Usability Insights for Requirements Engineering Tools: A User Study with Practitioners in Aeronautics

Hélène Gaspard-Boulinc, Stéphane Conversy

HAL Id: hal-01532781

https://inria.hal.science/hal-01532781

Submitted on 5 Sep 2017

Usability Insights for Requirements Engineering Tools:
a User Study with Practitioners in Aeronautics

Hélène Gaspard-Boulinc

University of Toulouse - ENAC
Toulouse, France
helene.gaspard-boulinc@enac.fr

Stéphane Conversy

University of Toulouse – ENAC
Toulouse, France
stephane.conversy@enac.fr

Abstract— **Requirements Engineering plays a crucial role in coordinating the different stakeholders needed for safe aeronautics systems engineering. We conducted a qualitative study, using interviews and mockups, with fifteen industrial practitioners from four aeronautics companies, to investigate what tasks are actually performed by requirements engineers and how current tools support these tasks. We found that RE-specific tools constrain engineers to a rigid workflow, which is conflicting with the adaptive exploration of the problem. Engineers often start by using general-purpose tools to foster exploration and collaborative work with suppliers, at the expense of traceability. When engineers shift to requirements refinement and verification, they must use RE-specific tools to grant traceability. Then, the lack of tool usability yields significant time loss and dissatisfaction. Based on scenarios of observed RE practices and walkthrough, we formulate usability insights for RE-specific tools to conciliate flexibility and traceability throughout the RE process.**

Index Terms—**practices, industry, usability, tools, aeronautics.**

I. INTRODUCTION

During the lifetime of a complex system, changes in the system environment lead to the creation and modification of a significant number of requirements [42]. Requirement engineers must manage this continuous change of requirements and demonstrate traceability. The use of requirements engineering (RE) specific tools is recommended by standards to support these activities [27].

Poor usability of such tools is identified as a factor contributing to unpopularity of traceability [11][49][50]. Yet, the certification process in critical systems requires traceability documents as a means of demonstrating safety. Our qualitative research work aims at investigating how tools actually support engineers when handling requirements in the context of critical systems. The research questions are RQ-1) what tasks are actually performed by requirements engineers?, RQ-2) how current tools support these tasks?, and RQ-3) how tool usability can be improved for Requirements Engineering?

First, we analyze the prior work related to tools and practices in requirements engineering. We then present the qualitative research method we used to identify tasks and usability issues of the industrial practitioners in aeronautics. The resulting contribution is a set of crosscutting RE practices from which we draw usability insights for RE-specific tools.

II. RELATED WORK

We present prior work related to (A) surveys on Requirements Engineering tools, adopting a usability perspective, and (B) field studies in Requirements Engineering, adopting a tool perspective. We then discuss the gap between the two perspectives (C).

A. Surveys on RE Tools - Usability Perspective

As the number of RE tools is expanding [34] (see an example in Fig. 1.), the ISO organization provides a guide [29] to help people compare and assess them. The guide classifies the capacities to be met by RE-specific tools into categories derived from the procedural tasks defined by standards [27]: elicitation, analysis, specification, verification and validation, management, plus « other capabilities ». Usability concerns, classified in the last category, relate to a classic window manager, "multiple views", an "interactive graphical input and manipulation of data", a web browser interface, edit-undo-cut-paste and macros features. Nothing is said about data representations, or the goals to be achieved by the user through data manipulation.

Previously used by INCOSE [27], the guide is the framework followed by Carillo et al. [3] to assess the features of 38 tools. Even if the method is debatable since tool representatives have evaluated their own product through a questionnaire, the statistical analysis shows that users have an interest in graphical interface and data integration. Moreover, "flexible tracing" obtained a weak rating: tracing between different elements (e.g. text, graphics) is proposed as a tool improvement.
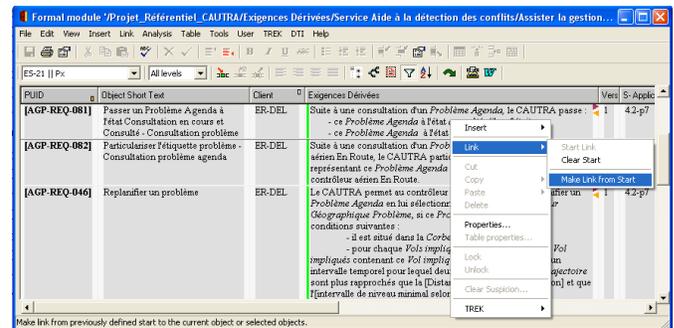

Fig. 1. A screenshot of the DOORS software, used to specify and trace requirements for complex systems

Carillo et al. revise the framework with new features (modelling, traceability and collaboration) and report updated tool information [5]. They find three clusters of RE tools with respect to RE capabilities. However, they argue that the qualities of the tools, such as usability, are difficult issues to address with the survey approach

Hoffmann et al. [24] propose a requirement catalog for requirement management tools. The authors use the ISO/IEC 9126 standard [30] as a framework to evaluate the qualities of tools based on their experience in automotive electronics. They identify "user-friendliness" as a key factor for traceability functions, but the related features are formulated as shallow guidelines such as "graphical views of the requirements should be available ", "all users must be able to customize the standard views without changing the template".

Cooper et al. [11] focus their survey on visualizations and propose a less process-centered definition of tasks to be performed. They evaluate visualizations from 29 research papers, using a visualization categorization. They note a prevalence of relational and hierarchical visualizations in the refinement and specification phases. The elaboration phase has the most varied types of visualization (sequential, tabular, relational, hierarchical). In contrast, little visualization is proposed to support verification and validation activities. They recommend developing visualization for verification and validation tasks, and promote the exploration of quantitative and metaphorical visualizations. They suggest that more effort is needed to understand the practitioners' tasks in which visualization is useful.

Niu et al. [38] identify the interactive visualization of requirements as an interesting technique for exploration, knowledge discovery and efficient reasoning. They apply a visual analytics-oriented framework to compare 8 interactive visualizations. The results show that multiple views are well supported while inter-view navigation, filtering, querying and annotating are not supported effectively. They propose a solution based on cluster visualization, but do not explore other visualization techniques. By conducting a systematic literature review on RE visualization, Abad et al. [1] show that there is no visualization support for the whole RE lifecycle and a need for more effort on storytelling, maps and interactive visualization.

