Analysis of trends in scientific publications by an NLP toolkit: A case study in Software Development Methods for Enhanced Living Environment

Melaku Girma*, Nuno Garcia†, Eftim Zdravevski§, Mesfin Kifle‡, Nuno Pombo†, Vladimir Trajkovik§

*Software Engineering Track, IT Doctoral Program, Addis Ababa University, Addis Ababa, Ethiopia
Email: melaku.girma@aau.edu.et

†ALLAB, University of Beira Interior, Covilhã, Portugal,
Email: {ngarcia,ngpombo}@di.ubi.pt

§Computer Science and Engineering, University Ss. Cyril and Methodius, Skopje, Macedonia
Email: {eftim.zdravevski,trvlado}@finki.ukim.mk

‡Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia
Email: mesfin.kifle@aau.edu.et

***Abstract*- As the number of published scientific articles increases, the analysis of trends and state-of-the-art in software engineering is becoming very time-consuming and laborious task. To address the ever-growing demands for systematic literatures review techniques, rapid review and scoping reviews techniques have emerged. We used an NLP powered tool, which employs the PRISMA surveying methodology, to automate most of the review processes. We used it to automatically review relevant articles indexed in IEEE Xplore, PubMed and Springer digital libraries on the topic "Software Development for Enhanced Living Environments and Ambient Assisted Living". The relevant articles identified by the NLP toolkit contained up to 21 properties clustered into 3 logical groups. We discovered that Software Development for Enhanced and Assisted living environments attracted an increased attention from the scientific communities over the last 10 years and showed several trends in the specific research topics that fall into this scope. The research uncovered that iterative software methodology had been the most attractive research topic in the field. Despite the enormous empirical evidence on application and success stories of agile development methodologies in many software development engineering, it received a little attention from the scientific community in the software development for Enhanced and Assisted Living Environments. The NLP toolkit identified the most relevant articles that contained the defined properties in the search. Hence, it significantly reduced the manual work, while also generating informative tables, charts and graphs.**

***Keywords- software development for enhanced living environments; Software development for ambient assisted living; NLP toolkit; Automated surveys; Scoping review;Systematic review;Rapid review*.**

I. INTRODUCTION

Software systems are being increasingly employed in all area of human endeavors. They are applied in many sectors, devices, and products at an alarming rate of growth. New generations of medical devices, automobiles, aircraft, manufacturing plants, nuclear power generating stations, automated trains, banking and investment systems, manufacturing systems, and a growing number of automated systems within our homes heavily dwell on software to enable new functions, provide pre-existing functions more efficiently, reduce time to service a user need, and reduce effort and competence required by people providing services [1]. In particular, they are heavily applied in safety and mission critical systems, with safety-critical systems being a subset of the wider area of mission-critical systems [2]. A mission critical system is defined as "a system that is essential to the survival of a business or organization. When a mission critical system fails or is interrupted, business operations are significantly impacted" [3]. Grant defined Safety-critical systems as "*subset of mission-critical systems for which, failure may result in harm or death to life, or damage or destruction of property*" [2]. Generally, a system is safety-critical if its failure can cause financial loss, damage to the environment, injury to people and in some cases, loss of lives [1] [4].

The application of software in safety-critical systems has increased in upward trajectory to the degree where software failures may completely impair system safety [5]. For example, the automotive domain include software in a break system, which in case of failure could result in unacceptable hazards, but also active safety functions that override driver behavior in certain situations to avoid a crash [5]. Similar applications in another domain include but not limited to railway [6] and avionics [2].

Safety-Critical Systems have stringent regulation and scrutinized requirement certification against industry standards by the relevant governing body [7]. Hence, a demanding need to design them with safety in mind. Unlike hardware where general safety design principles have been integrated into standards, the standards for development of safe software are still evolving. For example, the standards ISO26262, IEC61513, IEC62304, DO-178B for automotive, nuclear, medical and avionics, respectively, are envisioned to ensure best engineering practices are adhered to

978-1-7281-7219-4/20/$31.00 ©2020 IEEE

and are often based on traditional/waterfall also called plan-driven approaches [5]. This is mainly due to the fact that agile methods neglect upfront requirement specifications that model the solution on a paper before embarking upon a system implementation [8].

Agile methods emphasize close collaboration between customers and developers, while delivering software within time and budget constraints. As agile methods depend on frequent informal face-to-face communications rather than providing lengthy documentations, the process is repetitive, adaptive, and minimally defined [9]. The key features of the methods are continuous requirements gathering; frequent face-to-face communication; pair programming; refactoring; continuous integration; early expert customer feedback; and minimal documentation [10]. Scrum is the most widely used method [11].

