

Design Thinking and Acceptance Requirements for Designing Gamified Software

Luca Piras^{*}, Daniele Dellagiacomà[†], Anna Perini[‡], Angelo Susi[‡], Paolo Giorgini[§], John Mylopoulos[§]

^{*}Centre for Secure, Intelligent and Usable Systems, University of Brighton, Brighton, United Kingdom

Email: L.Piras@brighton.ac.uk

[†]DeltaLab Research and Development Division, Delta Informatica SpA, Trento, Italy

Email: daniele.dellagiacomà@deltainformatica.eu

[‡]Software Engineering Research Unit, Fondazione Bruno Kessler (FBK), Trento, Italy

Email: {perini, susi}@fbk.eu

[§]Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Email: {paolo.giorgini, john.mylopoulos}@unitn.it

Abstract—*Gamification* is increasingly applied to engage people in performing tool-supported collaborative tasks. From previous experiences we learned that available gamification guidelines are not sufficient, and more importantly that motivational and acceptance aspects need to be considered when designing gamified software applications. To understand them, stakeholders need to be involved in the design process. This paper aims to (i) identify key requirements for designing gamified solutions, and ii) understand if existing methods (partially fitting those requirements) can be selected and combined to provide a comprehensive gamification design method. We discuss a set of key requirements for a suitable gamification design method. We illustrate how to select and combine existing methods to define a design approach that fits those requirements using *Design Thinking* and the *Agon framework*. Furthermore, we present a first empirical evaluation of the integrated design method, with participants including both requirements analysts and end-users of the gamified software. Our evaluation offers initial ideas towards a more general, systematic approach for gamification design.

Index Terms—Acceptance requirements, Design thinking, Gamification, Requirements engineering, Empirical evaluation.

I. INTRODUCTION

Motivating people in accepting and using a software system is a crucial factor for the success of such systems [1]–[4]. *Gamification* has been identified as a mean to meet such *acceptance requirements* through the inclusion of game elements in systems that operate in non-game contexts [5] (e.g., air traffic management and decision making [3], software engineering tasks [6]).

This work has been partially funded by the SUPERSEDE project (H2020 EU Framework Programme, agreement number 644018). We would like to thank the participants to our empirical study.

This work was partially supported by ERC Advanced Grant 267856, titled “Lucretius: Foundations for Software Evolution”.

This work was partially supported by the DEFEND EU project, funded from the European Unions Horizon 2020 research and innovation programme under grant agreement No 787068.

This project has received funding from the SESAR Joint Undertaking under grant agreement No 699306 (PACAS EU project) under European Unions Horizon 2020 research and innovation programme.

It is worth noting that gamification is different from serious games where game playing is used to create a better understanding of a particular concept in a learning situation (e.g. emergency management training [7]). Although both serious games and gamification help to motivate users of a software application by making usage experience enjoyable, their design processes are different. Designing serious games requires a deep understanding of learning objectives [8], while gamifying a tool-supported task requires understanding the motivations driving users [2], [9]–[11]. Examples of motivations based on specific kinds of users are [12], [13]: (i) using the software in a collaborative/social way by interacting as much as possible with other users (socializers [12]); (ii) learning as much as possible how to use a system, for being able to help the others users, to be recognized as expert of a system (explorers [12]). Other motivation examples relevant for specific software contexts are: (i) fixing as many bugs as possible, while performing software testing, or (ii) identifying the most important requirements while performing requirements prioritization.

Current gamification solutions can be found both in academic and industrial software systems. On one hand, practitioners tend to use available gamification guidelines and resources provided in commercial platforms or in publicly available wikis¹, which neglect a deep analysis of users and usage context. For this reason, many gamified systems achieve poor engagement of users. On the other hand, scientific research literature offers methods from different fields, notably human-computer interaction, that ensure successful results but require a complex and expensive process of analysis. However, research literature on the evaluation of the effectiveness of the resulting solutions points out the limitations of current practices, identifies key concepts and discusses the need for better methods to design engaging software [2], [14], [15]. In particular, requirements analysis should be guided by systematic methods for exploring a design space of alternatives [1]–[4]. Such a design space is defined in terms of motivational,

¹e.g. https://en.wikipedia.org/wiki/Game_mechanics

psychological, cognitive, behavioral factors [2] that influence the fulfillment of *Acceptance Requirements* [1], [3], [4].

An interesting example of gamified software lacking of acceptance from its users is DMGame. Specifically, it is a gamified collaborative requirements prioritization tool [16], [17] developed within the SUPERSEDE project². The tool has been evaluated in the context of three industrial use cases. Moreover, the effectiveness of specific game elements was further investigated through a controlled experiment that confirmed a lack of acceptance by its intended users [16]. In this work, we consider the DMGame as an illustrative case study of a software system lacking of acceptance and, by using the novel method we propose in this paper, we improve its gamification on the basis of a deep, systematic analysis.

Specifically, the main goals of this paper are to i) identify key requirements for a suitable method for designing gamified tool-supported tasks, and ii) understand to what extent existing methods that partially fit those requirements, can be selected and combined to provide a suitable gamification design method. Moreover, we are interested in evaluating the resulting method.

To address the first research objective, we analyze related literature, observations collected during the DMGame case study and other experiences within EU projects (e.g., SUPERSEDE², Lucretius³, DEFEND⁴, PACAS⁵, VisiOn⁶, STREETLIFE⁷), and derive a set of requirements for a suitable gamification design method. We address the second one by analyzing characteristics of existing methods that can be related to the stated requirements of the method. Moreover, we propose two candidate methods that, integrated together, can result into a method that fulfills those requirements. One is *Design Thinking* (DT) [18], [19], a method for innovative design that has been applied in very different domains, ranging from IT, to Medicine and Architecture, which has been revisited from the perspective of requirements engineering, e.g. [20]. The other is the *Agon* method [1]–[4] for analyzing acceptance requirements and operationalizing them through game elements. The resulting method is called DTA (*Design Thinking & Agon*). In the context of our research we assess the effectiveness of DTA in terms of the degree of coverage of the stated key requirements, usefulness as perceived by its users, namely designers of gamified tool-supported tasks, as well as the quality of the resulting designed solution.

The rest of the paper is structured as follows. Section II presents the DMGame case study, the problem faced and lessons learned. In Section III, we propose the key requirements of a collaborative method for designing gamified software. Section IV shortly recalls the DT and Agon methods, and discusses how they individually satisfy part of the key requirements. In Section V, we propose DTA, the method that combines DT and Agon, and that aims at fully covering the key requirements.

Section VI describes the empirical evaluation of the DTA method. Section VII discusses related work, and Section VIII concludes and outlines future work.

