CALDET: A TRIZ-Driven Integrated Software Development Methodology

Stelian Brad[1(✉)], Emilia Brad[1], and Daniel Homorodean[2]

[1] Technical University of Cluj-Napoca, Memorandumului 28,
400441 Cluj-Napoca, Romania
stelian.brad@staff.utcluj.ro

[2] Arxia, Tipografiei 28, 400101 Cluj-Napoca, Romania

Abstract. Companies specialized in software development services and software outsourcing have adopted agile methodologies such as SCRUM and Kanban to drive the software development process. Along time, experts have introduced new methodologies that enhance agile roadmaps with lean patterns and design thinking tools. Despite the value added brought by aggregation of agile with lean and design thinking, the integrated methodology is still fuzzy at operational level. There is no scientific demonstration on the most appropriate way to alternate the steps of agile with those promoted by lean innovation and design thinking. This niche of opportunity is investigated by this paper. Conflicts and barriers generated by aggregation are treated with TRIZ. Inventive solutions are proposed to optimize the agile-lean-design thinking (ALDET) software development process considering a life-cycle perspective. Gaps identified in ALDET are additionally tackled with TRIZ and new tools of competitive engineering, including TRIZ contradiction matrix for software, are embedded within the optimized ALDET to enhance its potentiality. The methodology is called Competitive ALDET or CALDET. Its effectiveness was tested in a real project. Preliminary results demonstrate that CALDET provides a clearer and smoother path for project management, reduces ambiguities relative to traditional ALDET methodology, and increases the impact of outcomes to the user. By tackling conflicts in an iterative manner, CALDET avoids re-analysis and re-coding in software development, too. The presence of value engineering tools within the framework of CALDET reveals additional spaces of innovation, both technical and project management related.

Keywords: TRIZ · Conflict · Software development · Competitive engineering · Lean innovation · SCRUM · Agile · Design thinking · Qualitative optimization

1 Introduction

Software products and systems are characterized by complexity and invisibility [1]. These characteristics induce significant challenges for designers and especially for users to articulate how the system should look like from the early stages of conceptualization and development [1]; in contrast, for example, with a house project. Therefore, in software development, waterfall project management methodologies [2]

Published by Springer Nature Switzerland AG 2019
R. Benmoussa et al. (Eds.): TFC 2019, IFIP AICT 572, pp. 400–416, 2019.
https://doi.org/10.1007/978-3-030-32497-1_32

have been replaced in the last 15 years by more adaptive approaches, based on the paradigm “prototype-test-evaluate-learn-refine” (PTELR); the spiral model [2]. PTELR approach was actually encompassed into a new philosophy of software development some 20 years ago, under a set of 14 principles called “agile manifesto” [3]. These principles have been then deployed by experts into some practical project management methodologies, known as agile methodologies [1, 3, 4]. They incorporate frameworks such as SCRUM and Kanban [1, 4, 5] to assist developers in responding to uncertainty and unpredictability in building up new software systems.

In SCRUM, software systems are developed in small shippable increments that can be assessed by users very early in the development process, and all necessary adjustments, improvements and new ideas can be thus incorporated into the system in due time [6]. Every incrementing stage is called “sprint” and lasts about 2–4 weeks, with reviews at every 24 h, where both development team and system owner are engaged [6]. The schematics of this process is shown in Fig. 1. Kanban, another agile methodology, differs from SCRUM in several aspects such as: changes are accepted any time, working cycles follow the workflow, it does not use a backlog pool and it does not include roles, etc. [1].

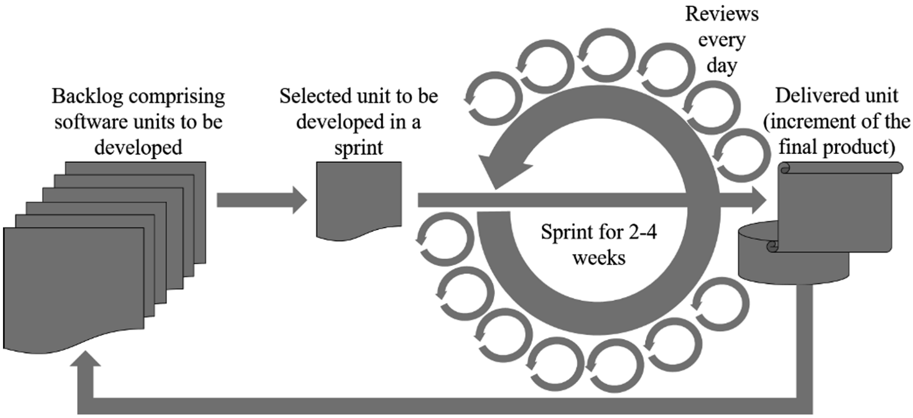
Backlog comprising software units to be developed
Selected unit to be developed in a sprint
Reviews every day
Delivered unit (increment of the final product)
Sprint for 2-4 weeks

Fig. 1. SCRUM framework in agile software project management.

Based on the same principles as agile methodologies (i.e. uncertainty and fuzziness in defining solution in the early stages, continuous discovery by prototyping and testing, steering implication of users in co-creation and feedback), new technology driven innovation adopted a lean approach in the last decade [2]. In lean innovation, every phase of evolution involves a spiral model of gradual prototyping and pivoting of solutions by testing hypotheses and adjusting them from the feedback received from the end beneficiaries [7, 8]. Thus, the main phases of lean innovation - meaning insight, problem understanding, solution formulation, and business model design - are tackled in a progressive way, with gradual prototyping and pivoting, with rapid learning and refining of solution, such as to avoid perpetuation of non-value added issues. One might see that between agile development and lean innovation is a strong synchronism.