Scrum framework composed roles, ceremonies, and artifacts [12]. The three distinct roles in the Scrum process are the Product Owner, the Team and the Scrum master [13]. The ceremonies of Agile Scrum embrace Daily Scrum Meeting, the Daily Scrum of Scrums Meeting, the Sprint Review Meeting and the Sprint Planning Meeting. Agile Scrum process also presents three artifacts, known as Product Backlog, the Sprint Backlog, and the Burndown Chart. Scrum is used together with Kanban method by agile software companies.

Kanban is an agile method developed in early 1940s as a method of the Toyota Production System [14]. This method pays attention to visualization and signaling of workflow. The main purpose of Kanban method is to match the volume of work in progress to the attributes of the team. It harnesses more flexible planning options, transparency in the development flow, makings tasks more visible and faster output [14]. The team in Kanban method attempts for continuous delivery, instead of having hard milestones for delivery. Unlike Scrum, it doesn't have any existing roles for the development team [14].

Despite its obvious shortcomings, agile methods have increasingly been employed to meet the ever increasing need for short development cycles and quick time to market software products [15]. Complex dynamic systems-of-systems require components of different vendors to interact at runtime. This dynamic nature of such environments may necessitate continuous deployment for product success as well as maintaining functional safety, e.g. in the automotive domain [16].The present focus on autonomous driving and intelligent vehicles is the cause for the high complexity of Safety-Critical Systems [5]. Companies like Tesla have proved how the ability to deploy new functions and experiment their performance in the field can get enormous advantages.

Large Software companies such as Google, Apple, and Amazon have entered into automotive market, increasing the need to develop competencies in continuous software engineering. Slowly, companies developing safety-critical systems realize the competitive advantages that agility can provide [5]. However, the existing agile frameworks should be tailored to the needs of the software development organization [17].

European Union has also sought solutions, which is safety-critical, for European citizens' aging issues by integrating Information and Communication Technology (ICT) solutions into habitats, along with improved building design, to enable them live at home active and productive for longer despite cognitive or physical impediments [18]. To this end, improving accessibility, functionality, and safety at home, at work and in society in general requires combining many disciplines together to develop solutions that integrate ICT, ergonomics, healthcare (psychological and physical), building and community design.

European Cooperation in Science and Technology (COST) with COST Action 16226, Smart Habitat for Elderly (SHELD-ON) aims to foster knowledge exchange and the development of a joint research agenda in terms of design and development of multifunctional indoor environments to meet the requirements of Europe's ageing population while promoting healthy and safe ageing [18].

II. RELATED WORK

This research is investigating on Software Development Methods for Ambient Living Environment/ Enhanced Living Environment (ALE/ELE). We identified potentially relevant articles with the following keywords: software Development for Ambient and assisted living, Software Development for Enhanced Living Environment, Software for Older People, and Apps for Older People along with properties in three categories as exhibited in Fig. 1.

This research is envisioned to undertake a Systematic Literature Review (SLR) with the help of automated Natural Language Processing (NLP) toolkit on the software development methodologies that have been effectively utilized in the engineering of software for SHELD-ON.

SLR is a technique that has been used to answer a research question (s) through searching, assessing and synthesizing relevant evidences in order to make conclusion (s). To this end, the "Preferred reporting Items for Systematic reviews and Meta-analyses: the PRISMA statement" [19] [20] is one of the widely used methodologies in most disciplines in general and Software Engineering, recently [21], [22] . Other review techniques wherein SLR model does not fit well have recently been introduced[23]. For example, the rapid review is used when time is of the essence. The scoping review can be used when the goal is not to get detailed answers to specific research question (s) but overview of a broad field [24]. The evidence map is similar to scoping reviews but is focused on specific visual presentation of the evidence across a broad field. Finally, the realist review is used where the question of interest includes how and why complex social interventions work in certain situations, rather than assume they either do or do not work at all.

The aforementioned review types are usually carried out manually, which is daunting and laborious. Therefore, the current research employed Natural Language Processing (NLP) toolkit, which has been developed by some of the authors of this work [25], identify relevant articles, and generate visualizations of trends and relationships and the like. By exploring the publications over the last decade, we have summarized the state-of-the-art software development methodologies, future research focus and publication statistics related to software development approaches in ALE/ELE.

