholder (end-user) interactive and iterative processes [18]. In terms of designing for immaterial qualities, or non-functional requirements such as ethics, values and emotional experience for instance, these forms of tools are highly valuable as they serve to connect immaterial qualities to tangible and concrete design and development decisions.

For this study, we selected four cards for the teams to apply to their processes in order to see how using ECCOLA would affect how the teams take ethical issues into account while writing user stories. These cards were predetermined and were the same for each team, i.e., in this case the development teams did not pick the cards themselves. Due to research technical reasons, only four cards were chosen to conduct this study. We discuss the role of ECCOLA in this study in detail in the next section.

## 4 Study Design

In this section, we discuss the methodology used in the study. The purpose of the study was to understand how user stories could be written in terms of integrating ethical considerations (principles) into the actionable logic of the interaction design and SE process. The study was conducted as an experiment in a controlled research setting via the university's distance learning tools. According to Wohlin et al. "experiments involve more than one treatment to compare the outcomes. For example, if it is possible to control who is using one method and who is using another method, and when and where they are used, it is possible to perform an experiment" [19]. Our main interest was to compare the output of two types of user story generating student groups - the test groups utilising ECCOLA and a control group utilising a card set without explicitly concentrating on ethics. ECCOLA was used as a framework to guide the user story creation in the student groups who were assigned the test group role. The main goal of ECCOLA as a development tool and artefact is to aid the translation of seemingly non-functional requirements such as ethics, into operational SE actions. This experimental setting was considered apt for determining ECCOLA's effectiveness from this perspective.

### 4.1 Data Collection Methodology and Study Context

In the current study, focus was placed on the production of ethical user stories through utilising ECCOLA cards. ECCOLA had a two-fold function in this exercise: 1) as a guide for deliberating ethics in SE based on ethical AI principles; and 2) as a subject of appraisal - we sought to validate ECCOLA's effectiveness through its operationalization in user stories. In order to achieve this, data was collected in the form of user stories ( $n = 298$ ) from 15 project teams. Out of these 15 teams, nine teams utilized ECCOLA to aid the user story writing process, while six did not. Originally there was a more equal delegation of the two groups (ECCOLA and non-ECCOLA/control), but some groups were merged to avoid undermanned teams as some students opted to not complete the course.

The data for this study were collected from a Master's level Information Systems (IS) course at the University of Jyväskylä, Finland. In the course, students worked in teams of 3–5 students to carry out a project for a real case company.

The duration of the software project was six weeks. During this time, the students received five assignments, one each week after the first week's introductory lecture. These assignments comprised two parts: non-technical and technical. The non-technical part was the focus of this study and formed the basis of data collection. User stories were discussed during the lectures to familiarize the teams with the practice of producing them.

The students were split into teams based on self-evaluations of their software development skills. In a pre-course questionnaire, students were asked to evaluate their confidence in programming abilities in any programming language on a scale of 0 to 100. Students were organised in an ascending order based on their level of programming confidence, and divided incrementally into teams (i.e., the

most confident students into one group, the least confident into another, and the rest in between). Division was made in this manner in order to avoid imbalance of technical skills, and thus, workload distribution within each team.

For demographic data, students were also required to report their previous work experience in software engineering/development, in Agile development, and their experience in utilizing Scrum (see Table 2). While 61 percent of the students reported to have at least some experience in SW development (the distribution of experience levels between students in both ECCOLA and non-ECCOLA groups were similar) some difference in experience related to Agile development and Scrum can be seen between the two groups - to the benefit of the ECCOLA group. Students' experience in Agile development can be seen to relate to their prior knowledge about user stories as user stories are used as a RE tool in agile development work.