II. THE DMGAME CASE STUDY

Software development projects have typically limited resources in terms of time and budget, thus it is essential for the development team to prioritize requirements and implement them accordingly. Tool-supported collaborative requirements prioritization tasks are increasingly used in software engineering practice, but are still rarely adopted by small companies [21]. Possible reasons for this fact have been investigated in the SUPERSEDE project², where a configurable tool-suite has been developed, which includes plugins for the popular issue tracking system JIRA⁸ [22]. This tool-suite has been validated in the project's industrial use-cases.

The SUPERSEDE tool for supporting collaborative requirements prioritization is called DMGame [17]. It is a collaborative requirements prioritization tool, which allows the members of the development team to express their preferences on requirements priorities, according to one or more criteria, and provides a final requirements ranking by combining these preferences. The DMGame exploits automated reasoning techniques such as the Analytic Hierarchy Process (AHP) and genetic algorithms.

Gamification has been applied to foster team members' engagement [17], especially in those activities perceived as boring, such as expressing their preferences on a large set of requirements pairs, where the AHP method is used. Using the DMGame, a collaborative prioritization task is presented as a game, where decision makers are players, who can be assigned the role of opinion provider and of negotiator. Examples of game elements used in DMGame can be seen in the upper part of the screenshot depicted in Fig. 1. For instance, *accomplishment percentage* indicates to the user her progress on the assigned task, and *points* can be gained by the user according to specific rules (e.g., one rule is based on the agreement index with the other task's contributors). In the lower part of the screenshot in Fig. 1, pairs of requirements are presented (columns *First Requirement*, *Second Requirement*), and their relative importance according to *User impact* criterion can be expressed on a "1...9" AHP scale. In the DMGame version based on genetic algorithm, instead, requirements are presented in a list and the user can input her evaluation on their relative importance by moving up or down in the list each box representing an individual requirement.

The gamification effectiveness in DMGame has been evaluated by the two small/medium enterprises and the two large companies involved in the project, as well as in a controlled experiment with University of Trento master students [16]. The analysis of the collected data indicates that the gamification impact was quite limited. Although gamification increased slightly the level of engagement and flow, users did not perceive employed game elements as useful and sufficiently engaging.

² SUPERSEDE is a EU H2020 project: <https://www.supersede.eu>

³ Lucretius is a ERC project: <http://www.lucretius.eu/>

⁴ DEFEND is a EU H2020 project: <https://www.defendproject.eu/>

⁵ PACAS is a EU H2020 project: <http://www.pacasproject.eu/>

⁶ VisiOn is a EU H2020 project: <http://www.visioneuproject.eu/>

⁷ STREETLIFE is a EU H2020 project: <http://www.streetlife-project.eu/>

⁸ An agile software development tool: <https://www.atlassian.com/software/jira>

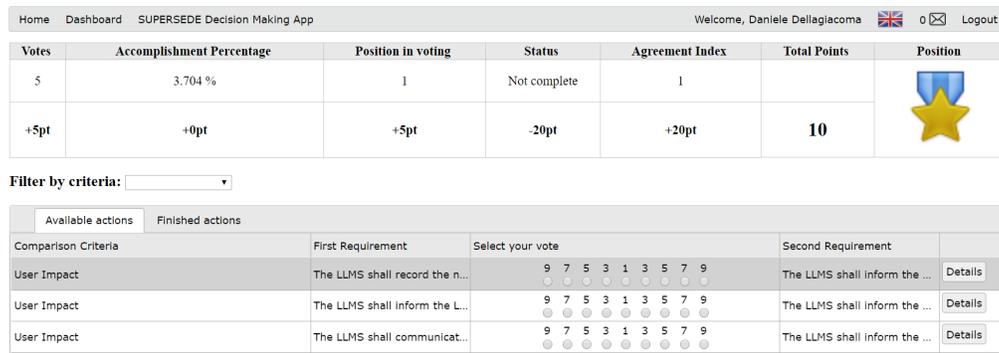


Fig. 1. DMGame: the GUI of the gamified AHP-based version [16].

Gamification in DMGame has been designed following practitioners' guidelines¹, but the resulting poor effectiveness can be seen as an indicator that guidelines are not sufficient [16].

A. Lessons Learned

The analysis of the literature, the DMGame case study, and experiences performed within EU projects (e.g., SUPERSEDE², Lucretius³, DEFEND⁴, PACAS⁵, VisiOn⁶, STREETLIFE⁷), allowed us to reflect on aspects concerning the gamification of tasks in terms of stimuli and game elements and, more generally, on the need of a structured method allowing to explore in a systematic, but creative, way alternative gamification solutions. We derived the next lessons learned.

LL1 : *A Wider, Creative, User-Focused Design Approach.*

Points-based elements [16] have to be integrated with other gamification elements to amplify the engagement effect on users [1]. In fact, different kinds of users react in different ways when exposed to different stimuli generated by different game elements [13], [23]. Therefore, to find proper stimuli for engaging the intended users, it is needed a more sophisticated and user-focused gamification design [11], [24], [25]. Indeed, both in the literature and in real cases from the industry, there are many evidences revealing that, in most of the cases, a gamification design only points-based is not enough to obtain acceptance and engagement on the target users [13], [16], [17], [23], [26]. Thus, It is required a user-focused deep analysis aiming at individuating a wider variety of effective elements to use, for producing a more exhaustive gamification design, made of heterogeneous elements, really able to engage most of the different typologies of users [2], [10], [12]. Furthermore, having a broader set of effective gamification alternatives to choose, enables also the analyst to select, more easily, the most engaging elements fitting also the other system requirements, and to discard the ones being in conflict [1], [3], [4]. In fact, it happens that some gamification solutions, even being effective for the

intended users, could be in conflict with important system constraints, and could potentially break other important requirements such as privacy and security requirements [3], [4]. Therefore, to enable the analyst to design a comprehensive gamification solution, made of all the elements needed to create a really engaging gamification solution for the intended users, not only based on points [16], requires also the employment of concepts and techniques highlighted in the next lesson learned;

LL2 : *Stakeholders Involvement and Collaboration.*

Successful gamification requires a deep understanding of the users and their motivations also in relation to the task they have to carry out by using the software [1]–[3]. Furthermore, this calls for involving deeply the stakeholders in the design process, making them empathize with the task through creative [21] and collaborative design methods [17], for extracting profound motivations of people when involved in a gamified task in a given context. This enables analysts and stakeholders to identify collaboratively which design mechanisms can concretely motivate the intended users of a specific context [3];

LL3 : *Understanding Acceptance Requirements.*

A deeper understanding of the social context of usage is needed [2], as well as the psychological, behavioral, cognitive considerations that help to determine how software should be gamified [3], [14];

LL4 : *A Systematic and Traceable Approach.*

To address the above lessons learned, gamification analysts need a systematic approach [1]–[3] that is able to promote and balance collaboration and creativity during the design process [15], [17], [21]. Artifacts generated during the design process should support traceability of stakeholder characteristics, task objectives and acceptance requirements, so to support evolution of the gamified tool-supported task.