The subsequent sections are organized into different sub sections. Section 2 will elaborate on the parameters of the NLP Toolkit [25], while also describing the data collection processes. Section three gives succinct discussion of the results. Finally, the last section provides conclusion based on the research findings and pinpoints further research works.

III. METHODOLOGY

This research employed the NLP toolkit developed by some of the authors of this paper [26] [27][25]. The toolkit ensures compliance with the terms of use of the digital libraries, in regard to the number of requests per unit time. Additionally, the plotting of aggregate results was integrated and streamlined using the Matplotlib library [28] and NetworkX [29]. The NLP toolkit input parameters are described as follows.

A. Search Input Taxonomy

The user input is a collection of keywords that are used to identify potentially relevant articles and a set of properties that should be satisfied by identified articles. The input is defined with following parameters, which are further enhanced by proposing synonyms to the search keywords and properties by the NLP toolkit, as described in the following sub-section 2.4:

Keywords Search terms or phrases that are used to query a digital library (e.g. Software Development for Ambient and assisted living, Software Development for Enhanced Living Environment, etc.). See example of searched keywords in Fig. 6 and 7. To this end, keywords are searched for independently and duplicates are removed in a later phase.

Properties The properties are words or phrases that are being searched in the title, abstract or keywords section of the identified articles. Example of such properties employed in this study are exhibited in Fig. 8, 9, 10.

Property synonyms In addition to the original form of the properties, also their synonyms or words with similar meaning in the domain terminology, are being searched for in the article's abstract, title and keywords. For each property, only one original form appears in the results for brevity, while the synonyms are omitted. Note that a synonym can be a completely different word, or another form of the same word, such as a verb in another tense or an adjective (e.g. synonyms of Agile: Kanban and Scrum, synonyms of Implementation: code; synonyms of Deployment: Installation; etc.). Therefore, instead of showing all those words, only one word per synonym set is being displayed in the results. Synonyms can be provided by the user, or proposed by the toolkit, with a possibility of fine-tuning the proposals. For the considered use case, the list of used properties and property groups is shown in Fig. 1.

Property groups are thematically or semantically or otherwise grouped properties for the purpose of more comprehensive presentation of results. Properties within property groups are being displayed together in charts or tables. The property group has a name (e.g. Development Methodologies, Critical Systems and Phases), and within a group there are sets of property synonyms (e.g. within the Development Methodologies propriety group: Agile (synonymous with Kanban and Scrum), extreme programming, waterfall, etc.). Exemplary summary results per property group are presented in Fig. 7, while exemplary result per property within groups are shown in Fig. 8, 9, and 10.

Start year the start year (inclusive) of the papers that we are interested in. Default: current year - 9.

End year the end year (inclusive) of the papers that we are interested in. Default: current year.

Minimum relevant properties a number denoting the minimum number of properties that an article has to contain in order to be considered as relevant. Default: 2.

```
{"Development Methodologies":[["agile", "Kanban", "scrum"],
                ["extreme programming"], ["spiral"],
                ["incremental"], ["iterative"], ["prototyping"],
                ["continuous integration"],
               ["rapid application development", "rapid development"],
                ["waterfall"]
            ],
 "Critical systems": [["Mission critical"],
                ["Safety critical"],
                ["Business critical"]
             ],
 "Phases": [["Analysis"], ["Design"],
           ["Product management"], ["Project management"],
           ["Implementation", "Code"], ["Integration"], ["Testing"],
           ["Deployment", "Installation"], ["Maintenance"]]}
```

LEGEND
Property group
Property (Main synonym)
Property (Other Synonyms)

Fig. 1: List of property groups and properties (main and the synonyms).

B. Enhanced Search Capabilities with WordNet

The user provided input in the form of keywords and properties is enhanced by proposing synonyms from WordNet for Python before the actual searching kicks-off [26] [27][25]. In most cases, this increases the robustness of the searched properties by including synonyms that the user might have neglected. However, considering that Word Net is a general-purpose database, some of the proposed synonyms might not be inappropriate. In such a case, the user can manually choose which of the proposed synonyms to be included before the actual processing starts. The toolkit also performs stemming of the properties and the abstract, for a more robust searching. If none of the properties of interest are identified within the abstract, then those articles are removed from the result set, which corresponds to the eligibility step in the PRISMA statement. In addition to this, we can specify the minimum number of properties that need to be identified within an article for it to be considered eligible and potentially relevant.