**Table 2.** Demographic data of working experience

Work experience:	How much work experience in the field of software engineering/development do you have?	How much experience in Agile development do you have?	How much experience of Scrum do you have?
	ECCOLA — Control	ECCOLA-Control	ECCOLA-Control
None	38% — 41%	49% — 64%	51% — 64%
Less than 1 year	28% — 23%	26% — 27%	28% — 23%
1–5 years	28% — 36%	21% — 9%	15% — 14%
6–10 years	0% — 0%	3% — 0%	3% — 0%
More than 10 years	3% — 0%	0% — 0%	0% — 0%
N/A	3% — 0%	3% — 0%	3% — 0%
TOTAL	100% — 100%	100% — 100%	100% — 100%

These teams were then split into two groups (X for odd numbers and Y for even numbers). Teams in group X used ECCOLA to help devise user stories, while group Y did not. Group Y, however, also received a set of cards. The cards issued were created for the present study setting. These cards contained instructions on writing user stories but did not discuss ethical issues. The purpose of the second set of cards was to encourage a sense of equal treatment between groups. Furthermore, the equality in issuing all groups with card sets was to ensure that learning outcomes were not compromised by perceived varying conditions and resources (i.e., tools at each group's disposal). Materials such as Card decks X & Y, the User Story Template, instructions and weekly assignments can be found at external repository at Figshare<sup>2</sup>.

As the course progressed, so too did the user story development process. During the first week of the project, after gaining a firm understanding of user stories, the groups were to write 4–6 user stories that featured functional requirements. Each user story was written on a template provided to the teams.

<sup>2</sup> <https://doi.org/10.6084/m9.figshare.14210753>.

The first week’s assignment required the students to utilize one card from their given set that comprised a Stakeholder Analysis theme. The second week’s assignment focused on examining the customer need/desired product description and writing 4–6 user stories through the lens of non-functional requirements (NFR). Students were informed about the differences of functional and non-functional requirements including examples of both types of requirements. For the rest of the project timeline the groups with X deck were allocated three additional cards to reference (four cards in total): Stakeholder Analysis, System Reliability, Privacy and Data, and System Security. The groups with Y deck were provided with two new cards in addition to Stakeholder Analysis (three in total): Non-Functional Requirements and Functional Requirements.

After the second week, each week featured user story revision, creation of new user stories if applicable, and a check to see if user stories were implemented into the product. At the end of the course, the groups were to review and return all their user stories with concluding remarks about the implementation process.

## 4.2 Data Analysis

The data was analyzed with coding techniques according to Grounded Theory Method (GTM) and the INVEST model. The use of GTM in IS studies varies in application rigour (degree of adoption) and type of research contribution [20].

There is no “unique, generally accepted set” of GTM procedures to guide the coding process [20], and the use of the method has evolved since its development. Regardless of the type of application, a key concept in GTM includes *coding* as a way to classify themes that arise in the data.

Before commencing analysis, the user stories were submitted by the student teams, categorized by assignment/week, and finally summarized in a table. The data were then analyzed in three phases. First, we looked at the data quantitatively in order to gain an overview and to look at any quantitative differences between the data from the two groups of teams. This was done due to the high volume of otherwise qualitative data.

In the second phase, we utilized a GTM approach to code the user stories one at a time. This process was carried out iteratively, with the list of codes updated during the process as new codes emerged. We chose this approach due to the fact that this is a novel area of research: we were not able to identify any existing studies on writing ethical user stories. Moreover, we chose a GTM approach as it is well-suited for discovering phenomena inductively [20]. We wanted to study the data by limiting possibilities for bias as much as possible. We also did not know which aspects of user story creation the use of ECCOLA might affect and how. Thus, we saw the need to examine the findings against a blank slate, making GTM ideal. In analyzing the user stories, we applied the GTM coding methodology of *open coding*, a coding where initial labels are attached to data [20]. The codes were not pre-determined, as we wanted to first apply themes to the data, and later categorize them in terms of their relevance to the research. It is possible, however, that a researcher bias from previous AI ethics research may have contributed to the themes that arose from the data.

In the third and final phase, we utilized the INVEST model. According to the INVEST model, the quality of the user stories can be evaluated with six attribute lists, in a method called INVEST [11, 12]. The acronym, introduced by William Wake (2003) stands<sup>3</sup>, for: I as an Independent, N as a Negotiable, V as Valuable to Purchasers or Users, E as an Estimatable, S as Small and T as Testable. A good user story can be composed through these elements, particularly when: it is not dependent on other user stories; can be negotiated as it does not go into detail; brings value to the customer; can somehow be estimated in terms of resourcing and anticipated amounts of customer support; and is small in size in order to be as accurate as possible for producing estimations. It should also be testable to assure the accuracy of the requirements.