III. KEY REQUIREMENTS OF A METHOD FOR DESIGNING GAMIFIED SOFTWARE

Reflecting on the lessons learned presented in the previous section, which root in literature and practical experiences within EU projects (e.g., SUPERSEDE², Lucretius³, DEFEND⁴, PACAS⁵, VisiOn⁶, STREETLIFE⁷), we identified the following key requirements for designing gamified tool-supported tasks. We discuss them highlighting how they relate to lessons learned.

R1 : *Systematic, Traceable Approach*. This is derived from **LL1**, **LL3**, **LL4** due to the following reasons. Specifically, the method has to be systematic and traceable [1]–[3] (**LL4**) (a) allowing the exploration of most of the factors influencing the user in being motivated to use a system [2], [14] (**LL3**), and (b) employing effective techniques for analyzing and tracing those factors and designing mechanisms able to satisfy them [3] (**LL1**, **LL3**);

R2 : *Participation/Collaboration and Creativity*. The design method should support participation/collaboration by stakeholders, and also encourage creativity during the design process. These are derived from **LL2**. In fact, to address factors of “**R1** (a)”, techniques and mechanisms mentioned in “**R1** (b)” [3] are needed as well as approaches that support deeply involving stakeholders [17], [21] (**LL2**) and making them to collaborate (**LL2**) in a creative way [21]. To involve stakeholders in the design is fundamental, because they have really the knowledge required for analyzing deeply the specific domain, the situation and the user [17], [21]. Collaboration [17] pushes them towards brainstorming finding more valuable ideas together. Creativity [21] fosters finding richer and more complete solutions. Thus, techniques mentioned in “**R1** (b)”, to maximize their effectiveness, should be executed in a participatory, collaborative and creative fashion [17], [21];

R3 : *Acceptance Orientation*. From **LL3** we derive that the design method should have an acceptance orientation, because the final aim of the method is to design engaging software for the user. Thus, it is fundamental to explore most of the factors (see “**R1** (a)” above) that positively influence the user such as psychological, motivational, cognitive and behavioral factors [2]. These are usually referred to *Acceptance Requirements* [1]–[3] and related techniques [3]. They are crucial for selecting psychological strategies as design mechanisms to use to make the software attractive [1]. Thus, the method has to be acceptance oriented (**LL3**);

R4 : *Gamification Orientation*. The method should support the selection of gamification elements to operationalize acceptance requirements. This is derived from **LL1** and **LL3**. In fact, factors analyzed through acceptance

requirements have to be mapped with gamification concepts able to fulfill such requirements. It is important to choose those concepts as well as to decide properly how to put them together in a coherent and effective gamification design [1]. Such method has to support both these aspects, namely it has to be gamification oriented. With gamification oriented we mean that the method should incorporate the gamification design knowledge [3] (**LL1**), and provide related techniques, able to support the analysts in producing a high-quality gamification design made of concepts, satisfying acceptance requirements, organized according to gamification best practices [1] (**LL1**, **LL3**);

R5 : *Context Characterization*. The method should support the characterization of the social context within users using the system. This is derived from **LL1** and **LL3**. The identification of acceptance and gamification strategies that are effective for a specific kind of user, depends strictly on the specific context variables [1]–[3] (**LL1**, **LL3**). Thus, such method has to support the characterization of context variables such as the human as a user and as a player [1], the goals and needs of the user, the task that the user carries out by using the software and related positive and negative user feelings, what can be produced by using the software and in which social context [2] (**LL3**). Moreover, a method supporting context characterization uses these considerations in individuating which acceptance and gamification strategies best fit the specific context [3] (**LL3**). Thus, this requirement is needed also for preventing the so called “paradox of mandatory fun”, which is discussed for example in [27], that will result in a poorly accepted software application;

R6 : *Guiding Approach*. The design method should offer guidance to designers on how to design gamified software. This is derived from **LL1** and **LL3**. Specifically, such method has to provide a reference to analysts and stakeholders (**LL1**, **LL3**) meaning that it has to: (i) guide them in all the phases, by providing techniques supporting all the other aspects [1], [3], (ii) make them to explore as many as possible relevant elements [2] and, above all, (iii) provide suggestions concerning psychological strategies, gamification concepts and suitable best practices for the context characterization [1], [3];

R7 : *Solution Ideation*. The method should support various forms of ideation, including sketching, prototyping, wireframing and mockups [17], [21]. This is derived from **LL2**. This is beneficial to make more concrete the suggestions obtained and ideas produced. Thus, the method has to support the techniques mentioned above for devising a solution [17], [21] (**LL2**) in a collaborative and creative way.

IV. SELECTING CANDIDATE METHODS

As an example of how we intend to understand if existing methods can be selected and combined to provide the gamification design method, we consider two methods, DT [18] and the *Acceptance Requirements Analysis* of the *Agon framework* [1]–[4], that separately are able to cover partially the requirements above. However, with this contribution, we are not claiming that DT and Agon is the only possible combination of methods for fulfilling those requirements. In fact, we do not exclude that other methods can be combined obtaining a method compliant with those key requirements as well as our combination.

A. Design Thinking

DT [18], [19] aims at boosting a state of mind towards innovation, leveraging on empathy with key stakeholders, in our case the intended software users. The DT method rests on a peculiar process, which involves different stakeholders (e.g. people playing different roles in a software company), who collaborate closely, generating many ideas and concepts, for devising a solution that best suits the initial problem. The DT process consists of the following five iterative steps, each one having its own objective: *Empathizing*, where participants empathize with the typical users of the software application to be renovated to understand their goals, needs and frustrations; *Problems Investigation*, where participants define the typical day of the user, by describing activities and positive/negative feelings; *Solution Ideation*, where participants creatively generate ideas and decide which should be used in next phases; *Prototyping* of a possible solution and *Solution Testing and Presentation*.

For each steps specific techniques can be used. For instance, in the first step, techniques taken from participatory design are usually exploited, such as the definition of *Personas*.