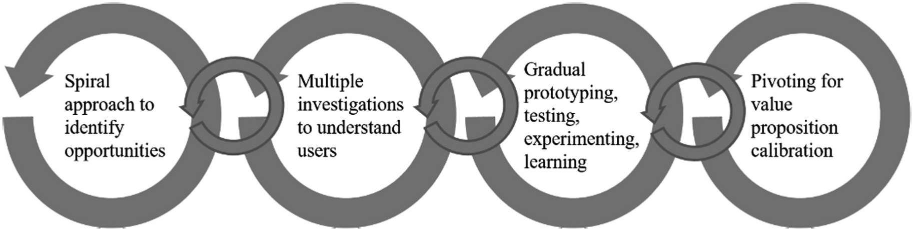
Spiral approach to identify opportunities
Multiple investigations to understand users
Gradual prototyping, testing, experimenting, learning
Pivoting for value proposition calibration

Fig. 2. Lean innovation methodology.

The graphical visualization of lean innovation is shown in Fig. 2. In addition, a third paradigm in new product development has gained in popularity in the last decade; that is, design thinking [9, 10]. This design model emphasizes on close collaboration with the end users during the development phases of a new product, starting with the process of need investigation, continuing with a deep understanding of users from multiple perspectives (roles), and then following an incremental process of solution formulation by looping ideas and prototypes in close collaboration with beneficiaries of solution [10, 11]. From the perspective of gradual prototyping and deep involvement of users in the design and development process, design thinking paradigm has many philosophical commonalities with agile development and lean innovation. An illustration of the main phases in design thinking model is shown in Fig. 3.

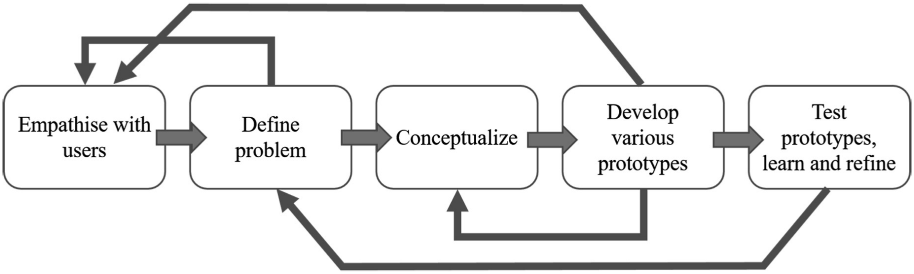

Fig. 3. Design thinking methodology.

Some researches have seen in the conceptual similarities of the three paradigms (agile development, lean innovation and design thinking) an opportunity to investigate the value added by combining them into an aggregated format. Hybridization of more frameworks is not a new approach in engineering. However, besides the benefits of bringing the strengths of each individual model, hybridization might come with some drawbacks, too, such as fuzziness of integration, more steps and higher complexity. Therefore, an important aspect of hybridization is to significantly incline the balance in the favor of benefits with respect to sacrifices. The second aspect is to refine the hybridized solution such as all conflicts brought by aggregation to be eliminated. The third aspect is to define a smooth framework from the combination of the individual concepts; otherwise, the adoption of the new model would not happen. The fourth aspect is to identify potential gaps or drawbacks in both the individual and the aggregated models and to complement them with new concepts (note: if they are available or possible to be discovered).

Thus, this paper continues with a synthesis of existent research contributions in combining the three concepts: agile, lean and design thinking. Value added and limitations of current results are highlighted in the background section. The third section of this paper introduces a roadmap for systematic analysis of the three concepts in order to discover areas with potential to set up an improved model of integration. Based on the conclusions obtained from roadmap application, a new model that integrates agile with lean and design thinking is formulated. Besides integration, the model is enriched with new tools for better handling product development, considering the necessity to tackle conflicts and to master value (i.e. the ration between impact and effort). The paper ends with findings from the application of the new model in a software project and with conclusions.

2 Background

In order to investigate the state-of-the-art on integrating lean, agile and design thinking methodologies in software development, several databases have been consulted: Web of Science, Scopus, Springer Link, IEEE Explorer, and Emerald. In addition, Google Scholar has been consulted. Searching process included combinations of "agile AND lean AND design thinking", as well as "TRIZ AND agile", "TRIZ AND lean AND agile", "TRIZ AND design thinking AND agile", as well as all these combinations with the additional keywords "software" and "project". After cleaning up information, the relevant papers selected for deeper investigation are introduced in the section "References". An important conclusion is that the majority of scientific papers on this topic are published in proceedings of international conferences or books with selected papers from international conferences. Another conclusion is that researches about combining lean and agile, or agile with design thinking are relative recent (not more than 8–9 years), whereas researches about the combination of the three concepts are quite recent published (in the last 2 years). Maybe, this is also the reason that only very few papers on this topic are present in the cited databases. Explicitly, only two research groups have published methodologies that integrate all three concepts [12–14].

Another important ascertainment in relation to the state-of-the-art is the fact that inventive problem solving and TRIZ are taken into account to improve the agile methodology in software development. Recent researches are reported in this respect, with representative results published in [15–17]. In closing the remarks on the searching process about the state-of-the-art relative to the subject under consideration in this paper, it makes sense to highlight the publication of researches that reflect integration of inventiveness in requirements engineering [18]. This element indicates that the early stages of either agile, lean or design thinking methodologies would benefit from structured innovation methods and tools.

Because various aspects of the combination between agile and lean, or lean and design thinking, or agile and design thinking have been analyzed also in relation with the integration of all three into a hybrid concept, it makes sense for the purpose of this paper to elaborate more on the findings currently obtained about the combination "agile-lean-design thinking" as a methodological approach to tackle software projects. The first group of findings comes up from the works [12] and [13], belonging to the same researchers who developed the methodology InnoDev that combines the three paradigms. Their analyses show that an integration of SCRUM practices with design thinking flow and lean innovation could be favorable. Their conclusions are actually based on perceptions of practitioners, as these have been extracted from a representative market survey. However, as the authors of this research highlight, the InnoDev model is only developed at the conceptual stage, with no validation in some projects. Thus, the current researches from papers [12, 13] cannot precisely articulate where and how the integration of agile, lean and design thinking should effectively happen at operational level, in which points methods, tactics and techniques are actually involved in the methodological flow. Nevertheless, besides the value added brought by these researches with the findings from market surveys, works from [12] and [13] indicate how the integration of agile, lean and design thinking would look like. It is a three-phase model, as indicated in Fig. 4. InnoDev introduces design thinking tools in an ad-hoc manner where blockers occur in relation to product development. Sprint and backlog concepts from agile/SCRUM are used in all project phases to plan and tackle activities, in order to provide transparency and to move forward while staying flexible to change requests.