B. Field Studies in RE practices - Tool Perspective

There are several field studies that concern RE practices and challenges until the nineties. Lubars et al. conducted a field study on requirements modeling [35] and found that many users resort to general-purpose tools (spreadsheets, word processors) instead of specific commercial tools for generation of prototype code, view of the model through multiple filters and simulation of a model. However, they do not provide explanations regarding this preference. Studying factors affecting completion of the requirements capture, Chatzoglou [7] found that tools are very important for the completion of the RE process in most projects (90 out of 103), among other factors such as experience, motivation, knowledge of the purpose of the project development, communication, and management style. The only recommendation about tools is to "pay greater attention to their quality".

More recently, the findings from several field studies conducted by Nguyen et al. [40][41] show that the RE process is creative and opportunistic, involving adaptive and responsive exploration of the problem space. They conclude that flexibility is needed for requirements engineers' creativity. Their work is focused on the use of a notation for design rationale and do not tackle how tools can support this creative process. Karlsonn et al. [32] report the lack of tools that support a flexible process as a problem for requirements engineers. The same survey reports difficulties of software development companies in managing the constant flow of new requirements, release planning and requirements traceability, inducing a large amount of effort. However, there are no details on which tools are used to handle prioritization and traceability. The survey conducted in 12 software companies by Hall et al. [21] identifies communication as a key problem in requirements engineering but argues that most requirements problems are organizational rather than technical. Sikora et al. [48] conducted a qualitative study with 7 large companies from five branches of the embedded systems domain: they found that models are regarded as a supportive means for analysis, but are not used in specification documents. Since the authors' focus is model-based research, they recommend providing industrial practitioners with a methodological guidance to foster the use of models, such as reference models and rules. Finally, an exploratory study based on 29 interviews with software practitioners [17] shows that current tools only provide primitive capabilities for handling concurrent and large requirements artifacts. The results suggest that improving the user interface of tools could lead to more efficient interaction with such artifacts. However, the study does not provide information on RE tasks in need of better interactive tool support.

C. Discussion

Surveys on RE tools mainly focus on *functional* features of tools by following the *procedural* tasks of the Requirements Engineering process defined by ISO standard (elicitation, analysis, specification, verification and validation, management). However, as shown in section B, field studies reveal that the the RE process does not have predefined stages but rather involves adaptive and responsive exploration of the problem space. The main identified challenges are communication, the fluctuating nature of requirements and difficulties when prioritizing the requirements for the next release. Tools are pointed as an important factor to consider, but none of these studies primarily focuses on how tools actually support users to face the identified challenges. Surveys on RE tools propose general usability guidelines, such as the availability of multiple views of the same data, but do not define the data, the type of view and the context of use. The ability to interact with visualization is presented as promising for knowledge discovery (anomaly detection, detailed explanation, efficient reasoning) but the benefits of information visualization are not yet fully explored. Finally, a better understanding of the tasks performed by industrial practitioners is frequently recommended [7][15][38] since "human factors" are designated as hindering the management and traceability of requirements [24][50].

III. METHOD

Our aim is to investigate how tools actually support engineers when handling requirements. We have adopted a qualitative approach since we wanted an in-depth understanding of software and systems engineering as a socio-technological activity [51][52]. We followed a Case Study Research methodology [52] to get richer and more contextualized information of the Requirements Engineering phenomenon, rooted in its natural settings. We describe in the next subsections the selection of data sites, the participants, the data collection and the data analysis.

A. The selection of Data Sites: Aeronautics

We chose to observe the phenomena of Requirements Engineering in aeronautics. Aeronautics is one of the most demanding domains with respect to requirements engineering due to its safety-critical nature and the life span of its products. The DO178 standard [12] provides the aeronautics industry with guidelines for demonstrating an acceptable level of safety: engineers must provide evidence of the links (or traces) between all requirements and their implementation as a key means of demonstration.

B. Study Participants

The study involved a total of fifteen requirements engineers, with a minimum of four years of experience, from four different companies: an aircraft manufacturer, an avionics manufacturer, an aircraft manufacturer sub-contractor and an air traffic control system provider (see details in TABLE I.).

TABLE I. STUDY PARTICIPANTS

Company	Participants		
	Years of experience	Contextual interview	Design Walkthrough
Aircraft manufacturer (A) 2 participants	>10	*	
	4	*	
Aircraft manufacturer sub-contractor (B) 1 participant	4	*	
Avionics Manufacturer (C) 3 participants	>10	*	
	>10	*	
	>10	*	
Air Traffic Control system provider (D) 9 participants	>10	*	* *
	4	*	* *
	4	*	
	>10	*	
	>10		*
	>10		*
	>10		*
	>10		*
	>10		*

C. Data collection 1: Contextual Interviews

We started with contextual interviews to understand the tasks currently performed by requirement engineers, and identify real life practitioners' problems. Contextual Interviews are one-on-one interviews conducted in the user's workspace that focus on observations of ongoing work [25]. We asked ques-

tions about what was going on in the work and flexibly reacted to the user's answers, investigating issues emerging during the interview. Our goal was to capture the actual RE activities in the workplaces, beyond self-reported practices or official processes (RQ-1). We interviewed ten engineers in their work environment. Each contextual interview lasted two hours. We video recorded them, except in the case of three interviews for confidentiality reasons. The interview focus was tool support in the performed RE tasks. We used the following questions on RE artifacts as a starting point [25]: "where did the information come from?", "how is the artifact created?", "how is it used?" and "who will get it next?". We wrote a transcription of each interview. We collected artifacts during the interviews and we used them while transcribing the interviews for a clearer understanding.

D. Data collection 2: mockups and walkthroughs

After eight interviews, we detected significant problems of data integration and visualization. In order to gain insights about these problems, we designed several mockups to benefit from varied visualization components, and conducted design-oriented research [14]: "the knowledge that comes from studying the designed artifact in use or from the process of bringing the product into being is the contribution, while the resulting artifact is considered more a means than an end". Hence, the goal was not to develop mockups as a final product, but rather as probes to gather new insights about tasks and usability through grounded discussions with participants.