C. Article sources

The NLP toolkit indexes the following digital libraries (i.e. sources): IEEE Xplore, Springer and PubMed. From PubMed all articles that match the given search criteria (i.e. a keyword) are analyzed. IEEE Xplore results include the top 2000 articles that match a given criteria, sorted by relevance determined by IEEE Xplore. For Springer the search for each keyword separately is limited to 1000 articles or 50 pages with results, whichever comes first, sorted by relevance determined by Springer.

D. Survey Methodology

The methodology used for the selection and processing of the research articles in this section is based on PRIMA [19], [20], as shown in Fig. 2. The goal of PRISMA is to standardize surveys.

The first part is gathering articles based on search keywords. After the articles are collected, the duplicates are removed as well as some of the articles are discarded for various reasons, such as

relevance, missing meta-data, and invalid publication period. Finally, from the selected subset of articles, a qualitative analysis is performed and from those articles, only a certain number is selected for more thorough screening. The NLP toolkit developed by [26] [27], automates most of the steps in the PRISMA approach to significantly reduce the number of articles that need to be manually screened.

Identification and duplicate removal The NLP toolkit perform the identification automatically. First, the candidate articles are identified by querying the integrated libraries with the same search terms (i.e. key-words). While integrating the results from multiple sources (i.e. digital libraries), duplicate removal was also carried out by using the article DOI as their unique identifier. Articles that had already been identified by the search keys from another source were counted towards the number of identified articles per source. To this end, the retrieved articles per source are not disjoint. After the candidate articles were identified, they were processed, and the properties of the texts are used for selection of the relevant articles. The process of article selection is the same as the one presented in [19][20], except for the last part where articles are manually processed by several researchers.

Augmented Screening and Eligibility analysis by NLP After the duplicates were removed, during the screening process discarded articles which were not published in the required time period (e.g. last ten years) or for which the title or abstract could not be analyzed due to parsing errors, unavailability or other reasons. Afterwards, the eligibility analysis was performed, which involved tokenization of sentences [30] [31], English stop words removal, stemming and lemmatization [31] using the NLTK library [32] for Python. At the beginning, this was applied to each property, based on which a reverse lookup was created from each stemmed word and phrase to the original property. The same process was also applied to the title, keywords and abstract of each article. As a result of the stemming, for each property, the noun, verb and other forms were also considered. As a result of the lemmatization and the initial synonym proposal, the synonyms of properties were also taken into account. This resulted in a more robust analysis. Then, stemmed and lemmatized properties were searched in the cleaned abstract and title and the article was tagged with properties it contained.

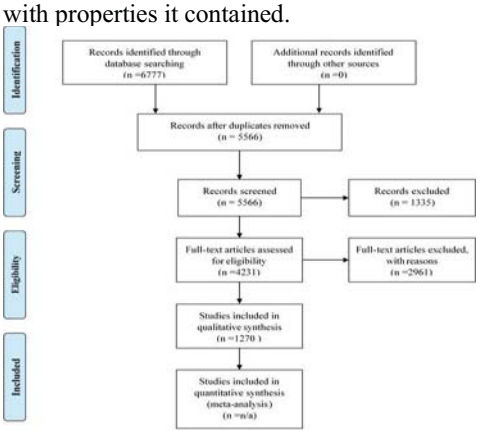


Fig. 2. PRISMA statement workflow with total number of articles for the current survey.

The identified articles were labeled relevant provided they contained at least the minimum relevant properties, defined as an input, in its title or abstract (considering the above NLP-enhanced searching capabilities, thus performing a rough screening). To help in the eligibility analysis, the remaining relevant articles were sorted by number of identified property groups, number of identified properties, number of citations (if available) and year of publication, all in descending order. For the relevant articles, the toolkit automatically generated a Bibtex file with most important fields that can be included in an article for simplified citations. An Excel file was also generated with the following fields: **DOI, link, title, authors, publication date, publication year, number of citations, abstract, keyword, source, publication title, affiliations, number of different affiliations, countries, number of different countries, number of authors, bibtex cite key, number of found property groups, and number of found properties**. The researcher can use this file to drill down and find specific articles by more advanced filtering criteria (e.g. by importing it in Excel). This can facilitate deciding which articles need to be retrieved from their publisher and manually analyzed in more detail in order to determine whether it should be included in the qualitative and quantitative synthesis.

Visualization of aggregate results The results of the processing and retained relevant articles were aggregated by several criteria. The output contained CSV files and charts in vector PDF files for each of the following aggregate metrics:

- By source (digital library) and relevance selection criteria (see Fig. 3).
- By publication year (see Fig. 4a).
- By source and year (see Fig. 4b).
- By search keyword and source (see Fig. 5).
- By search keyword and year (see Fig. 6).
- By property group and year (see Fig. 7).
- By property and year, generating separate charts for each property group (see Fig. 8, 9 and 10).
- By number of countries, number of distinct affiliations and authors, aiming to simplify identification of multidisciplinary articles (e.g. written by multiple authors with different affiliations) (See Fig. 13).