B. Agon: Acceptance Requirements Analysis based on Gamification

The Agon framework [3], [4] and its method [1], [2], [4] support the analyst in analyzing acceptance requirements and fulfilling them with gamification design concepts. Furthermore, Agon provides the analyst with models [3], techniques [1], [2] and a tool for executing its method, a systematic acceptance requirements analysis in which gamification is considered as solution mechanism. In a first phase, the analyst defines and elicits acceptance requirements on the basis of the context characterization. The method supported by Agon analyzes and characterizes the (as-is/to-be) software, looking for the crucial functionalities to be gamified, focusing on functionalities that are complex, or not attractive and whose fulfillment lead to the fulfillment of software goals. Moreover, Agon allows to characterize the user, specifying age or gender, her player categories (e.g., socializer, achiever, explorer and killer according to the Bartle's taxonomy of player types [12]) or her expertise regarding software usage. In a next phase the analyst can use the Agon Acceptance Model [3] to receive suggestions, as psychological needs and strategies that best fit the specific context [1], [2] identified in the first phase. As last

phase the analyst can use the Agon Gamification Model [3], containing different gamification elements, for producing a gamified design as a solution [1]–[4].

C. Why Design Thinking and Agon?

DT and Agon constitute the complementary baseline for the method we envisage in this work called DTA (Design Thinking & Agon).

They were chosen for two reasons. Firstly they both build on the principle that users need to be motivated in order to use any method or tool. DT does this for its users, requirements engineers and stakeholders, while Agon does this for the end-users of a system. Secondly, together they address all the requirements identified earlier.

DT partially covers **R5** considering user characteristics, needs and goals, while Agon fills the DT's gaps by characterizing the user regarding her player characteristics and the social context where she uses the software to accomplish her goals. Though DT can partially consider context variables, we cannot define it as completely acceptance oriented (**R3**), because DT does not guide the analyst in using context variables for eliciting which psychological strategies to use for improving software functionalities, and which components fulfilling those strategies could be inserted. The same is for **R4**, because DT does not specifically consider gamification design and related techniques for enhancing software functionalities. While, Agon covers the **R3** and **R4** requirements. Furthermore, DT is partially systematic (**R1**) in the sense that, it provides process guidelines and proposes design mechanisms for analyzing relevant motivational factors, but those are focused only on a subset of relevant variables pertaining **R5** and **R3**. Accordingly, DT partially covers **R6** due to the lack in providing **R3** and **R4** suggestions. Agon is **R1** and **R6**, though, focusing specifically on **R3** and **R4** aspects and the related **R5** requirement.

In summary, DT enables a participatory, creative process, where solution prototypes are developed and validated as part of the exploration of the solution space, which fully covers **R2** and **R7** requirements. Thus, we propose DT as the backbone of DTA. On the other side, Agon provides key requirements engineering concepts for gamification design, such as acceptance requirements and context characterization. Moreover, Agon provides a systematic way to address those requirements by using gamification mechanisms, fully covering the **R1**, **R6**, **R3** and **R4** requirements.

V. THE DTA METHOD

The DTA steps and key concepts are shown in Fig. 2. At the top, there are the steps derived from DT and at the bottom the ones from Agon. The arrows show contributions (suggestions) coming from Agon to DT. Next, DTA is summarized.

Context Characterization. The first step of DTA is the context characterization. In Fig. 2 are listed the context variables analyzed and characterized. The context variables include user characteristics [1], [4] (e.g., age, gender, etc.), user awareness regarding software usage, how the user perceives the task (e.g., communication level, clarity level, etc.) and the goal

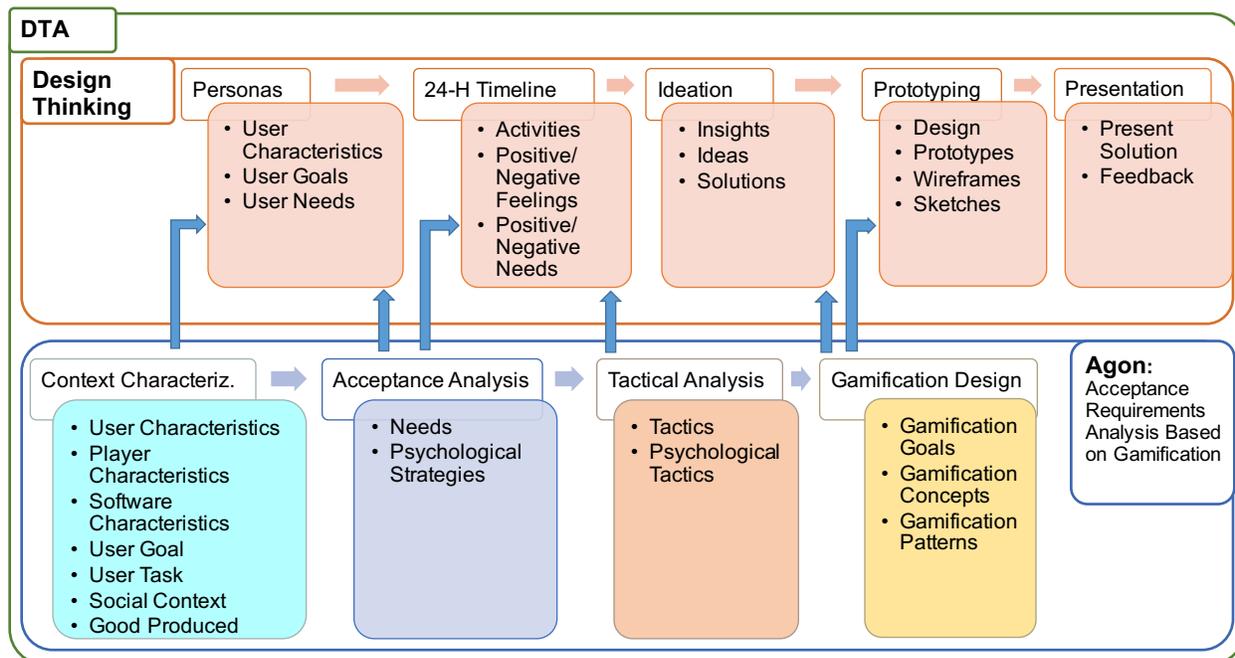


Fig. 2. The DTA method obtained integrating Design Thinking and the Agon Method. DTA steps can be iterated for further refinement of envisaged solutions.

related to the software use, in which social context the software will be used (e.g., hierarchical or neutral) and which kind of good is produced by using the software [2], [4] (e.g., private or public good). Another variable used to characterize the context is the player type [3], [4]. According to the Bartle's taxonomy [12], players can be divided into socializers (i.e. they desire to interact with other users and participate in an active and wide community), explorers (i.e. they desire to discover as many as possible parts of the system such as services and features), achievers (i.e. they prefer to gain rewards from the game) and killers (i.e. they want to prevail against other players). The context characterization can be accomplished beforehand by interviewing the users or stakeholders, or through questionnaires [3], [4].