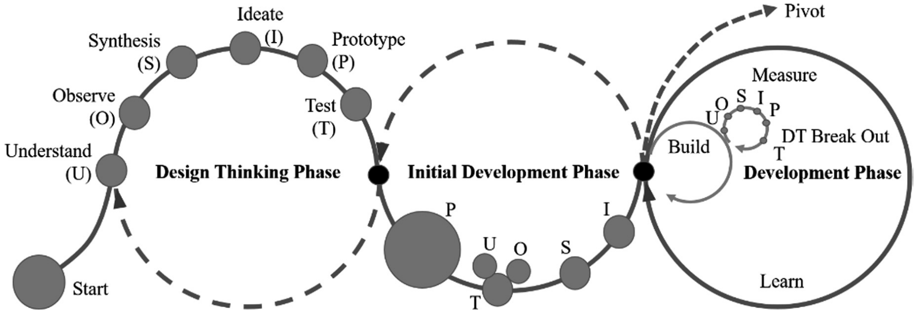

Fig. 4. InnoDev model for lean-agile-design thinking integration in software development (adapted from [12]).

According to InnoDev model, in the "Design Thinking Phase" only the vision to the problem is defined. The refined vision and the proof-of-concept (or minimum viable product (MVP)) realization are encompassed in the "Initial Development Phase" [12, 13]. In this second phase, concepts of user interfaces are designed and tested, technology stacks analysed and tested and the most important features are implemented in order to ensure a tangibility of the concept (e.g. viability, desirability and feasibility) [12]. In the "Development Phase" MVP is tested and continuously expanded towards the final solution, following the cycle "measure-learn-refine" or even pivoting [13]. By investigating the InnoDev model with over 60 SCRUM experts during 6 workshops, our conclusion is this model is still confusing, because in the vast majority of cases during the mentioned workshops, the impression was that the model can be reduced to the last circle from Fig. 4. This means that the first two phases of the model are seen only stages in the agile model, where design thinking tools might be considered within the sprints of the agile methodology. Moreover, the InnoDev model was compared by the consulted SCRUM masters with the model proposed by Gartner [19], which is illustrated in Fig. 5. Experts see the Gartner model and InnoDev model identical with respect to what it is called "customer problem definition" (phase I in InnoDev), but still, the lean startup phase is not well articulated in both InnoDev and Gartner models, as long as it simply looks like a depiction (for visualization purposes) of a sprint within the agile methodology.

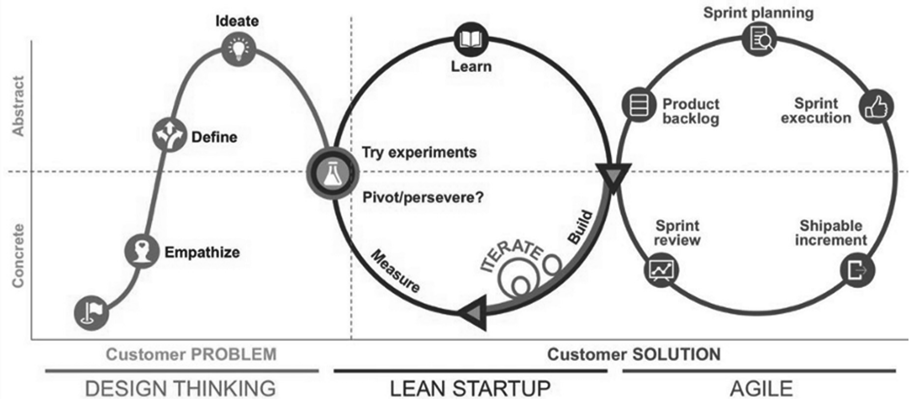

Fig. 5. The Gartner model of the combination agile-lean-design thinking paradigm (source [19]).

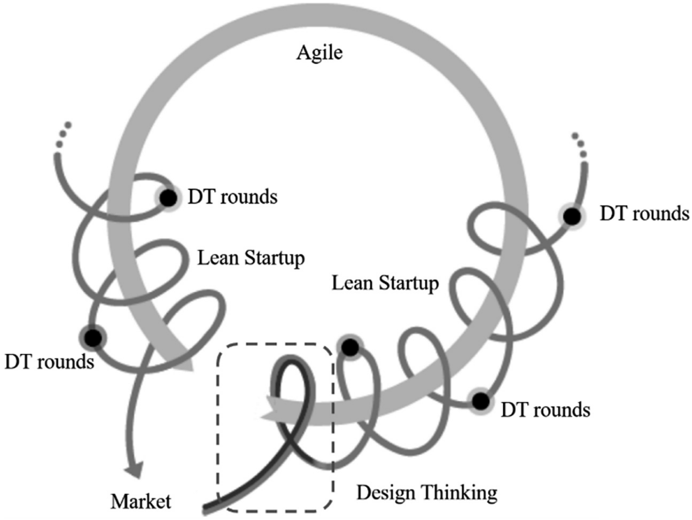

Fig. 6. The Converge model (adapted from [14]).

Another model that combines the three concepts (agile, lean, design thinking) is proposed in the work [14], under the name Converge. The authors indicate that this model is based only on empirical observations, and the combination of the three paradigms was experimented with a group of students over 8 weeks-project lab to develop a homonymous data storage app [14]. This research suggests that combination of the three concepts is productive in comparison with traditional software project management methods, but it cannot precisely indicate where, and why, this combination brings better results. The graphical representation of the Converge model is shown in Fig. 6. The quality of this model stands in the fact that it better visualizes the lean innovation as part of the sprints within the agile methodology. In terms of design thinking, the Converge model does not differ from InnoDev or Gartner models, because, as in those cases, design thinking is seen as the initial phase for project inception and included, where necessary, within the lean innovation loops.