In addition to that, the toolkit also generated graph visualization of the results, where nodes are the properties and the edges are the number of articles that contain the two properties it connects. Articles which do not contain at least two properties and properties that are not present in at least two articles were excluded. An example of this is presented in Fig. 11 and 12. For a clearer visualization, only the top 25% property pairs by number of occurrences are shown (i.e. ones above the 75-th percentile).

A similar graph for the countries of the author affiliations is also generated (see Fig. 13). The top 50 countries by number of collaborations are considered for this graph. Additionally, we show only countries and an edge between them if the number of bilateral or multilateral collaborations between them in the top 5% (above 95-th percentile) within the top 50 countries.

IV. RESULTS

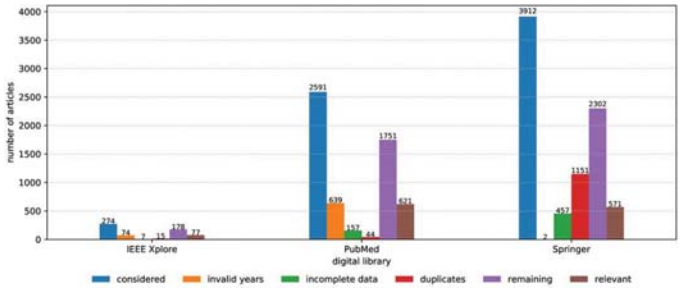
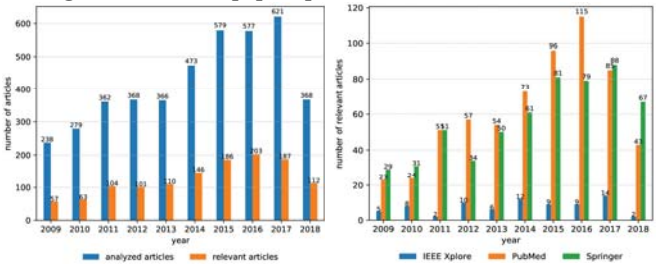


Fig. 3. Number of papers per relevance selection criteria.



a. Number of remaining and relevant articles per year.

b. Number of relevant articles from each digital library per year.

Fig. 4. Number of articles per year and source.

The NLP system with the keywords mentioned in Fig. 6 were used. We searched for these keywords and automatically identified and screened the articles, as exhibited in Figure 2. A more detailed analysis was carried out using the properties that were aggregated into three groups of properties each containing at least three property synonyms, as exhibited in Fig. 1.

Fig. 3 clearly depicts the selection process based on the adopted methodology. From all identified articles based on the keywords, first, the system eliminates the ones with incomplete or invalid meta-data. Second, the duplicate entries are eliminated and lastly, from the remaining ones, the relevant articles are selected if they contain the minimum number of properties (in this case 1). Fig. 4a is the number of remaining and searched for articles from each year and Fig. 4b is the number of relevant articles from each source.

The number of relevant articles grouped by keywords from each source can be seen in Fig. 5. The top 3 keywords by number of papers are “Software for Old People", “Software Development for Ambient and Assisted Living" and “Software Development for Enhanced Living Environment". It is interesting to see that they vary in frequency between different sources, which can be expected, considering that for PubMed the number of analyzed articles is unlimited, unlike the other sources.

On Fig. 6, we can see the distribution of papers per keyword for each year. Notably, the number of papers for some of the keywords is increasing through the years, while for others it is relatively small. Similarly, on Fig. 7, we can see the trends of articles mentioning at least one property from each property group, and evidently, all property groups are becoming more relevant. Apparently, the articles are not covering critical systems as often as the other themes (i.e. Development Methodologies and Phases).

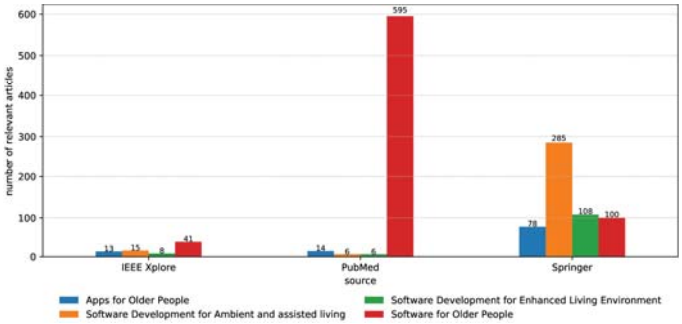


Fig. 5. Number of relevant articles for each keyword from each source.