To operationalize INVEST, firstly two teachers from the course evaluated the user stories through the framework. Each teacher analyzed equal number teams producing the user stories. Both teachers evaluated as many control group teams and ECCOLA teams each to reduce any potential bias. Then, one of the researchers scored all the user stories using INVEST, independently of the teachers' evaluations. The evaluation was binary: either a user story fulfilled the requirements of an INVEST attribute or not.

## 5 Findings

As discussed in the study design section, we collected 298 user stories from 15 student teams. These teams were split into two groups: group Y (the control group, i.e. the teams that did not use ECCOLA) and group X (the teams that used ECCOLA). Group Y had six teams in it, whereas group X had nine. Overall, group Y produced 119 users stories (average 19,8 per team) and group X produced 179 user stories (average 19,9 per team).

In the GTM coding, each story was attributed three high-level themes: (1) stakeholder, (2) requirement, and (3) technical orientation (T) vs human orientation (H). Inside these themes were lower level codes attributed to each theme, as seen in Fig. 1. Not all codes were present in every user story. For example, every user story had some stakeholder(s) present but the stakeholder(s) varied between user stories. Additionally, the human orientation vs. technical orientation codes were mutually exclusive, serving as a way of categorizing the user stories into two groups.

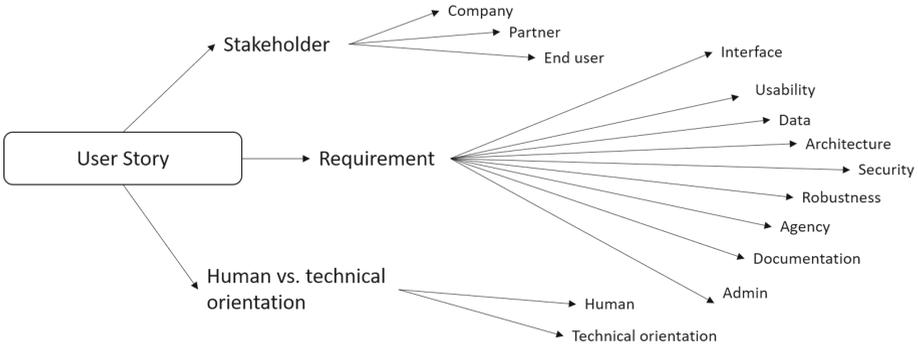
Whether a particular user story was human-focused or technology-focused was considered interesting from the point of view of ethics. This was of interest from the ECCOLA viewpoint, as we wished to understand to what extent ECCOLA might have influenced the user stories in relation to, e.g. consideration for human aspects. While this was a binary split in our analysis, the stories involved both human and technical aspects of the system, and were categorized based on which aspect was more dominant.

---

<sup>3</sup> <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.

The ECCOLA group produced more human-centric user stories (61%) than technology-centric ones (38%). The control group, on the other hand, produced more technology-centric user stories (65%) than human-centric user stories (31%). Based on these results, it seems that the use of ECCOLA could encourage developers to produce more human-centric user stories.

**PEC1:** (Primary Empirical Contribution): Using ECCOLA seems to result in more human-centric user stories.



**Fig. 1.** Grounded theory coding from user stories

Regarding the codes under the other two themes, the codes under the requirement theme were largely similarly represented in the user stories of both groups. For example, *security* codes were found in exactly 15% of the user stories of both groups. Thus, ECCOLA did not seem to result in any significant differences in the requirement codes.

The only notable differences could be seen in the *usability* and *agency* codes. The usability code was present in 29% of the user stories of the ECCOLA group, but only 9% of the user stories of the control group. The agency code was present in 8% of the ECCOLA group’s user stories, but only in 2% of the control group’s. It could be that the ECCOLA cards, in addition to resulting in more human-centric user stories in general, also served to highlight the user in terms of e.g., ease of use. The agency %’s in both groups were ultimately so low that it is too weak of an indicator of anything based on this data alone. Thus, using ECCOLA did not seem to increase consideration related to the ECCOLA card themes (System Security, Privacy & Data, and System Reliability) that were present in the cards utilized by the ECCOLA groups.

**PEC2:** Using the ECCOLA cards did not affect how the teams wrote user stories in terms of the themes present in the cards.