However, the challenge with respect to all three models analyzed in this paper is around the capacity to master the lean concepts within the agile procedures. Lean is also about avoiding waste generation, not only about avoiding waste perpetuation and accumulation. Here, two important paradigms are seen important and not yet explored: (a) structured innovation (TRIZ and ideality), and (b) competitive engineering (value engineering, value analysis and value innovation). Researches of the authors of this paper on refining the agile-lean-design thinking model, as well as on improving its effectiveness by integrating the two additional paradigms in the model, are introduced in the subsequent sections of this paper.

3 Research Methodology

In order to improve the practical value of current models that combine agile with lean and design thinking, the first step in the research roadmap was to tackle the issue of fuzziness of the aggregated models proposed in the literature. In this respect, the first principle of the SAVE method was considered [20]; that is, the resonance principle. According to this principle, the goal is to harmonize the three concepts. To solve this problem, the following generic research approach was revealed: (a) do not consider combination of concepts, but rather creation of a new concept (model) that takes the best of the individual concepts and organize them into a value generation stream (b) depict all the three concepts (agile, lean, design thinking) along the whole life-cycle of a project in order to visualize them at operational level and see where they superpose (are similar), complement each other and where they are in conflict; (c) for the cases of superposition, keep only one of the three concepts, for the cases of complementarity add to the new model those practices, and for the cases of antagonism use inventive problem solving tools to fix the problem (e.g. TRIZ contradiction matrix [21]).

In order to best organize the modules generated from the first step of the research methodology, AIDA method is considered [22]. It can quantify the best arrangement from a set of possible (logical and viable) arrangements of modules. The next step is to investigate the selected arrangement (called ALDET: agile-lean-design thinking) for discovering gaps in terms of effectiveness and efficiency. The reference for gap identification is the paradigm of ideality from TRIZ [21]. In the case of software systems, ideality is related to complexity. The less complex the systems, the closer they are to ideality.

The ideal architecture for a software system is the one that includes completely independent elements (modules), where each element performs a single function. In such an ideal architecture/design the system complexity is minimized. Thus, from the ALDET perspective, the goal is to avoid generation and accumulation of complexity in both project life-cycle and product solution. To solve the conflict between dynamicity and complexity, TRIZ suggests application of asymmetry by more pivoting, concurrent development, fast-preplanning, and discard and recover resources.

To master value in system development from the perspective of lean philosophy, within the research methodology we opted for considering competitive engineering paradigm. This paradigm embraces things from concurrent engineering, lean six sigma, design for six sigma, and quality function deployment and it can be synthesized under the framework from Fig. 7.

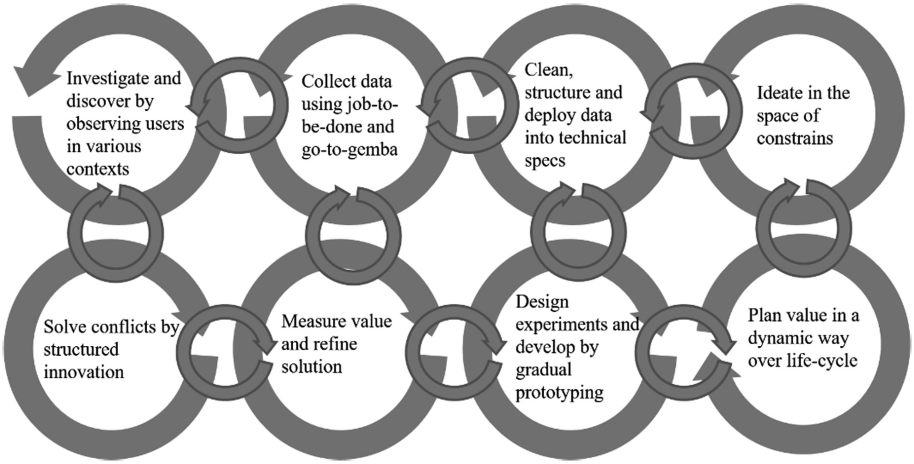
Investigate and discover by observing users in various contexts
Collect data using job-to-be-done and go-to-gemba
Clean, structure and deploy data into technical specs
Ideate in the space of constrains
Solve conflicts by structured innovation
Measure value and refine solution
Design experiments and develop by gradual prototyping
Plan value in a dynamic way over life-cycle

Fig. 7. Competitive engineering methodology (authors' synthesis).

By means of the same approach indicated in the first step of the methodology, competitive engineering can be captured in the ALDET model. The result is called in this paper CALDET (note: Competitive ALDET). The next section of this paper highlights the main findings from the application of the proposed research methodology.

4 ALDET and CALDET Models

The three methodologies have been depicted into smaller steps within the main phases, according to the recommendations of the proposed research roadmap. By analyzing similarities and differences, the key finding in this stage of investigation was that the three paradigms do not actually differ too much in practice and in philosophy, excepting the fact that each paradigm uses a different language because each of them has a different source of origin. Agile sprang from the world of software industry, lean has the source of inspiration in lean production and uncertain market management, whereas design thinking comes up from the world of designers and creative industry. Thus, this stage of the research led to a set of information that are summarized in the following Table 1.

Table 1. A view of agile, lean and design thinking by superposing their operational layers.