Personas Characterization. Then, participants use the context characterization variables for defining collaboratively personas. Personas are fictional characters that represent the typical users of the system. Defining personas helps to understand users' needs, experiences, behaviours and goals. Moreover, participants receive as suggestions (thanks to the *Acceptance Analysis* of Agon [1], [4]) further information concerning needs, psychological factors and strategies that best fit the context and, above all, the user from different perspectives. DTA helps participants to collaboratively define personas providing suggestions from the context characterization, and users' needs and psychological strategies from the Agon acceptance analysis. Participants are free to decide to use such information, re-adapt them or add other user's needs which can be relevant in characterizing personas.

Problems Investigation. During this step, participants collaboratively discuss and compile the 24-hours timeline for describing the user activities of 24 hours, concerning software use and correlated tasks, and individuating the related user's

positive/negative feelings and needs. This step, along with the previous one, helps participants to empathize with the typical users of the software application to be renovated. Participants carry out this step being helped by context variables, personas, needs and psychological factors and strategies suggested so far. Moreover, they receive as further suggestions (thanks to the *Tactical Analysis* of Agon [1], [4]) more concrete sets of possible psychological elements, tactics refining psychological strategies, that best fit the context. These suggestions give participants a broad set of elements to consider for describing more precisely the 24-hours timeline. For instance, software and task characterizations analyzed before, or the suggestions on psychological needs received previously, can help and speed up the participants in this step.

Solution Ideation. Here, participants collaboratively produce and discuss ideas. They can use all the elements collected so far such as characterizations, personas, 24-hours timeline and received suggestions (e.g., psychological and tactical strategies). Moreover, on the basis of collected elements, participants receive also gamification design suggestions (thanks to the *Gamification Design* of Agon [3], [4]). Such suggestions, being based on collected elements, are the most suitable gamification concepts and gamification design patterns for the specific context. For instance, they are the most suitable for the user to be motivated, the software to be used, the task to be carried out, etc. Thus, participants are deeply helped with ready-to-use gamification design suggestions that stimulate their creativity to produce more valuable ideas in less time.

Prototyping. In this step, participants put into practice ideas generated in the previous step. They collaboratively create sketches, wireframes, mockups or even prototypes of their ideas. As for the previous step, they can use all the elements collected so far, especially suggestions regarding gamification concepts

and gamification design patterns for the specific context.

Solution Presentation. Finally, in the last step, each team presents its gamification solution and prototype to receive feedback from the other teams.

VI. EVALUATION

The DTA method fits the proposed requirements for a suitable gamification design approach by construction. An empirical evaluation can support us to get empirical evidences about the effectiveness of the method as perceived by requirements analysts and gamification designers. Moreover, we can get further empirical evidences by evaluating with end-users the quality of a gamified solution developed with DTA.

We use a case study evaluation. In the first case study (Section VI-A), we use DTA to understand how to improve the DMGame user's engagement by exploiting gamification. The existing DMGame version is considered, and the objective is to develop ideas and a prototype for improving user engagement in the DMGame tool. In the second case study (Section VI-B), we use an interactive mockup of the renewed DMGame, which was obtained by applying the DTA method, and ask end-users, of the original version of the DMGame, to evaluate it. Complete results of the evaluation are detailed in [28].

A. Requirements Analyst Perspective Case Study

This case study focuses on the perspective of the requirements analyst, who uses DTA to design a gamified software system. Our objective is to get an empirical evidence about the perceived usefulness of DTA for the analysis of requirements that can be operationalised in terms of gamification solutions. **Case Study Design.** The participants to this case study are four members of the research team in charge of developing the DMGame tool and one of the author, who knows DMGame and Agon, and joins the team with the role of conductor of the case study. The agreed objective of the case study is that of improving the tool with respect to user engagement and gamification design aspects, and to address some issues related to acceptance that emerged in the previous evaluations with end-users. A post-study questionnaire is designed to assess as the study participants perceived the usefulness of DTA.

The case study conductor plans for the first step in DTA, concerning the context characterisation, that will be performed running a survey with DMGame users, and for the connected Agon-oriented steps (see steps depicted in the lower part of Fig. 2), thus preparing material to be used in a two-hours plenary session with the team. In the session, his role is mainly to guide the team following the five steps depicted in the upper part of Fig. 2 within the available time, and using the input from the Agon-oriented steps, but without participating to the discussion and design decision. The team members work in pairs during this session.

Case Study Execution. The first activities performed in this case study concerned the elicitation of requirements for the system to be gamified, that is the DMGame, and the context of the user such as user characteristics (e.g., level of expertise regarding software usage, how the task and the goal correlated

to the software use are perceived, in which social context the software will be used). These aspects have been elicited through a questionnaire that was filled in by developers, analysts and designers from companies involved in the SUPERSEDE project², who are potential users of DMGame. Eleven replies to the questionnaire were collected. Main findings from their analysis are the following.

Most of the users of the DMGame are *males* and their age is between *31 and 40 years old*. Their experience with requirements prioritization tools is quite limited and they have never used requirements prioritization tools similar to the DMGame before. The goal of the DMGame is clear to them.

Referring to the usage of a gamified software system, users of the DMGame are mainly *socializers* (i.e. they desire to interact with other users, talk with new people, create new relationships and friendships, participate in an active and wide community) and *explorers* (i.e. users who desire to discover as many as possible parts of the system such as services and features and, by acting in this way, their aim is to acquire important skills and to become like a "Guru" for the other users, colleagues, especially for the novice ones).

These context characterization variables are used to derive a set of suggestions about psychological needs and strategies that best fit the specific context, to be used then in the planned DTA session. According to the Agon suggestions, the target users can be motivated to use the DMGame in different ways. The first one is to *give them just the perception*, that the use of the tool can provide some *benefits*. For instance, creating the perception that using the DMGame tool there are improvements in terms of their speed in deciding, quality of the decision, and, more generally, productivity or effectiveness of their work.

Another strategy is to show them that by using the DMGame tool they can obtain *personal benefits*. For instance, saving time for interesting activities, improving themselves and their skills to increase the chance for a promotion, and obtaining prizes for their activity.