In addition to the GTM analysis, we utilized the INVEST framework to analyze the user stories. The results of this analysis are summarized in Table 3. Overall, the ECCOLA teams scored higher in quality according to the INVEST

framework. The ECCOLA group had an average INVEST score of 60,68% and the control group had an average score of 53,17%. The ECCOLA teams scored higher in every category of the INVEST framework aside from V(aluable). All the highest individual team scores were also in the ECCOLA group.

**PEC3:** Teams using ECCOLA produced higher quality user stories when measured using the INVEST framework.

Additionally, one of the largest differences in the INVEST scoring categories could be seen in the I(ndependent) category. Average INVEST scores for I – Independent for ECCOLA teams was 69,92% and for Control teams 46,54%. The user stories of the ECCOLA groups were notably more stand-alone than those of the control groups, i.e. they overlapped less in concepts. This can be beneficial as independent user stories can be produced and tackled before subsequent ones are written (as opposed to e.g., functionality 1 ->functionality 2).

**PEC4:** Using ECCOLA results in more independent user stories that consider the software from a wider perspective than just that of its functionalities.

**Table 3.** User store quality

Group	I	N	V	E	S	T	Average
Control	46,54%	8,80%	78,12%	70,53%	52,00%	63,01%	53,17%
ECCOLA	69,92%	12,88%	72,74%	81,25%	62,34%	64,97%	60,68%
Difference	-23,38	-4,08	5,38	-10,72	-10,34	-1,96	-7,52

INVEST analysis is conducted with binary coding. 1=User story fulfilled the requirements of an INVEST attribute 0=User story didn't fill the requirements. Shown on the table is percentage of positive scores.

In addition to the general INVEST analysis of all the user stories produced during the project, we looked at the second week’s user stories in detail. During the second week, the teams were tasked with producing non-functional user stories. The scores for each group and how they differed from the average INVEST scores of that group can be found in Table 4. Based on these scores, ECCOLA seemed to improve the quality of the non-functional user stories. More importantly, the overall good INVEST scores of the non-functional user stories of the ECCOLA group seem to support the idea that non-functional user stories can be written with ECCOLA, and particularly user stories of high quality.

In summary, the teams utilizing ECCOLA, while writing more human-centric user stories, considered ethical aspects in their user stories more than the control group. The control group largely focused on traditional SE aspects such as features and other technical design properties (Table 4).

**Table 4.** Non-functional user story quality

Group	I	N	V	E	S	T
Control group - Week 2 scores	92,61 %	N/A	77,71 %	44,46 %	29,56 %	29,68 %
Control group - difference from avg	46,07	N/A	4,83	-26,07	-22,44	-33,33
ECCOLA group - Week 2 scores	82,24 %	N/A	71,14 %	68,87 %	53,33 %	51,11 %
ECCOLA group - difference from avg	12,32	N/A	-1,60	-12,38	-9,01	-13,86

INVEST analysis is conducted with binary coding. 1=User story fulfilled the requirements of an INVEST attribute 0=User story didn't fill the requirements. Shown on the table is percentage of positive scores.

**PEC5:** Non-functional user stories can be written with the assistance of the ECCOLA method.

These findings and observations indicate that the team members utilizing the ECCOLA cards consider ethics in user story processing, while the control groups concentrated more on traditional SE development activities such as features and other technical design properties. Even though the chosen ECCOLA cards were the most technically-oriented the end result was then a human-oriented approach to user story production.

## 5.1 Validity Threats

In discussing ethical user stories, one limitation to consider is related to construct validity. How to measure the level of ethical consideration? This is also a general question related to studying the implementation of AI ethics. In this case, we have utilized a framework based on existing literature (ECCOLA). While there is currently no universally accepted consensus on what AI ethics is and what principles it should comprise, ECCOLA is constructed from some of the most prominent AI ethics principles (many of which e.g. [1] discuss).

Another potential limitation of the study is the empirical setting. To improve the reliability of the results, we chose an A/B testing based study setting and formed standardized procedures for data collection and analysis. We also utilized student data in this study. In this regard, we turn to Höst et al., [21] who argue that the differences between students and professionals are not statistically significant. We also argue that the use of students is justified by the novelty of the topic: we are not aware of any existing study that has looked into ethical user stories.