	Design thinking	Lean innovation	Agile management
Similarity	Iterative prototyping Periodic testing Measure results Learn and refine Empathize with users Define problem Co-creation Ideation Feedback loops Develop	Progressive prototyping Phase-gate testing Measure results Learn and refine Identify opportunities Understand users Co-creation Conceptualization (No explicit step) Develop	Sprint prototyping Sprint-framed testing Measure results Learn and refine (No explicit step) Backlog initiation Product owner Brainstorming Backlog updating Build
Complementarity	Jump-back loops	Pivoting	Sprint framework
Conflictuality	Deeper user planning Choose and focus	Early user mapping Concurrent design	Rapid solutions Late change requests

As Table 1 indicates, integration of the three paradigms requires either to select one path of the three in the conflicting areas or to solve conflicts in an inventive way. In essence, the following conflicting problems occur:

- start fast with what you have at the first glance and then adjust during sprints vs. spending a bit more time in the very early stage for mapping and planning users
- fix the solution at a certain stage of progress for consolidation and avoidance of complexity vs. accept change requests even in the very late stages of the project

With the support of TRIZ contradiction matrix, the problem above can be formulated as: speed versus reliability and adaptability versus complexity. For the first case, TRIZ suggests the following generic measures: moderation in advance, change the concentration of state and/or flexibility, dispose something expensive with several inexpensive elements, and introduce a softer approach. For the second case, TRIZ indicates the following generic actions: reconfigurable construction, interchangeability, expansion and contraction, as well as introduction of softer approaches.

Generic measures/actions have been translated into the following practical solutions: (a) any project will start with a time-boxing session where a task-force with representative of all stakeholders in the project will accelerate the formulation of a robust vision and a critical mass of specs; (b) besides the every morning session of 15 min. in the sprint framework, a mid-week ideation & co-creation session is included (including a platform for idea management, too); (c) technical development should consider in coding and architecting the core principles of reconfigurability for every sprint-related shippable prototype, i.e. modularity, convertibility, customizability, interoperability, scalability, portability, decoupling and avoidance of software asymmetry; (d) various tools and sessions can be added or eliminated in the context (e.g. application of various paths and means for problem formulation, ideation, etc.).

With these completions added to the elements from Table 1, and selecting only one perspective for those elements that are similar in the three models, the set of stage-gate elements of the ALDET model comprises: (e1) empathize with users, (e2) backlog initiation driven by user understanding, (e3) co-creation, (e4) sprint prototyping, (e5) sprint testing, (e6) measure results, (e7) learn, (e8) refine, (e9) ideation, (e10) backlog updating from multiple loops, (e11) build, (e12) sprint, (e13) pivoting, (e14) jump-back loops, (e15) time-boxing session, (e16) daily sprint session, (e17) mid-week ideation/co-creation session, (e18) reconfigurability-driven design and coding, (e19) flexible selection of tools. For every element from the set of 19 elements that can have more options, AIDA causality analysis was taken into account [22]. For exemplification, the selection of the best option for the element "co-creation" is shown in Fig. 8. The analysis criteria, also called constrains, are for this case the followings: (a) avoid complexity; (b) increase productivity; (c) reduce waste; (d) increase convergence. The options for "co-creation" are: (o1) application only at the early stage; (o2) application once in each sprint; (o3) application each mid-week session; (o4) application in each testing phase; (o5) inclusion of o1 and o2; (o6) inclusion of o1, o2, o3 and o4.

Step 1: influence of constrains on options [1 - low, 2 -medium, 3-high]

	o1	o2	o3	o4	o5	o6
a	3	2	3	2	3	1
b	3	2	2	1	2	1
c	3	2	1	1	2	1
d	3	2	1	1	2	1

Step 2: influence of options on constrains

	a	b	c	d
o1	3	1	1	1
o2	2	1	2	2
o3	1	1	3	3
o4	2	1	2	2
o5	2	1	2	2
o6	1	1	1	1

Step 3: products of coefficients from step 1 and step 2

	o1	o2	o3	o4	o5	o6	Σ C/6
a	9	4	3	4	6	1	4,5
b	3	2	2	1	2	1	1,8333
c	3	4	3	2	4	1	2,8333
d	3	4	3	2	4	1	2,8333
Σ O/4	4,5	3,5	2,8	2,3	4	1	

Fig. 8. Exemplification of AIDA to identify the best option for the element "co-creation".

According to the results from Fig. 8, option (o1) “application of co-creation only at the early stage” is selected. The result is counter intuitive. This indicates the usefulness of applying structured methods of analysis in the conceptualization process of new methodologies. The same approach has been applied for the rest of the 18 elements. At the end of this process, the necessary information to formulate the ALDET model was revealed. The result is shown in Fig. 9.

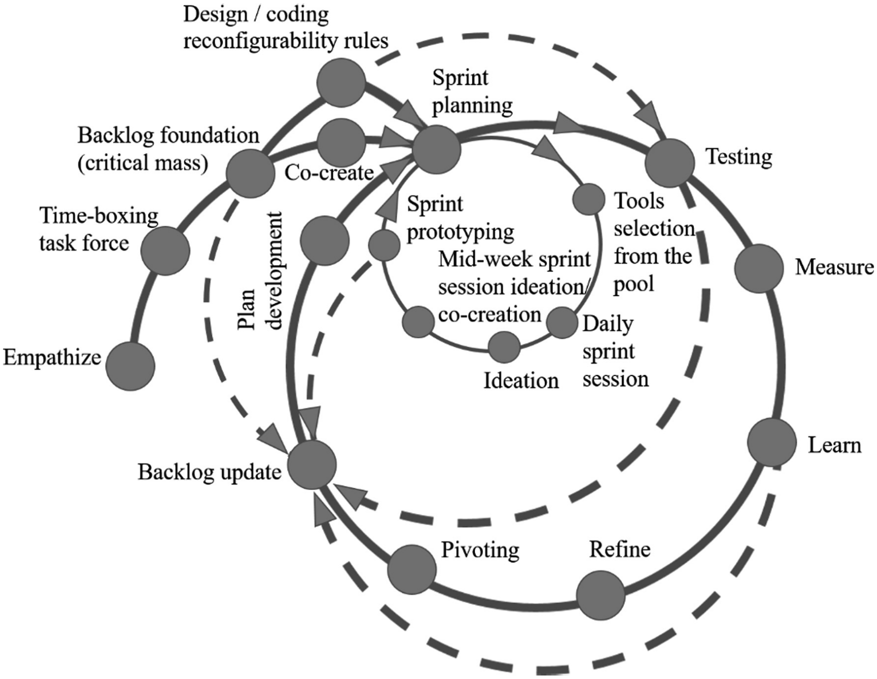

Fig. 9. The ALDET model.