1) Mockups

We designed and developed the mockups using the d3 JavaScript library [3]. First, as suggested by prior work, we explored hierarchical and quantitative visualizations, as well as interactions for filtering. Then, based on the extended use of matrices in the artifacts collected during the interviews, we explored alternative visualizations of matrix. Finally, we explored how linking between visualizations [1] can support inter-view navigation and traceability.

We refined visualizations and interaction throughout an iterative process, using paper prototypes and interactive mockups. A paper prototype is a representation of the product, built out of Postit® notes and various other pieces of paper [25]. Paper prototypes are movable and changeable in order to add or modify content in response to the user trying to use it. In addition to paper prototyping, we developed interactive mockups to address scalability problems, using requirements data that we had collected during the interviews. The resulting mockups are outlined in section "V- Usability Insights".

2) Walkthroughs

We ran nine sessions of 2-hour walkthroughs [25][36] with seven participants in their working environment. During walkthroughs, the partnership is one of co-design: as the user works with the prototype following a task they need to do or did in the recent past, the user and the researcher discover new problems. They interpret together what is going on in the probe usage and come up with alternative designs.

To this end, we fed the mockups with real-life data, including around 4000 requirements. For each designed mockup, we

presented the main features and collected qualitative participants' feedback. We also asked participants to show how they performed the discussed tasks with their current tools. A report of each walkthrough had been written, including feedback on mockups and discussions with participants. The participants validated the reports.

E. Data analysis: Scenarios, Critical Tasks, Usability Insights

We translated the raw data (interview transcripts, walkthrough reports and collected artifacts) into consistent scenarios, according to the involved users (requirement engineer, component supplier, RE team, managers), their goals, the tools in use and the related artifacts. Scenarios synthesize stories about users and their activities and are considered suitable to report grounded information [6][10][39]. This preparation of data allowed us to obtain comparable data and keep users actively engaged in the analysis [37]: three participants from A and C companies validated the scenarios. We then performed a step-by-step analysis of the scenarios, by identifying critical tasks related to usability: error-prone tasks (effectiveness), time-consuming tasks and tedious tasks (efficiency) and tasks made under stress or discomfort (satisfaction) [31] [33]. We also used the cognitive dimensions of notations framework [19] which provides a suitable vocabulary for analyzing the usability of programming languages and tools. Finally, we derived usability insights from the critical tasks identified in the scenarios. An overview of our case-study research methodology is proposed in Fig. 2. Scenarios, including critical tasks and derived usability insights are available for further reading [15][16].
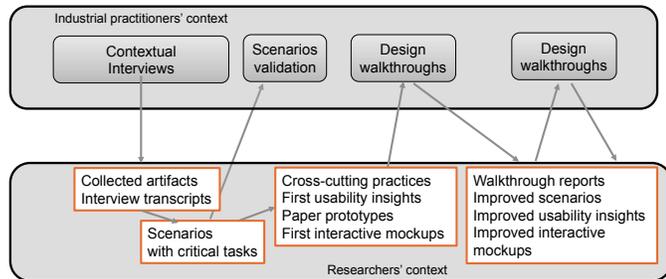

Fig. 2. Our case study research methodology

As we wanted to go beyond "user-friendliness", we combined two definitions of usability to formulate detailed usability insights. First, we considered the ISO guidance on usability [31], in which usability is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness (accuracy and completeness with which users achieve specified goals), efficiency (resources expended in relation to effectiveness), and satisfaction (freedom from discomfort, positive attitudes towards the use of the product), in a specified context of use". In addition to the ISO definition, we considered the six styles of usability defined by Lauesen [33]: performance (specification of task time), defect (specification of problem which causes the user to make a mistake), subjective (satisfaction of the user), process (specification of the process for ensuring usability, such as iterative pro-

totype-based process), design (specification of the user interface with screen pictures) and guidelines (such as the ones provided by HCI literature [10][39][22][43]).

The combination of the two definitions resulted in the framework of Fig. 3. , in which a usability insight has the following syntax: "In order to achieve [goal], the system allows [user] to perform [task] with [usability measure] [value]".
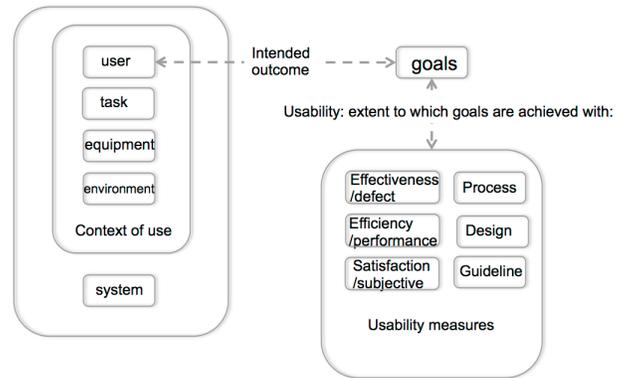
In order to achieve [goal], the system allows [user] to perform [task] with [usability measure] [value]

Fig. 3. A usability framework, combining ISO 9241 with [33]

IV. FINDINGS ON RE PRACTICES AND TOOLS

The findings on RE practices and tools (RQ-1, 2) are related to (A) the nature of projects and artifacts, (B) the discovery of crosscutting RE practices with respect to tools. We finally propose a summary of the observed RE practices coupled with the tools in use (C).

A. Projects and Artifacts

Nine different projects were observed during the study. Five projects (helicopter cockpit, traffic collision avoidance for civil aircraft, evolution of avionics suite of a turboprop, airport departure manager, communication service between current and future air traffic control systems) relate to evolutions of existing systems i.e. the system is already in operation, system components and interfaces are known, and the project goal is to integrate a new service. This type of project is frequent for aeronautics engineers due to the long lifetime of air and ground systems (20 years at least). The establishment of a new version of the system is time-consuming whatever the content of the version, due to the need to produce documents for certification audit. The number of system requirements varies from 100 to 2500 for a new service. Usually, one system requirement is refined into three detailed requirements at the lower level.

Among the four other observed projects, one relates to a new system for air traffic control communications, with call for tender, supplier evaluation and comparison. The last three observed projects are research projects in which new system modeling techniques and tools are studied for future business.