## 6 Discussion and Conclusion

In this paper, we have studied ethical user stories through the lens of the ECCOLA [10] method. In an experiment, we had developer teams (n=15) write user stories related to a real-world project. These teams were split into an ECCOLA group that utilized the tool to support them in writing user stories, and a control group that did not use ECCOLA to do so. We analyzed 298

user stories from these teams using two different analysis approaches. In Table 5 below, we summarize the Primary Empirical Contributions (PECs) of this study that we highlighted in the preceding section. Here, we discuss the implications of these findings before concluding the paper.

**Table 5.** List of primary empirical contributions

PEC	Description
1	Using ECCOLA seems to result in more human-centric user stories
2	Using the ECCOLA cards did not affect how the teams wrote user stories in terms of the themes present in the cards
3	Teams using ECCOLA produced higher quality user stories when measured using the INVEST framework
4	Using ECCOLA results in more independent user stories that consider the software from a wider perspective than just that of its functionalities
5	Non-functional user stories can be written with the assistance of the ECCOLA method

As the summarizing PECs in the above table show, the ECCOLA method [10] seemed to improve user stories in various ways. However, PEC2 also highlights an interesting observation in that ECCOLA did *not* make the user stories notably more focused on the themes of the ECCOLA cards in question. Moreover, the ECCOLA cards used in this study contained typical SE themes such as system security and privacy & data. These themes should be familiar for anyone concerned with SE and thus their lack of an effect in this study needs to be considered when moving forward with developing the method. Even if overall, ECCOLA produced positive results in this study, the contents of the cards may need adjusting based on PEC2.

Aside from evaluating ECCOLA, this study provides an initial look into writing ethical user stories. Bridging the gap between research and practice in AI ethics has been a recurring challenge in the area, with companies struggling to implement abstract ethical guidelines in practice [2, 3]. User stories can help us bridge this gap. Ethical issues should be considered as non-functional requirements among other ‘-ilities,’ such as usability and quality, and user stories can help companies formulate them into such. Although the ECCOLA method resulted in more ethical user stories in this study, it is but one option for supporting the creation of user stories that involve ethical consideration.

User stories traditionally place emphasis on Functional Requirements (FR) over Non-Functional Requirements (NFR) [22]. Depicting NFRs as User Stories has been suggested to include certain added challenges compared to FRs, such as that NFRs are not backlog items themselves, but rather constraints on development that are defined in the acceptance criteria for multiple backlog items<sup>4</sup>,

<sup>4</sup> <https://www.scaledagileframework.com/nonfunctional-requirements/>.

and being solution-wide, they may conflict with the user story requirement of independence<sup>5</sup>. The use of user stories in defining NFRs is not a novel concept, but the perceived difficulties taken into account, the creation of NFR user stories in this paper can be deemed successful.

Indeed, this study serves as Proof-of-Concept for ethical user stories. Using ECCOLA, developer teams were able to produce non-functional user stories that received high scores in INVEST (a framework for evaluating user stories). To facilitate the implementation of ethics in different context, such as AI ethics, formulating ethical issues into user stories can go a long way in making ethical issues tangible. Industry expert Mike Cohn posited that producing non-functional user stories is challenging but possible<sup>6</sup>, and our results seem to support this idea in the case of ethics as well, at least from the point of view of INVEST.

Future studies should look further into how ethics could be more easily transformed into requirements in SE. While user stories provide one possible avenue for doing so, other alternatives are also worth investigating. If the challenge in implementing ethics in practice (AI ethics or otherwise) is that ethical principles are difficult to translate into code and action, we should look into tools that developers are familiar with in order to make this process more accessible for those working hands-on with these systems.

**Acknowledgments.** The authors would like to thank the students for their active participation in the experiment. The authors also gratefully acknowledge being funded by three Business Finland research projects: Sea4Value Fairway, APPIA, and AMALIA-2020.