ALDET model starts with the “empathize” phase in order to identify opportunities, and continues with a rapid development procedure (time-boxing session) to understand users and define a mature vision. Then, the backlog is fed with the critical mass of information. Before project starting, a co-creation hackathon is included. Simultaneously, the rules for reconfigurable design and coding is run. From this point, the first sprint is started. Each sprint comprises, besides the traditional elements, the continuous enhancement of the toolkit and selection of the most appropriate tools for the specific sprint, as well as the individual ideation and the mid-week session for collective ideation and co-creation. The iteration loop adds to the traditional agile process the stage of pivoting if lessons indicate this necessity. New to the model is the circuits of feedback loops (highlighted with dashed lines in Fig. 9).

The upgraded version of ALDET results at the intersection of ALDET with competitive engineering. The enhanced ALDET is called CALDET and its framework is illustrated in Fig. 10.

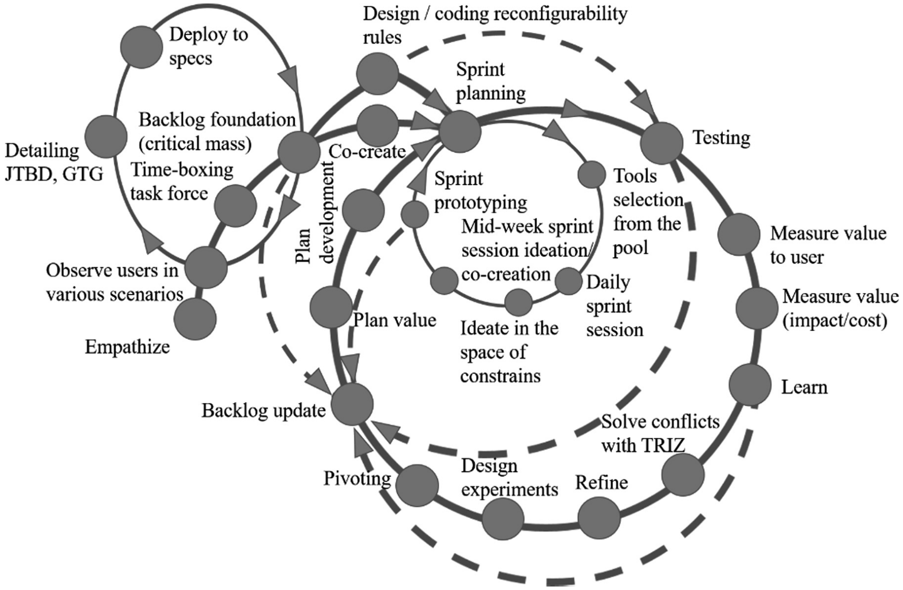
Deploy to specs
Design / coding reconfigurability rules
Sprint planning
Backlog foundation (critical mass)
Testing
Detailing JTBD, GTG
Co-create
Time-boxing task force
Sprint prototyping
Tools selection from the pool
Plan development
Mid-week sprint session ideation/ co-creation
Measure value to user
Observe users in various scenarios
Daily sprint session
Measure value (impact/cost)
Plan value
Ideate in the space of constrains
Empathize
Learn
Backlog update
Solve conflicts with TRIZ
Pivoting
Design experiments
Refine

Fig. 10. The CALDET model.

CALDET goes deeper to master value quantification and management, and to speed up convergence to a mature solution by considering constrains and conflicts for innovating in a systematic way during spring progression and iterations. In addition, CALDET considers a loop for deeper investigation of users, which is simultaneously run with the main loop of ALDET. To handle value, this paper recommends the use of value analysis matrices (e.g. QFD, relationship matrices, etc.). This aspect is not treated here.

The next section introduces the feedback collected from a team which applied the CALDET model in a software project. The focus of the following section is on team's experiences and impact in using CALDET, rather than on illustrating the results of the project, which is less relevant for the purpose of this research.

5 Experimentation and Discussions

In this section we describe the experience of employing CALDET in a project called InnDrive, dealing with the development of a software tool dedicated to start-up entrepreneurs, in order to evaluate the innovative potential of their business ideas, execution maturity and capacity of the their teams.

The initial definition of the scope for the software tool was based on a systematic assessment of the challenges confronting the start-up team members involved in early stage innovations, with a particular focus on innovative projects having an IT component. Engaging with early stage entrepreneurs and start-up team members, conducting interviews and structured assessments on the knowledge and readiness levels of the subjects, the innDrive team concluded that there is a clear need for a tool that can assist the start-up founders to evaluate and refine their understanding on all dimensions of the new business, identify the weak points in their plan and the aspects where they should focus, gain knowledge or put more work.

The entire project was strictly confined in the calendar and in the budget determined by a grant. Thus, the innDrive team had to handle the project in the given time and cost and to make ready for the market a product that is both useful and competitive.

According to CALDET methodology, observation of entrepreneurs in various scenarios was the first challenge. This process was conducted with contextual inquiry interviews. One-day workshop can generate a significant amount of information, with clear advantage on calibrating the task-force team and the key subjects to be discussed during the time-boxing session. Time-boxing session was organized within two meetings, each meeting of 4 h, with several inter-disciplinary working teams (developers, researchers, consultants, mentors, entrepreneurs) operating in parallel to extract needs, functional specifications and candidate technologies. It was proved that, by calibrating the time-boxing in this way, sufficient information have been collected such as to feed the backlog to a critical mass of elements for starting the sprint processes. Co-creation in this project was directed towards getting suggestions for GUI, how the help module to be designed and what content to be included. Here experts, consultants and end users collaborated in a face-to-face session, continued by a collaborative work on a web platform dedicated for such purposes.