The output data that we collected are diverse in nature: architecture diagrams, specification documents, requirements spreadsheets, traceability matrices, slides, tests catalogs, scripts, meeting minutes and e-mails. Specification documents

(System and Subsystem Specification, Software Requirements Specification) contain hundreds of pages: they are first generated from tools and then edited manually.

B. Crosscutting RE practices

Among the different projects and companies, we discovered three crosscutting contexts of use, as defined in Fig. 3. :
(1) Collaborative exploratory design without RE-specific tool, (2) Experimenting requirements modeling with a RE-specific tool, (3) Requirement refinement and management with a RE-specific tool.

1) Collaborative exploratory design without RE-specific tool

We observed this first context of use in three different companies (B, C and D in TABLE I.). It happens at the early stage of the RE process, when a new service needs to be integrated into an existing system. This new service is defined by initial high-level system requirements. The users are the requirement engineer and the system component suppliers.

We observed that the requirements engineers do not decide the allocation of system requirements by themselves, nor write the requirements for each system components to finally transmit them to the component suppliers, as prescribed by the official V-cycle process. On the opposite, they establish a list of questions raised by the new system requirements to fulfill (or anomalies to correct), and then discuss these questions with the suppliers during workshops. Their objective is to create a panel of technically feasible solutions for requirements allocation, since they have been discussed and studied with suppliers. Engineers do not collaborate around requirements but around design questions to be solved: the engineers' goal is to create a matching problem-solution pair, in compliance with the problem-solution co-evolution model [13]. As described by Rittel [43], the designer's reasoning is disorderly, because "learning what the problem is *is* the problem". These observations are consistent with the opportunistic and adaptive RE process reported in prior work [40][41].

Engineers draw system behavior representations as a reasoning means to discover the design questions. General-purpose tools are used to support this work: drawing editors or whiteboards to express behavioral views and elicit design questions, Excel or PowerPoint documents to record the design questions, email and video-conferencing to share them.

They are very satisfied by this work: they consider the design as the core activity. They do not complain about tools since they use pliant ones [20] i.e. flexible and responsive. However, the different produced files are stored in a common repository at best, but more frequently on personal computers. The requirement engineers do not spend time upgrading the official tools with their separate records. The consequences are dispersion and loss of information in terms of pre-traceability of requirements.

2) Experimenting requirements modeling with a RE-specific tool

We observed this second context of use in three different companies (A, C and D in TABLE I.) on R&D projects. In order to counterbalance the huge number of textual requirements and foster a model-centric approach, some companies have chosen to buy and try system-modeling tools that rely on formalisms such as SysML.

The observed companies are not willing to transpose baselines of existing systems into SySML models (too expensive), but rather to assess the capacities/expressiveness of system modeling on R&D projects or future business. The official goals are the modeling of several system behaviors and the comparison of modeled alternatives through simulation to make asserted design choices. We observed that system-modeling tools are not used in this prescribed way. Only the selected solution is modeled with the tool. Trade-off studies are really done through workshops and discussions, using white-boards and drawing editors, as described before.

Some basic usability problems known as viscosity i.e. resistance to change [19], can explain the user preferences for drawing editors. For instance, when the requirement engineer wants to add an output on a system element, he has to perform the graphical re-arrangement manually for each instance of the modified system element: "*this is box by box, macros are missing*".

Moreover, since the work on requirements is not linear thinking, requirements engineers are not able to fill in the requirement properties at once. When resuming the graphical system model, they spend a lot of time remembering the work done and the work to be done, by browsing system elements and displaying the properties window for each one: "*with 11 tabs, it's so cumbersome that we have the feeling that properties are classified in the reverse order we need*".

3) Requirement refinement and management with a RE-specific tool

We observed this third context of use in two different companies (C and D in TABLE I.). It happens at later stages of the RE process. The users are requirement engineers that are responsible for the requirements management of an existing system. Several evolutions (and corrections of anomalies) are under study, and several system releases are under development. The goal of the engineers is to build and maintain the requirements baselines of system releases, in order to provide the traceability documents required by the certification process [12]. They mainly use a RE-specific tool (DOORS) to manage the requirements baselines and generate documentation, and Excel as a by-pass.

The requirement engineers express much dissatisfaction at these stages: tasks are evaluated as routine and "administrative", characterized by less creativity and more rigor. The main tasks we observed are driven by the search for completeness and traceability of requirements: writing consistent requirements, conducting coverage analysis and managing the content of system releases. But the tool in use does not fully support these tasks. Since the data model is not integrated, the engineers have to perform the character formatting of domain concepts manually. When performing coverage analysis, the graphical representation is the same little arrow whatever the number of refined requirements (Fig. 1.), which conveys a feeling of flat structure reported as unclear by participants. Finally, the collaboration with the component suppliers through

the tool-generated documents is not effective: the list of the refined requirements conveys a piecemeal view of the system and makes the documents unreadable to anyone but the writer.

The consequences are loss of motivation and loss of time in establishing and maintaining traceability of an overwhelming number of requirements.

C. Summary

The detailed analysis of the crosscutting practices allows us to identify five aspects that significantly change over time during the RE process: traceability, requirement maturity (specification dimension for Kohl [42]), collaboration, system behavior representation and user satisfaction. The assessment of these five aspects reveals two opposite states over time that we can couple with tool usability (Fig. 4.).

At the early stage of the process (context of use IV.B.1), the requirement engineers' work starts as a collaborative activity with component suppliers centered on design questions. They use general-purpose desktop tools for their flexibility and do not spend time synchronizing RE-specific tools, at the expense of traceability (grey line in Fig. 4.).
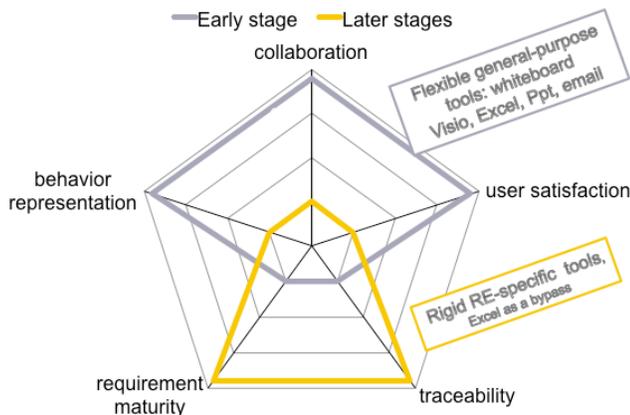

Fig. 4. Two opposite states in RE activities over time

At later stages of the RE process (contexts of use IV.B.2) and IV.B.3)), when the focus shifts to requirement refinement and verification, engineers must use RE-specific tools to grant traceability. The lack of tool usability yields significant time loss and dissatisfaction, in addition to the intrinsic lower level of interest that requirements engineers demonstrate for these tasks (yellow line in Fig. 4.).