## References

1. Jobin, A., Ienca, M., Vayena, E.: The global landscape of AI ethics guidelines. *Nat. Mach. Intell.* **1**(9), 389–399 (2019)
2. Vakkuri, V., Kemell, K.K., Kultanen, J., Abrahamsson, P.: The current state of industrial practice in artificial intelligence ethics. *IEEE Softw.* **37**, 50–57 (2020)
3. Mittelstadt, B.: Principles alone cannot guarantee ethical AI. *Nat. Mach. Intell.* **1**, 1–7 (2019)
4. Miller, K.W., Larson, D.K.: Agile software development: human values and culture. *IEEE Technol. Soc. Mag.* **24**(4), 36–42 (2005)
5. Judy, K.H.: Agile principles and ethical conduct. In: 2009 42nd Hawaii International Conference on System Sciences. IEEE (2009)
6. Friedman, B., Kahn, P.H., Borning, A.: Value sensitive design and information systems. In: *The Handbook of Information and Computer Ethics*, pp. 69–101 (2008)
7. Nathan, L.P., Friedman, B., Klasnja, P., Kane, S.K., Miller, J.K.: Envisioning systemic effects on persons and society throughout interactive system design. In: *Proceedings of the 7th ACM Conference on Designing Interactive Systems* (2008)

<sup>5</sup> <https://www.linkedin.com/pulse/how-can-agile-teams-capture-non-functional-phil-robinson/>.

<sup>6</sup> <https://www.mountaingoatsoftware.com/blog/non-functional-requirements-as-user-stories>.

8. Andrews, A.A., Pradhan, A.S.: Ethical issues in empirical software engineering: the limits of policy. *Empir. Softw. Eng.* **6**(2), 105–110 (2001). <https://doi.org/10.1023/A:1011442319273>
9. McNamara, A., Smith, J., Murphy-Hill, E.: Does ACM’s code of ethics change ethical decision making in software development? In: *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 729–733 (2018)
10. Vakkuri, V., Kemell, K.K., Abrahamsson, P.: ECCOLA: a method for implementing ethically aligned AI systems. In: *Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications, (SEAA2020)*, pp. 195–204. IEEE (2020)
11. Cohn, M.: *User stories applied for agile software development*. In: Cohn, M. (ed.) 13th ed. Pearson Education Inc., Indiana (2009)
12. Dimitrijevic, S., Jovanović, J., Devedžić, V.: A comparative study of software tools for user story management. *Inf. Softw. Technol.* **57**, 352–368 (2015). Mihailo Pupin Institute, Volgina 15, Belgrade, Serbia; FON, School of Business Administration, University of Belgrade, Jove Ilića 154. Belgrade, Serbia
13. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods: review and analysis. *Proc. Espoo* **2002**, 3–107 (2002)
14. Cohn, M.L., Sim, S.E., Lee, C.P.: What counts as software process? Negotiating the boundary of software work through artifacts and conversation. *Comput. Support. Cooper. Work (CSCW)* **18**(56), 401 (2009). <https://doi.org/10.1007/s10606-009-9100-4>
15. Suwa, M., Gero, J.S., Purcell, T.: The roles of sketches in early conceptual design processes. In: *Proceedings of Twentieth Annual Meeting of the Cognitive Science Society*, pp. 1043–1048. Lawrence Erlbaum Hillsdale, New Jersey (1998)
16. Rugg, G., McGeorge, P.: The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. *Exp. Syst.* **14**, 80–93 (1997)
17. Bers, M.U.: *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge, Abingdon (2020)
18. Papatheocharous, E., Nyfjord, J., Papageorgiou, E.: Fuzzy cognitive maps as decision support tools for investigating critical agile adoption factors. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J. (eds.) *LESS 2013. LNBIP*, vol. 167, pp. 180–193. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-44930-7\\_12](https://doi.org/10.1007/978-3-642-44930-7_12)
19. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering*. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-29044-2>
20. Wiesche, M., Jurisch, M.C., Yetton, P.W., Krcmar, H.: Grounded theory methodology in information systems research. *MIS Q.* **41**(3), 685–701 (2017)
21. Höst, M., Regnell, B., Wohlin, C.: Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empir. Softw. Eng.* **5**(3), 201–214 (2000). <https://doi.org/10.1023/A:1026586415054>
22. Behutiye, W., Karhapää, P., Costal, D., Oivo, M., Franch, X.: Non-functional requirements documentation in agile software development: challenges and solution proposal. In: Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (eds.) *PROFES 2017. LNCS*, vol. 10611, pp. 515–522. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-69926-4\\_41](https://doi.org/10.1007/978-3-319-69926-4_41)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