One more strategy is related to the addition of elements for augmenting the *social influence*, supporting *social behaviour* and promoting *collaboration*. For example, in relation to the *social influence*, the user behaviour can be influenced by increasing the social relationships and suggestions (e.g., suggestions to use the system) from people who are considered important by the user. With the aim of increasing the participation of the individual, *social influence* can be used also by showing the user that it can improve her social status (e.g., with colleagues). Accordingly, it can be useful to show to the user that many people (e.g., colleagues) use the tool, and that the use of the tool is both *socially supported* (e.g., the company invests on improving the skills of the employees regarding the system usage, for instance by courses or online tools) and *socially pushed* (e.g., the company pushes in a positive way, i.e. not forcing, all the employees in using the software, for instance by showing some advantages).

Besides psychological needs and strategies, Agon uses context characterization variables to suggest gamification elements for that specific context. In this case, Agon suggests to

define *point systems*, in particular, assigning to users *skill points* (i.e. awarding the user for specific actions related to the usage of the system) or *karma points* (i.e. awarding for actions focused on social matters, for instance, helping/supporting a colleague in using a system; karma points can be redeemed then for other social matters, for improving social relationships among colleagues, for example, by giving the user the possibility to make gifts to other colleagues). These point systems require the definition of achievement rules for specifying when and in which way the user earns each kind of points. Points can be used to set leaderboards. Different kinds of *leaderboards* can be implemented, e.g., social leaderboards, time-dependent leaderboards or leaderboards connected to particular users roles.

Another gamification element suggested by Agon is the *badge* concept. In particular, the most suitable badge kind for users of our case study are the badges that should be awarded for successful actions. Furthermore, badges should be *publishable*. Also in this case, badges need to be supported by achievement rules. To support publishable badges, it is necessary to define a *virtual environment* where users can form a *community*. Moreover, the system can make suggestions to the users, for example: to recommend events (e.g., for taking part to system activities by collaborating with other colleagues, or for respecting system deadlines), to ask a user to motivate another user to attend an important activity for the system.

All the models, enclosing these elements, can be found online at [29].

The collected information gathered with the questionnaire proposed to former users of the DMGame tool was used as input to the two-hours plenary session, which started with the step concerning the characterization of the personas, and was followed by the analysis of the users' needs and psychological strategies suggested by the method. Each step is performed separately by the two team pairs, but results can be mutually shared under the conductor guidance. In the following step, the case study participants described a typical working day of a DMGame user, showing the positive and negative feelings related to each activity during the day. On the basis of the insights from these initial steps the team-pairs started thinking about a possible re-design of the system and how to gamify it. Furthermore, this ideation step was supported by suggestions about useful gamification elements to be exploited, based on the users' characteristics. In the prototyping step, each pair consolidated design decisions on paper prototypes.

In the last step, participants showed their prototypes and presented their gamification solution, produced by using the DTA method, to the other participants. Once the case study ended, we asked participants to fill in a questionnaire to evaluate the DTA method from the requirements analyst's perspective.

Case Study Results. The analysis of the results is based both on the post-session questionnaire completed by participants (Table I), and on the examination of the actual work produced by them. Generally, all participants agreed that the users' needs and psychological strategies proposed by DTA were clear. The same result has been achieved with the gamification elements. Regarding the appropriateness of needs and psychological

strategies for the software and the target users, the participants agreed on their appropriateness. Also regarding the appropriateness of the gamification elements for the software and the target users, all the participants agreed on that. One of them strongly agreed that the gamification elements proposed by DTA were appropriate for the target users (Table I). For all these reasons, we can consider the clearness and appropriateness of the concepts proposed by DTA positive.

To verify the usefulness and the real usage of the concepts proposed by DTA, we need to look in more detail to the work done by participants. Most of the participants considered the users' needs and psychological strategies proposed by DTA helpful in defining personas. Most of the needs and psychological strategies suggested by DTA have been used by the participants, however not all of them. A similar observation concerns the needs used in defining the 24-hours timeline. This could be explained taking into account that DTA proposes needs that can be suitable for the different kinds of users, then it is up to the analysts to decide, also on the basis of the other kinds of software requirements and functionalities, which ones they prefer to consider for the specific needs of the stakeholders in their particular domain, in this case the one of the DMGame. The same applies for the suggested gamification elements. On the other hand, all the participants agreed about the usefulness of the proposed gamification elements, in a way that facilitates and enriches the design of the gamification solution (Table I).

B. User Perspective Case Study

Case Study Design. This second case study aims at evaluating the solutions, which have been designed in the case study described in Section VI-A. Precisely, in this case study we aim at evaluating the mockup that we developed on the basis of the design outcome of DTA, which was produced on paper. The interactive implementation is available online⁹. This mockup provides the specifications for a new version of the DMGame. It includes three main pages supported by few other pages. A screenshot of the main dashboard is depicted in Fig. 3, which includes most of the gamification elements that were proposed in the design of the new DMGame version. For instance, a *progress bar* indicates the progress of the user (the player) on an assigned prioritization task (see for instance button *Game 1*), also using a colour code. A *leaderboard* is depicted at the right-hand side, where the score of the user is shown in a ranked list of participants to the prioritization tasks (i.e. the active players). This score is computed taking into account points gained by the user as result of her contribution to the ongoing collaborative processes, including those depicted in the windows *chat* and *pinboard* respectively. The first one is a gamified *chat*, which rewards the users with points for communicating with other users, the second one is a *pinboard* where the user sees questions, answers and decisions shared among users. Details on the points gained for each different activity are depicted in the window labelled *total points*, which is placed below the *leaderboard* window.

⁹Available at <https://u5dzh2.axshare.com/home.html>

TABLE I
 QUESTIONS ON CLARNESS, USEFULNESS AND APPROPRIATENESS OF USERS' NEEDS, PSYCHOLOGICAL STRATEGIES AND GAMIFICATION ELEMENTS
 PROPOSED BY DTA.