V. USABILITY INSIGHTS

From the findings on RE practices and tools, we seek to improve RE-specific tool usability (RQ-3). To this end, as described in section III, we performed the step-by-step analysis of the scenarios and identified critical tasks related to usability. This analysis allowed us to formulate usability insights, structured by the usability framework (Fig. 3.): "In order to achieve [goal], the system allows [user] to perform [task] with [usability measure] [value]".

A. Usability insights to support early stage of the RE process

Traceability can be improved at the early stage of the process if RE-specific tools help engineers perform the activities we observed in the context of use IV.B.1)"Collaborative exploratory design without RE-specific tool", currently performed with other tools for their flexibility: reasoning on system behavior representation to elicit design questions and collaboration with suppliers.

1) Reasoning on system behavior representation to elicit design questions

In order to elicit design questions raised by a set of initial system requirements, RE-specific tools shall allow the requirement engineer to:

- Draw behavioral views of the system using graphical elements such as boxes and arrows (effectiveness: no need to remember or comply with semantic meaning of notation)
- Change behavioral views of the system with a minimum number of actions (efficiency in editing actions to support exploration)
- Create a design question from a graphical element (box or arrow) of the behavioral view with a minimum number of actions (efficiency to foster traceability)
- Edit and visualize a tabular list of design questions, linked to the initial system requirements (design: the tabular list of the design questions is a complementary view of the behavioral view)
- Edit requirement, drawing and design question in any order in any view (behavioral view or tabular list) (effectiveness in the number of elicited design questions) (efficiency: no predefined order in edition to support exploration).

2) Collaboration around design questions

In order to collaborate around design questions raised by a set of initial system requirements, RE-specific tools shall allow the requirement engineers and the component suppliers to:

- Share the tabular list of design questions and the behavioral view in a co-located or distributed meeting (efficiency in sending and reading the data)
- Discuss each design question in any view (behavioral view or tabular list) (efficiency in editing and accessing discussions to foster traceability)
- Change a requirement statement in a visible and persistent manner (design: *was*+ old statement" and *is* + new statement) linked to a discussion (effectiveness: traceability of changes)

B. Usability insights to support later stages of the RE process

In order to find usability insights to support later stages of the RE process, we started from the analysis of scenarios of contexts of use IV.B.2), IV.B.3). However, participants expressed so much dissatisfaction that analyzing scenarios from a usability point of view was not sufficient to find valuable usability insights. For these key issue scenarios, we resorted to mockups and walkthroughs to better understand and formulate usability insights. In the next sub sections, we present the resulting usability insights from the scenarios and walkthroughs analysis, with the help of mockups when necessary.

1) Refining a system model

In order to refine a system model, RE-specific tools shall allow the requirement engineer to:

- Modify graphical properties of a set of objects with a minimal number of actions (efficiency in editing actions to foster alternative modeling)
- Search and edit properties by key word from a graphical element (efficiency of search action)
- Retrieve the completion status of properties from a graphical element (efficiency in resuming the model).

2) Writing consistent requirements

In order to write consistent requirements, RE-specific tools shall allow the requirement engineer to:

- Select and edit words of the data model during requirement statement edition (efficiency in editing action + effectiveness: the words are compliant with the data model)
- Facilitate highlighting of words of the data model in requirement statement (efficiency in editing action and reading)
- Detect requirement similarities during requirement statement edition (effectiveness: no redundancy).

3) Conducting coverage analysis

For this scenario, we explored more deeply a treemap visualization [47]: the entire system fully occupies the view, subdivided into rectangles that represent the requirements (see Fig. 5.). The space allocated to each category is related to the number of requirements. Requirements that have not yet been refined are coded in black, showing the "requirements hole". One can navigate through the structure of requirements by selecting the title bar and can then focus on a level. At the lowest level, we display the requirement text at the top and the refined requirements below (one rectangle for each refined requirement). In addition, we made some actions (verify, reject, modify, comment) available on each requirement by right clicking to support verification task.
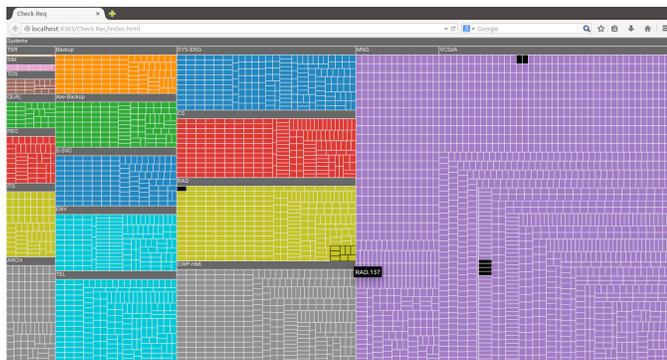

Fig. 5. A treemap visualization for requirements overview

Participants were enthusiastic about the treemap and we analyzed the reasons why the visualization is adequate: it allows them to see the overview of the requirements, focus on a level, and read the system requirement and the refined requirements at the same time for coverage analysis. The available actions on requirements were evaluated as helpful for refinement of requirement while reading. The link with the data model was also addressed through discussions on the right clicking actions. The black hole feature was evaluated as worthwhile for anomaly detection at a glance, in comparison with the tedious matrix reading.