In parallel with the time-boxing session and co-creation session, innDrive team experts conduced a systematic analysis of the problem using the Job-To-Be-Done (JTBD) method in order to reveal processes in start-up evolution, outcomes, which outcomes are underserved by current means and tools, and to extract relevant specs for feeding the backlog. It is important to highlight the fact that, by simultaneously feeding the backlog with information generated from two different streams (JTBD and time-boxing), is essential in CALDET (both for reciprocal confirmation of the work performed and reciprocal support with sources of information).

The refinement process of data generated with JTBD actually continued even after the first sprint started. This is a powerful element of CALDET, because it allows to update the backlog anytime, with quality inputs, keeping no huge pressure on the pace to which data and info are generated. Another powerful element in CALDET is the package of rules to design and code based on reconfigurability rules. This input can be actually done outside the scope of a given project, as a good practice in the company. Once the know-how is present, it can be instantaneously deployed to any new project. Nevertheless, CALDET suggests to revisit these rules in the early stage of any new project, just to see if not new elements would be necessary for every particular project. This work is concurrently done with the co-creation process, it is run by the technical stuff, and it uses inputs from the backlog foundation stage.

In this particular project, sprints were calibrated at a two-week cadence. Another key strength of CALDET revealed in the project was the mid-week sprint session for ideation and co-creation, where various representatives of all stakeholders have been involved. The key finding in this process was the fact that, not the same representatives of the stakeholders have to be involved over the whole project duration. By introducing new representatives, the project gets new perspectives and proofs the capacity of fast understating and adoption by the new comers. Thus, in this project, entrepreneurs from both local ecosystem (i.e. Cluj-Napoca city) and international ecosystems (e.g. EU and non-EU countries) have been involved in various sprints. In the particular case of the InnDrive project, major contributions from the stakeholders during the mid-week sessions, with visible effects on the project's speed, were on clarity and relevancy of the content embedded in the developed tool, GUI's usability and intuitiveness, as well as functional details of the platform.

Another key elements of CALDET, which indicated its effectiveness in the project, were the value mapping step and problem solving step. Both are activated to treat the results of every completed sprint, in parallel with the run of the next sprint. For value mapping, a QFD-based matrix was used, but in a dynamic way (meaning that inputs and outputs in the relationship matrix can be changed any time). Value mapping created alarms for developers to handle sprints in a more innovative way, especially in relation with the ideation stage within the sprint. Value mapping creates a pressure on developers to achieve a certain level of performance in a given, justified, budget. This pressure was actually proved to be the engine for the TRIZ sessions within the incremental loops of CALDET. For example, one issue was in relation with the quantity and type of info to be included in the help module related to each assessment criteria. TRIZ was applied in this case such as: reduce the "amount of substance" without affecting "capacity". TRIZ indications are: reconfigurable construction, make the module more movable, design a heterogeneous construction such as to increase the local quality in a dynamic way, and/or dispose parts which are not anymore relevant. With these indications, the system was designed such as the "help module" to start with a info-graphic schematics and then to be continuously enriched with data, info, examples relative to each issue/assessment criteria by any contributor in the platform, anytime, under a control for uploading content in the system. Inputs can be added into modules that can be afterwards interlinked, added, enhanced or eliminated. It works like a living organism.

Pivoting stage of each iteration proved to be another strength of CALDET methodology. For the first rounds of sprints, the innDrive team used the pivoting stage only with start-up entrepreneurs and experts in mind. Once the innDrive system approached the minimal viable product (MVP) state, engagement was directed towards start-ups outside the focus group, including various organizations from various countries involved in the evaluation, mentorship or financing of start-ups, even incubators, accelerators, venture capital funds, innovation program management units at national level, management authorities of public financing programs, innovation management units in the private corporations, bank managers responsible with the evaluation of credit requests targeted to support new business projects, evaluators of calls for grants.

This massive engagement revealed new patterns that determined innDrive to pivot, both in how it supports various types of users and in the business approach. The major discovery was that start-ups that were approached with a more mature application and were not involved in the development process lacked the discipline to conduct the complex assessments by themselves, although invariably they expressed initial enthusiasm. The novelty and level of focus required by conducting a self-assessment based on the platform determined a staggering level of abandon, of 70–80% of the subjects. Off the other actors in the ecosystem that were engaged, the strongest positive reaction was obtained from the program managers who are facilitating or managing start-up funding and mentorship programs and they were very eager to implement the innDrive system into their programs, once it is optimized to support the mentorship and coaching interactions in the application itself.

As result, the innDrive team had to redesign the functional workflow, introducing the roles of program managers and mentors/coaches, and giving to those roles the possibility to initiate, assign, and evaluate project assessments in relation with the entrepreneurs and to develop and follow action plans resulting from the reports generated from the assessments. At this level, CALDET proved to show another strength, meaning reconfigurability-driven design and coding style. It was easy in the forthcoming sprints to make the necessary modifications without a big effort and in due time.

The entire development of innDrive was finished in time and in budget, preserving the initial scope of assisting start-ups and project teams in assessing their capacity to sustain the project, while introducing a strong emphasize on the assistance roles and focusing the commercial focus of the innDrive project towards the managers of venture programs.

6 Conclusions

This paper introduces researches conducted to understand the state of scientific progress on integrating sectorial models for managing innovative software projects (agile, lean, DT) into hybrids, with the purpose to bring the capacity of handling complex projects to an upper level. Three models (InnoDev, Gartner and Converge) have been analyzed in this respect, concluding that many empty spaces are present in doing integration and thus, opportunities for further researches are envisaged. The key findings about the state-of-the-art were the fuzziness of integration on one side, and the incapacity to demonstrate the effectiveness of the integrated model (e.g. why to follow the proposed stream of actions in that order and not differently) on the other side. However, from the three studied models, our researches conclude that Converge is better equipped, even if some drawbacks are signaled in this case, too, such as the justification of fully application of design thinking in every loop of lean innovation. Our investigation on this issue with SCRUM masters indicates a rejection in practice rather than adoption, because of time consumption that affects sprints and because it is not clear how to, for example, apply all steps of design thinking in a late loop of lean innovation.