N.	Questions	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
1	The users' needs and psychological strategies proposed by DTA were clear.	0	0	0	3	1
2	The users' needs and psychological strategies proposed by DTA were useful in defining personas.	0	0	1	3	0
3	The users' needs and psychological strategies proposed by DTA were appropriate for the DMGame.	0	0	0	4	0
4	The users' needs and psychological strategies proposed by DTA were appropriate for the target users.	0	0	0	4	0
5	The gamification elements proposed by DTA were clear.	0	0	0	3	1
6	The gamification elements proposed by DTA were useful in defining the gamification solution.	0	0	0	4	0
7	The gamification elements proposed by DTA were appropriate for the DMGame.	0	0	0	3	1
8	The gamification elements proposed by DTA were appropriate for the target users.	0	0	0	4	0

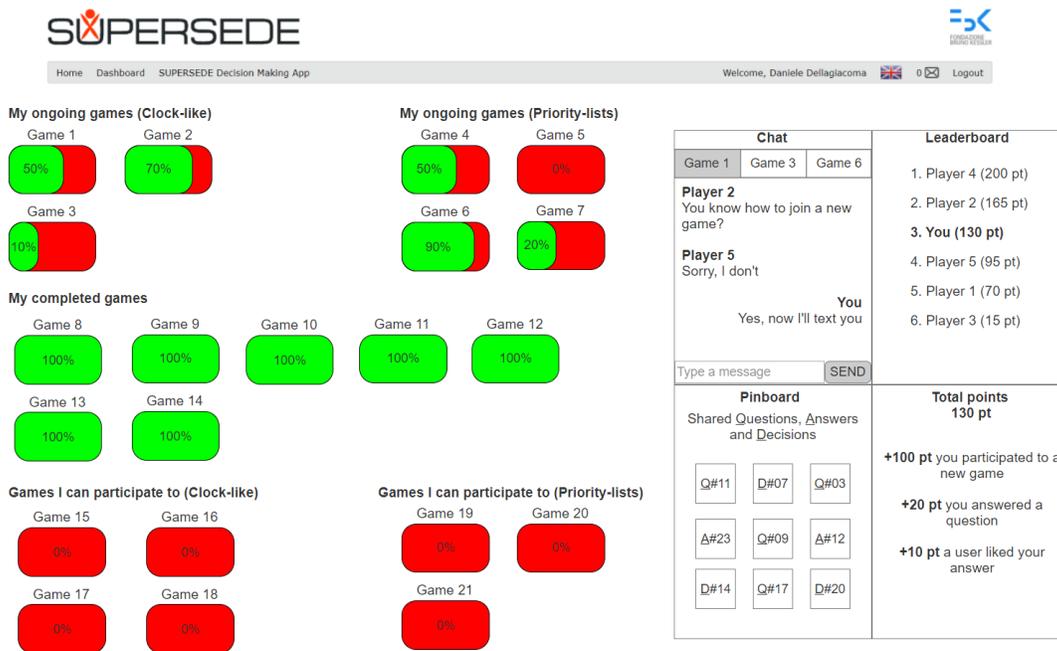


Fig. 3. Screenshot of the mockup generated by applying DTA: Main dashboard.

Other two GUIs are depicted in Fig. 4. The first one allows representing prioritised requirements according to a clock metaphor, where requirements to be considered first are moved (by “drag & drop” on the requirements box) in the first clock quadrant, and those that are less important in the last one. The second one allows to get an overview of the requirements to be prioritised for a given criterion, along three piles, namely high, medium, low priority piles. This allows the user to express a first coarse grained prioritization just moving a requirement from a pile to the one considered most appropriate. Within each pile requirements can be further ordered by moving them up or down, allowing the user to express a finer grain prioritization preference. Switching from one criterion to another allows the user to contribute to multi-criteria prioritization. The case depicted in Fig. 4 (right), shows three criteria, i.e. *User impact*, *Development cost*, and *Business benefit*.

In this study, we intend to involve people who are familiar

with the DMGame tool, and that have already used it. Therefore, they do not need to try again the old version of the tool, before starting the evaluation with the revised mockup.

To evaluate the revised DMGame we defined the tasks that the participant to the study have to perform via using the interactive mockup, and designed a questionnaire, to compare the previous version of the DMGame with the new version, which was obtained by using DTA. These tasks require the interaction with the gamification elements and the different prioritization techniques. The questionnaire is divided into different sections, each aiming at validating a specific feature of the mockup of the new DMGame version, improved by using DTA. The first section concerns whether the new software functions address the *needs* of the typical user of the DMGame, elicited in the previous case study, more than in the previous DMGame version. The second section aims at comparing the *engagement* offered by the

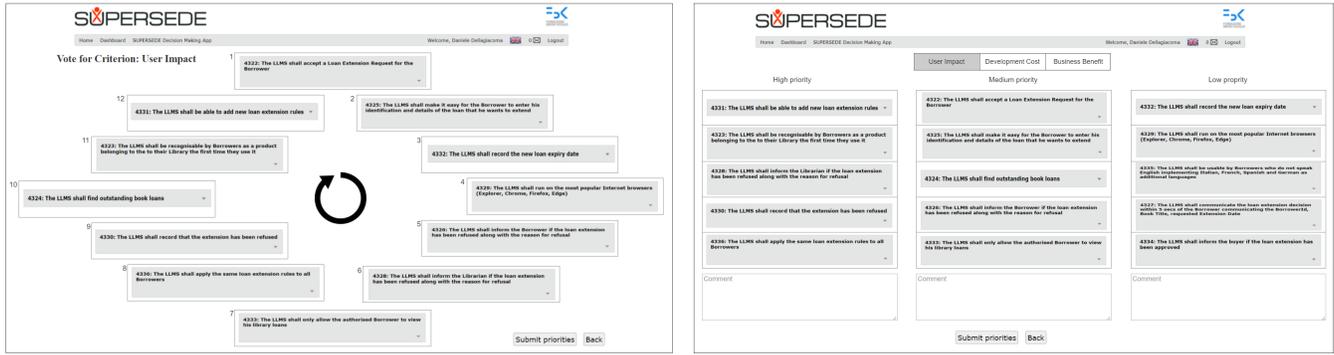


Fig. 4. Mockup generated by applying DTA: Clock-like requirement prioritization (left). Priority-list by criterion (right).

previous and the new version of the DMGame. In this case, the engagement is provided through gamification. The third section aims at comparing the *functions* of the previous and the new version of the DMGame. Specifically, the clock-like and priority-lists prioritization methods implemented in the new version of the tool. Finally, the last section includes open questions about strengths, weaknesses, and improvements of the revised DMGame.

Case Study Execution. Seven people agreed to participate to the study. They were provided with instructions to perform a few tasks with the interactive mockup, and then to fill in the questionnaire.

Case Study Results. Full charts and questionnaire replies can be found in [28]. The questionnaire begins with a description of the typical user of the DMGame. Then four questions to understand if his/her needs (elicited by Agon in the previous case study) are fulfilled by the new version more than the previous one. Here, participants agree that the new software functions address the needs of the typical user of the DMGame more than in the previous version. In detail, almost all the participants agree that the new version makes users perceive that the use of the DMGame can provide some benefits, more than the previous version. For instance, creating the perception that using the DMGame there are improvements in terms of their performance speed, quality, or effectiveness of their work.