Hence, in order to conduct coverage analysis, RE-specific tools shall allow the requirement engineer to:

- Visualize the overview of requirements without any action (efficiency in overview understanding and communication to outsider) (treemap as an example of design)
- Access the requirements of a given level with a minimal number of actions from the overview visualization (efficiency of navigation) (treemap as an example of interaction design)
- Read all the refined requirement statements and the initial requirement statement always displayed in its entirety (efficiency in reading) (treemap as an example of design)
- Access direct actions on a requirement from the overview visualization: validate, reject, comment, modify (efficiency) (treemap with right click actions as an example of interaction design).
- Identify the requirements not refined at a glance on the overview visualization (efficiency in anomaly detection) (black requirement on the treemap as an example of design).

The next step for participants was to deal with the identified anomalies. This is compliant with the visual information-seeking mantra "overview first, zoom and filter, then details on-demand" [44].

4) Identifying and processing anomalies

For this scenario, we integrated an interactive inspector for multiple objects [23] into the treemap visualization (Fig. 6.). The inspector can be triggered on the system level, on the category level or on the requirement level, by clicking on the text label: it shows all property values used by the set of corresponding requirements. By toggling values, one can finely tune the filter: in the figure, only the requirements written by the toggled authors (HGB, SCO) are highlighted.
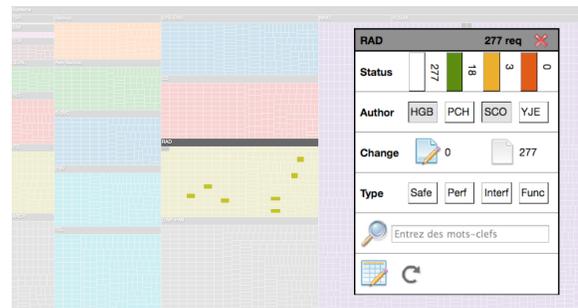

Fig. 6. A requirement inspector for interactive filtering

One can access the tabular visualization of the resulting filtered requirements by clicking on a tabular icon at the bottom of the inspector. We have integrated a search function in the inspector tool to support consistency checking. For instance, if one enters "flight sequence timeline", all requirements that

contain the group of words are highlighted on the treemap and can then be displayed in a tabular visualization to support comparison and detect similarities.

We conducted discussions about anomaly detection and processing thanks to the inspector, which led to new usability insights. Participants' feedback shows that the status (modified-commented-rejected) filter and the direct feedback on the treemap were valuable to get an overview of the verification work completion. The "author" property was evaluated as a valuable filter, because crosschecking is frequent in final verification. Tabular visualization of filtered requirements was evaluated as helpful to process the filtering results step by step and work directly on the tabular visualization.

Through participants' discussions, it appears that the search feature, firstly designed for consistency checking, can also support impact analysis in order to identify the nature and number of requirements that must be modified to implement an evolution. Other filters have been identified through discussions such as the requirement type or the change status. However, we do not provide a full list of filters here, only the ones that are necessary for verification tasks.

In summary, RE-specific tools shall allow the requirement engineer to filter requirements *interactively* with *immediate* feedback on the overview visualization, at *any* system level (efficiency in filtering and results visualization) (inspector and treemap as an example of design):

- By keywords, in order to process consistency checking and impact analysis
- By author, in order to process crosschecking.
- By completion status, in order to resume the work and evaluate the work progress.
- By type (safety, performance, functional, interface), in order to prioritize the checking process (safety requirements are mandatory first).

In order to process filtered requirements, RE-specific tools shall allow the requirement engineer to display requirements on-demand after interactive filtering in an *editable* tabular visualization (efficiency in handling anomalies).

5) Managing system release

For the scenario of system release management, we have explored the chord diagram visualization (Fig. 1.) as an alternative graphical method to display matrices [26]. System evolutions (in red), system components (in blue), and operational sites (in green) are arranged radially around a circle. The relationships between them are drawn as arcs connecting them. The selection of a given evolution highlights the system components that implement it and the impacted sites (Fig. 8. , top).

The participants evaluated as helpful the chord diagram in the context of communication with the steering committee. They proposed to add extra information for a better decision support, such as requirement statistics of the evolution (number of new requirements, number of modified requirements, number of unchanged requirements – top of Fig. 8.). They suggested that it could also improve the impact analysis of a new system versioning, by browsing from one evolution to another.

The participants also imagined this visualization being used for communication within the RE team to get a better aware-

ness of the multiple evolutions under study, and with the component suppliers to filter on specific information. Finally, they were willing to use this visualization as a filtering tool, allowing the extraction of all the new/modified requirements related to an evolution, classified by system components "*by drag and drop of the selection in a separate window*".
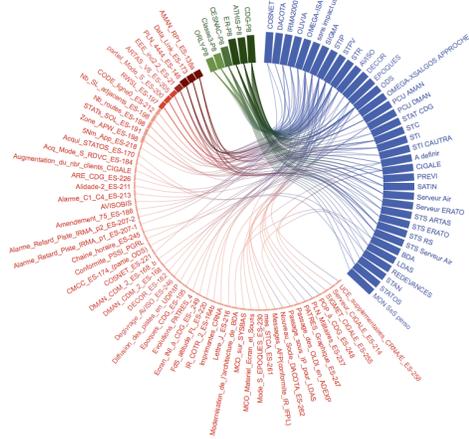

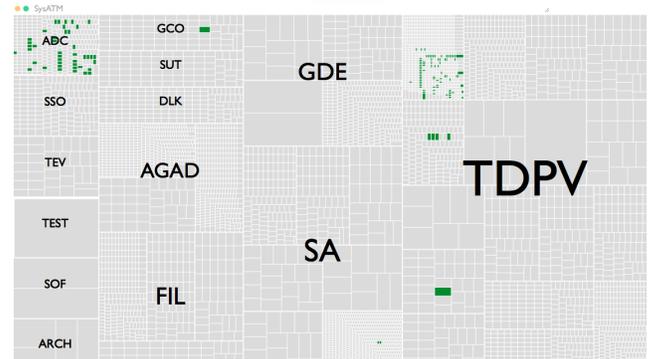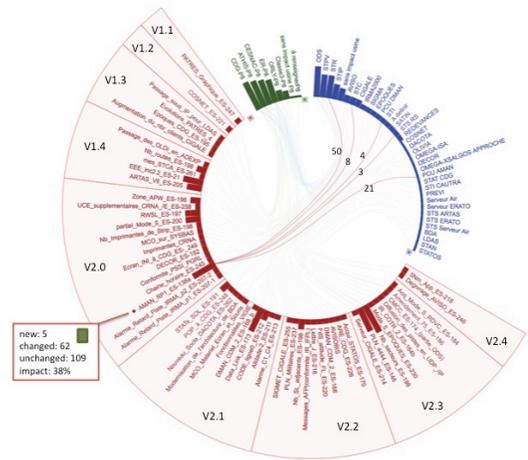Fig. 7. a chord diagram visualization for release planning