Reflecting on the current state of developments with TRIZ tools, this paper introduces a model called CALDET that claims advantages with respect to previous models because it avoids redundancies between the integrated concepts, orders the steps in a justified way (due to the use of quantitative tools to prove this) and cleans the roadmap from fuzziness because the model not just integrates agile, lean and design thinking, but rather embeds them into a new flow that comprehends the best of the three sectorial approaches. Beyond these aspects, the novelty of CALDET comes also from the inclusion of the fourth concept; that is, competitive engineering, offering on this way tools to measure value in the lean approach and to reveal the relevance of structured innovation in any iteration of the agile flow. Even if up to this date CALDET was tested only in a single project, it proves to be very effective, encouraging the research team to continue investigating the model. In this respect, future researches will be conducted to enrich the model with refined templates at each stage, and especially to deepen investigation on the stage referring to reconfigurability of code, because this will bring a huge advantage in agility, especially when pivoting is involved.

References

1. Zapata-Roldan, F.: Design capabilities in software innovation settings. In: Portland International Conference on Management of Engineering and Technology, pp. 1236–1243. IEEE, Portland (2017)
2. Fraser, D., Abrahamsson, P., Davies, R., Kerievsky, J., Poppendieck, M., Succi, G.: The future of lean in an agile world. In: International Conference Agile Processes in Software Engineering and Extreme Programming, pp. 263–266. Pula (2009)
3. Staron, M., Meding, W.: Monitoring bottlenecks in agile and lean software development projects – a method and its industrial use. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 3–16. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21843-9_3
4. Rodriquez, P., Markkula, J., Oivu, M., Garbajosa, J.: Analyzing the drivers of the combination of lean and agile in software development companies. In: International Conference on Product Focused Software Process Improvement, pp. 145–159. UPM, Madrid (2012)
5. Glomann, L.: Introducing 'Human-Centered Agile Workflow' (HCAW) – an agile conception and development process model. In: Ahram, T., Falcão, C. (eds.) AHFE 2017. AISC, vol. 607, pp. 646–655. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-60492-3_61
6. Laanti, M.: Characteristics and principles of scaled agile. In: Dingsøyr, T., Moe, N.B., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (eds.) XP 2014. LNBIP, vol. 199, pp. 9–20. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14358-3_2
7. Halas, M.: Lessons for TRIZ from design thinking and lean 3P. In: Koziołek, S., Chechurin, L., Collan, M. (eds.) Advances and Impacts of the Theory of Inventive Problem Solving, pp. 159–168. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96532-1_15
8. Elberzhager, F., Holl, K., Karn, B., Immich, T.: Rapid lean UX development through user feedback revelation. In: Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (eds.) PROFES 2017. LNCS, vol. 10611, pp. 535–542. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69926-4_43
9. Canedo, E., Parente da Costa, R.: The use of design thinking in agile software requirements survey: a case study. In: International Conference of Design, User Experience, and Usability, pp. 642–657. UNB, Brasilia (2018)

10. Cockton, G.: Integrating both user-centered design and creative practices into agile development. In: Cockton, G., Lárusdóttir, M., Gregory, P., Cajander, Å. (eds.) Integrating User-Centred Design in Agile Development. HIS, pp. 249–276. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32165-3_11
11. Hildenbrand, T., Meyer, J.: Intertwining lean and design thinking: software product development from empathy to shipment. In: Maedche, A., Botzenhardt, A., Neer, L. (eds.) Software for People, pp. 217–237. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-31371-4_13
12. Dobrigkeit, F., Paula, D.: The best of three worlds - the creation of INNODEV a software development approach that integrates design thinking, SCRUM and lean startup. In: International Conference on Engineering Design, pp. 319–328. Design Soc., Vancouver (2017)
13. Dobrigkeit, F., Paula, D., Uflacker, M.: InnoDev: a software development methodology integrating design thinking, SCRUM and lean startup. In: Leifer, L., Meinel, C. (eds.) Design Thinking Research, pp. 199–227. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-319-97082-0_11
14. Ximenes, B., Alves, I., Araujo, C.: Software project management combining agile, lean startup and design thinking. In: International Conference of Design, User Experience, and Usability, pp. 356–367. UFPE, Recife (2015)
15. Casner, D., Souili, A., Houssin, R., Renaud, J.: Agile'TRIZ framework: towards the integration of TRIZ within the agile innovation methodology. In: Cavallucci, D., De Guio, R., Koziołek, S. (eds.) TFC 2018. IAICT, vol. 541, pp. 84–93. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02456-7_8
16. Ma, J., Zhang, Q., Wang, Y., Luo, T.: Construction of the dependence matrix based on the TRIZ contradiction matrix in OOD. In: Tan, R., Cao, G., León, N. (eds.) CAI 2009. IAICT, vol. 304, pp. 219–230. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03346-9_24
17. Fulbright, R.: Incorporating innovation into iterative software development using the inventive problem solving methodology. Int. J. Innov. Sci. **5**(4), 203–212 (2013)
18. Sa'adah, H., Azrina, K., Novia, A., Marzanah, J.: Towards developing an approach to confluence innovative in requirements engineering. In: Fujita, H., Selamat, A., Omatu, S. (eds.) New Trends in Intelligent Software Methodologies, Tools and Techniques, pp. 164–175. IOS Press, Amsterdam (2017)
19. Mantini, N.: Design thinking, lean startup and agile: what is the difference? https://medium.com/xplor8/design-thinking-lean-startup-and-agile-what-is-the-difference-1eed3594b121. Accessed 03 Apr 2019
20. Brad, S.: Structured Activation of Vertex Entropy (SAVE): another way around creative problem solving for non-technical applications. INNOVATOR J. Eur. TRIZ Assoc. **01/2017** (03), 76–81 (2017)
21. Petrov, V.: TRIZ – Theory of Inventive Problem Solving – Level 1. Springer, Heidelbeeg (2019)
22. Brad, S.: Improving the use of AIDA method. Acta Technica Napocensis, Ser.: Appl. Math. Mech. **50**(1), 45–48 (2007)