Participants agree that the game elements implemented can engage users more than the previous version, and the engagement produced by these game elements can favour the quality and effectiveness of the requirements prioritization activity. Most of all, it seems that gamification can favour communication among users, and there are also positive results even concerning the collaboration among users that adopt the tool. Moreover, the new version shows an improvement in stimulating users in checking and improving the solution designed. The new version can also stimulate users in participating actively in the DMGame platform activities more than the previous version. Only one participant does not agree with this statement. To sum up, gamification has been strongly appreciated by participants. In fact, they think that the new version of the DMGame, improved by using DTA, represents a great improvement concerning the users' engagement compared to the previous version.

These results are also related to the two prioritization methods (i.e. clock-like and priority-lists), implemented in the new version, and to their usability, always focusing on the comparison with the previous prioritization method. Although both solutions have been appreciated and considered better than the previous one, the priority-lists solution has been preferred to the clock-like, as can be seen in Table II. In fact, participants do not completely agree that the clock-like method makes the requirements prioritization activity easier and faster than the previous version (questions 1 and 2 of Table II). Slightly better are the organization of on-screen information, and the quality and the effectiveness of the requirements prioritization activity provided by the clock-like method compared to the previous version (questions 2 and 3). On the other hand, the priority-lists method undoubtedly makes the requirements prioritization activity easier and faster than the previous version according to participants, as shown in questions 1 and 2 of Table II. Even the organization of on-screen information, and the quality and the effectiveness of the requirements prioritization activity provided by priority-lists method received a positive reception (questions 3 and 4).

TABLE II
COMPARISON BETWEEN CLOCK-LIKE AND PRIORITY-LISTS SOLUTIONS.
AVERAGE ON A 5-POINT LIKERT SCALE.

N.	Questions	Clock-like	Priority-lists
1	The method makes the requirements prioritization activity easier compared to the previous version.	3.00	4.29
2	The method makes the requirements prioritization activity faster compared to the previous version.	3.00	3.86
3	The method provides a clearer organization of information on screen than the previous version.	3.43	4.14
4	The method improves quality and effectiveness of the requirements prioritization activity compared to the previous version.	3.29	3.86

In summary, participants considered the new version of the DMGame (improved by using DTA) better than the previous version, with respect to many aspects, and that it provides a greater engagement, thanks to the gamification elements introduced. On the other hand, two participants think that the

new version is still complex to be managed as it contains too many elements and information.

C. Threats to Validity

As described before, the results of the case studies are positive, but it is worth to mention some threats to validity, besides the known limits about generalizability that are inherent to case study evaluation.

In both the case studies, a threat is the absence of multiple iterations. Although the first iteration of DTA produced a satisfactory result, multiple iterations are a core part of the DT approach. In fact, case studies should be followed, at least, by another iteration to improve the result. Furthermore, the DTA method we are proposing is suitable for an iterative approach, and we suggest to use it in such way to have more accurate results coming from a very deep analysis supported and favoured by DTA.

Another threat is related to the number of participants. Although our results are very positive and participants' opinions are homogeneous, it could be beneficial to repeat the case study with more participants, even from different organizations, to further confirm our positive results. This is one of our future works.

Finally, DTA provides a method to design gamification solutions, but the final result can be influenced by many factors such as the skills of the designers. Moreover, the version tested by users is a (interactive) mockup. Only the most representative functionalities, for showing to users the potential new features, have been implemented. This has been enough to get relevant feedback from them. However, as a future work, we will repeat the case study with a completely implemented version.

VII. RELATED WORK

Research literature in software engineering includes works on techniques for gamification design, as well as on the application of gamification to enhance software engineering methods, e.g. within requirements engineering tasks [30], [31], or to improve team engagement in a software development process [32]. An overview of the application of gamification in software engineering is reported in the mapping study presented in [6]. The rest of this section focuses on works closer to the context of this paper, i.e. works proposing gamification design approaches.

The MDA [33] approach is a formal and iterative approach to game design. It was developed in game design workshops and consists of three components: mechanics, dynamics, aesthetics. The aim of the MDA framework is to “bridge the gap between game design and development, game criticism, and technical game research”. Although the authors present reasonable examples, it is unclear how the framework was developed, and key requirements such as **R1**, **R2**, and **R6** seem to be ignored. Similar gaps seem to affect another approach that proposes to use gamification patterns, which were derived from the analysis of several gamified systems [34]. The authors conducted a case study, but did not describe how design patterns were integrated and applied to the development process.

Deterding [35] introduces a gameful design method based on four steps where the designer uses skill atoms and design lenses. The method was applied in two case studies, but as stated by Deterding, “it lacks a formal empirical evaluation of its utility and usability” [35]. Although their work adopts a multidisciplinary approach, Deterding does not consider existing requirements engineering approaches. In contrast, concepts such as acceptance requirements are key in our approach.

Model-driven approaches for developing gamified solution have been proposed in [36], [37]. For instance, the framework described in [37] provides a graphical modelling tool that supports a gamification expert to design a gamification strategy. For example, in the education domain, a strategy could be: “A student who watches the introductory video gets 5 basic points”. The framework provides also functionalities for automatic code generation, and real time monitoring of gamification strategies. Both these model-driven approaches focus actually on the design and implementation of a gamification strategy, which is defined by a domain expert, while in our proposal we take a requirements engineering perspective and focus on eliciting and analysing user needs and acceptance requirements first.

In [38] a situational method engineering framework for gamification design is presented. This framework allows to define the design approach to be used in a specific project by assembling method fragments, which are available in a method base that has been built by analysing scientific literature, descriptions of practical applications, and expert interviews. The framework provides guidelines for the selection and combination of method fragments, which rest on a set of design principles for engineering gamified software, and on contextual characteristics of the software to be gamified, such as business processes, corporate culture, and technological constraints. Moreover, the framework suggests to continuously monitor the effectiveness of the gamified application, after deployment, in order to evolve it according to the changing user needs. The framework has been evaluated by a group of experts and by applying it to an app for parking information that is based on crowdsourcing. While we can see interesting similarities and potential synergies between the design principles proposed in [38], and the requirements of the method presented in Section III, we propose a concrete approach, the DTA method, which combines two methods that together fit with the identified requirements. In particular, in our approach, Agon allows to capture acceptance requirements, and in combination with DT cover the notion of context as described in [38].

The analysis of these works provides further evidences about the practical relevance of the discussed lessons learned and requirements of our method.

VIII. CONCLUSIONS

In this paper, we discussed lessons learned in developing gamified software applications, in light of challenges and open problems on the design and evaluation of gamified solutions, which are presented in several recent research works (e.g., [5], [6], [15], [32], [38], [39]). Indeed, the analysis of these works