Fig. 8. Linking between visualizations: the selection of a given evolution on chord diagram shows the corresponding requirements on treemap

In summary, in order to manage system release, RE-specific tools shall allow the requirement engineer and the steering committee to:

- Visualize the evolutions under study and the corresponding impacted components (efficiency in selecting and reading, chord diagram as an example of evolution visualization design)
- Reschedule evolutions over time in a minimal number of actions (efficiency in tracing).
- Browse from one evolution to another with requirement statistics (new, modified, unchanged) for impact comparison (efficiency in browsing, chord diagram as an example of interaction design)
- Share evolution visualization (effectiveness in RE team cognition) (chord diagram as an example of design)
- Filter the initial requirements and the refined requirements by component (efficiency in filtering) (effectiveness of communication with suppliers) (chord diagram as an example of interaction design)

VI. THREATS TO VALIDITY

As a qualitative research study, the findings may depend on 1) the data collected in the studied setting and 2) the data analysis [51]. We mitigated the data quality risks by ensuring diversity in companies and engineers, and collecting artifacts to underpin participants' opinions. As for the analysis, only one researcher performed it. To mitigate the bias risks, we used an established usability framework for the analysis, and wrote in-depth details and materials in the scenarios. We also made 3 participants (from A and C companies) validate the scenarios to assess their validity.

We cannot claim that the results of the study can be generalized to any industrial domain. Nevertheless, the real-life projects that we investigated are diverse (air traffic control, helicopter, aircraft, research project). Besides, the driving forces of the phenomena we observed in aeronautics are the production of complex systems (high number of heterogeneous and interconnected components) with respect to requirements traceability for safety assurance and certification purpose. Thus, we assume that the obtained results in aeronautics may be useful to similar safety-critical complex systems [45].

VII. CONCLUSION

In this work, we addressed three research questions 1) what tasks are actually performed by requirements engineers, 2) how current tools support these tasks and 3) how usability tool can be improved for Requirements Engineering. To do so, we conducted a qualitative study, using interviews and mockups, with fifteen industrial practitioners from four aeronautics companies.

We found three crosscutting RE practices, in which RE-specific tools constrain engineers to a rigid workflow, conflicting with the adaptive exploration of the problem and the fluctuating nature of requirements. At early stage of the RE process, engineers start by using general-purpose tools to foster exploration and collaborative work with suppliers, at the expense of traceability. At later stages of the process, when engineers shift to requirements refinement and verification, they must use RE-specific tools to grant traceability. Then, the lack of tool usability yields significant time loss and dissatisfaction. Based on scenarios of observed RE practices and design walkthroughs, we formulated 29 usability insights for RE-specific tools to support engineers' tasks throughout the RE process.

From the results on usability insights, RE-specific tools implementers may improve their products to better support companies in safety-critical systems engineering. We intentionally devised the usability insights as a support for technology transfer. Findings on cross-cutting contexts and scenarios can be used by RE researchers to target industry concerns on requirements management and perfective maintenance of complex systems. The practitioners involved in the study setting were willing to reflect on their activities: engineers are trained to perform intellectual and open-ended tasks [44] and are consequently inclined participants to case study methodology.

We plan to further expand the design space of interaction and visualizations, in order to blur the boundaries between stages of the RE process and make traceability compatible with flexibility.

REFERENCES

[1] Abad Z. S. H., Noaeen M. and Ruhe G. 2016. Requirements Engineering Visualization: A Systematic Literature Review. In Prof. of RE'2016, IEEE, pp 6-15.

[2] Baldonado M.Q.W., Woodruff A. and Kuchinsky A. 2000. Guidelines for Using Multiple Views in Information Visualizations, in Proc. Of the working conference on Advanced visual interfaces (AVI 2000), pp. 110-119.

[3] Bostock M., Ogievetsky V. and Heer J. 2011. D^3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12, pp 2301-2309.

[4] Carrillo de Gea J., Nicolás J., Fernández Alemán J., Toval A., Ebert C., Vizcaíno A. 2012. Requirements engineering tools: Capabilities, survey and assessment, Information and Software Technology, Volume 54, Issue 10, pp. 1142-1157.

[5] Carrillo de Gea J., Nicolás J., Fernández Alemán J., Toval A., Ebert C., Vizcaíno A.2015. Commonalities and differences between requirements engineering tools: A quantitative approach. Comput. Sci. Inf. Syst. 12(1): pp 257-288.

[6] Carroll JM. 2000. Making use: Scenario-based design of human-computer interactions, MIT Press, Cambridge, Massachusetts.

[7] Chatzoglou P.D. 1997. Factors Affecting Completion of the Requirements Capture Stage of Projects with Different Characteristics, Information and Software Technology. 39, pp 627–640.

[8] Cheng B. and Atlee J. 2007. Research Directions in Requirements Engineering. In Proc of Future of Software Engineering (FOSE '07), Washington, DC, USA, pp 285-303.

[9] Cleland-Huang J., Gotel O., Huffman Hayes J., Mäder P. and Zisman A. 2014. Software traceability: trends and future directions. In Proc. of FOSE 2014. ACM, pp. 55-69.

[10] Constantine L. and Lockwood L. 1999. Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. ACM Press/Addison-Wesley Publ. Co., NY, USA.

[11] Cooper J.R., Lee S-W., Gandhi R.A., Gotel O. 2009. Requirements Engineering Visualization: A Survey on the State-of-the-Art, 4th Inter. Workshop on RE Visualization, pp.46-55.

[12] DO-178B/ED-12B Software considerations in airborne systems and equipemnt certification, Requirements and Technical Concepts for Aviation.

[13] Dorst K. and Cross N. 2001. Creativity in the design process: Co-evolution of problem-solution. Design Studies. 22, pp 425–437.

[14] Fallman D. 2003. Design-oriented human-computer interaction. In Proc. of CHI '03. ACM, pp 225-232.

[15] Gaspard-Boulinc H. and Conversy S. 2014. Exigences d'utilisabilité pour les outils d'ingénierie des exigences, in Proc. of IHM '14, ACM, pp 70-79.

[16] Gaspard-Boulinc H. and Conversy S. 2017. Annex A: Scenarios https://hal.inria.fr/hal-01532781

[17] Ghazi P. and Glin M. 2016. An Exploratory Study on User Interaction Challenges When Handling Interconnected Requirements Artifacts of Various Sizes. In Prof. of RE 2016, IEEE, pp 76-85.

[18] Gotel O. and Mader P. 2009. How to Select a Requirements Management Tool: Initial Steps, in Proc. Of RE '09. IEEE, pp 365-367.

[19] Green T. R. G. 1989. Cognitive dimensions of notations. In A. Sutcliffe and L. Macaulay (Eds.) People and Computers V. Cambridge, UK: Cambridge University Press, pp 443-460.

[20] Harris J. and Henderson A. 1999. A better mythology for system design. In Proc. of CHI '99, ACM, pp. 88-95.

[21] Hall T., Beecham S. and. Rainer A. 2002. Requirements Problems in Twelve Software Companies: An Empirical Analysis, In Proc. of the Conference on Empirical Assessment in Software Engineering, Keele, pp. 1-17.

[22] Hix D. and Hartson H. 1993. Developing User Interfaces: Ensuring Usability Through Product and Process. Wiley & Sons, Inc., New York, NY, USA.

[23] Hoarau R. and Conversy S. 2012. Augmenting the scope of interactions with implicit and explicit graphical structures. In Proc. of CHI '12, ACM, pp 1937-1946.

[24] Hoffmann M., Kuhn N., Weber M. and Bittner M. 2004. Requirements for requirements management tools, In Proc. of 12th Requirements Engineering Conference, IEEE, pp. 301-308.

[25] Holtzblatt K, Jessamyn B. and Shelley W. 2005. Rapid Contextual Design a How-To Guide to Key Techniques for User-Centered Design, Interactive Technologies. San Francisco: Morgan Kaufmann.

[26] Holten D. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data, IEEE Trans. Visualization and Computer Graphics, 12(5), pp. 741-748

[27] INCOSE. Requirements Management Tools Survey http://www.incose.org/ProductsPublications/techpublications

[28] ISO/IEC ISO 29148:2011, Systems and software engineering - Life cycle processes-Requirements engineering, IEEE.

[29] ISO/IEC TR 24766:2009, Information Technology – Systems and Software Engineering – Guide for Requirements Engineering Tool Capabilities, IEEE.

[30] ISO/IEC 9126, Software Engineering- Product quality- part3: internal metrics", IEEE.

[31] ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals - Part 11 Guidance on usability, IEEE.

[32] Karlsson L., Dahlstedt S.G, Regnell B., Natt och Dag J. and Persson A. 2007. Requirements engineering challenges in market-driven software development - An interview study with practitioners. Inf. Softw. Technol. 49, 6, pp 588-604.

[33] Lauesen S. 1998. Usability requirements in a tender process. In Computer Human Interaction Conference, Proceedings, Australasian, pp 114-21.

[34] List of Requirements Engineering tools:
http://www.capterra.com/requirements-management-software/
http://www.scenarioplus.org.uk/vendors.htm

[35] Lubars M., Potts C. and Richter C. 1993. A Review of the State of the Practice in Requirements Modelling, In Proc. of the First IEEE Int. Symposium on Requirements Engineering, pp. 2–14.

[36] Mackay W. E. 2003. Educating Multi-Disciplinary Design Teams. In Proc. of Tales of the Disappearing Computer, pp. 105-118.

[37] Miles M. 1979. Qualitative Data as an Attractive Nuisance: The Problem of Analysis. Administrative Science Quarterly 24, n° 4, pp 590-601.

[38] Niu N., Reddivari S. and Zhangji C. 2013. Keeping requirements on track via visual analytic, In Proc of the 21st IEEE International Requirements Engineering Conference, pp 205-214.

[39] Nielsen J. 1993. Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA

[40] Nguyen L., Carroll J. and Swatman P.A. 2000. Supporting and monitoring the creativity of IS personnel during the requirements engineering process, System Sciences, Proc. of the 33rd Annual Hawaii International Conference on, On page(s): 9 pp. vol.1

[41] Nguyen L. and Swatman, P. A. 2000. Essential and incidental complexity in requirements models, in Proc. of 4th IEEE International Conference on Requirements Engineering ICRE 2000.

[42] Pohl K. 1994. The 3 dimensions of requirements engineering: a framework and its applications. Inf. Syst. 19(3), pp 243-258.

[43] Rittel H. 1987. The reasoning of designers, International Congress on Planning and Design Theory in Boston.

[44] Schön DA. 1983. The reflective practitioner: How professionals think in action, Basic Books, New York

[45] Seddon P. and Sheepers R. 2012. Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. EJIS 21(1), pp 6-21.

[46] Shneiderman B. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In Proc. of the IEEE Symposium on Visual Languages, p336-343, Washington.

[47] Shneiderman B. 1992. Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph. 11, 1,

[48] Sikora E., Tenbergen B. and Pohl K. 2011. Requirements Engineering for Embedded Systems: An Investigation of Industry Needs. In Requirements Engineering: Foundation for Software Quality, Springer, pp 151-65.

[49] Winkler S. 2008. On Usability in Requirements Trace Visualizations. In Proc. Of REV '08. IEEE, pp 56-60.

[50] Winkler S. and Pilgrim J. 2010. A survey of traceability in requirements engineering and model-driven development. Softw. Syst. Model. 9(4), pp 529-565.

[51] Wohlin C. and Aurum A. 2014. Towards a decision-making structure for selecting a research design in empirical software engineering. Empirical Soft. Eng. Journal, Springer.

[52] Yin R. 2009. Case Study Research: Design and Methods. SAGE.