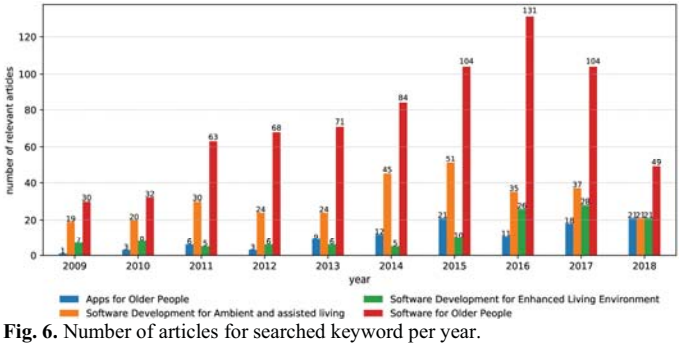


Fig. 6. Number of articles for searched keyword per year.

Properties and keywords follow a similar trend in the number of articles, with most of them reaching the highest number in 2016 and 2017. However, some terms, such as “Apps for Older People", is still on the rise. Note that the numbers from 2018 are inconclusive because, at the time of this analysis, 2018 is not yet finished. Also, the number of articles is increasing in IEEE Xplore and Springer and the in PubMed the number of articles starts decreasing after 2017.

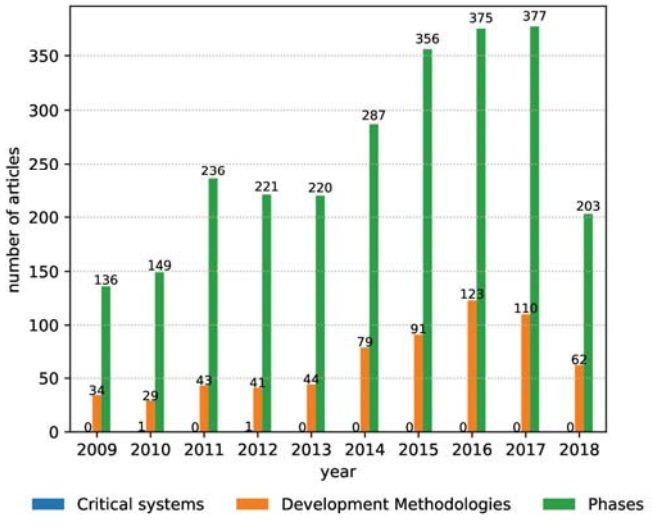


Fig. 7. Number of articles mentioning each property group per year.

After the initial property analysis, for each property group, we analyze the articles based on each property. In Fig. 8, the result about the "Development Methodologies" property group is displayed. To this end, the property agile (embracing Kanban and Scrum), extreme programming, spiral, incremental, iterative, prototyping, continuous integration, rapid application development, rapid development and waterfall in the analysis including their synonyms were considered. The trend clearly depicts that all of the terms are increasing in popularity in the respective research communities. The most popular development methodology in the papers is iterative followed by prototyping and finally agile which slowly and steadily increases in popularity.

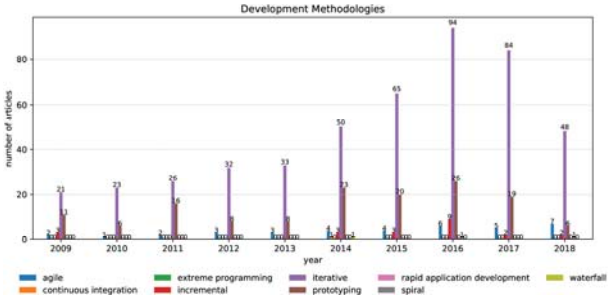


Fig. 8. Article distribution per year and properties in Development Methodologies property group.

Fig. 9 exhibits the result of the second property group, "Phases", which embraces Analysis, Design, Product Management, Project Management, Implementation (Code), Integration, Testing, Deployment (Installation) and Maintenance. The result shows that most of the published articles highly focused on analysis, design, testing, integration and implementation, which also synonymous with code, respectively. Analysis is the most researched topic, which shows the need for upfront software requirements gathering before embarking upon design, and coding in ALE/ELE, whereas software testing, design and integration are also demanding considerable attention in the field.

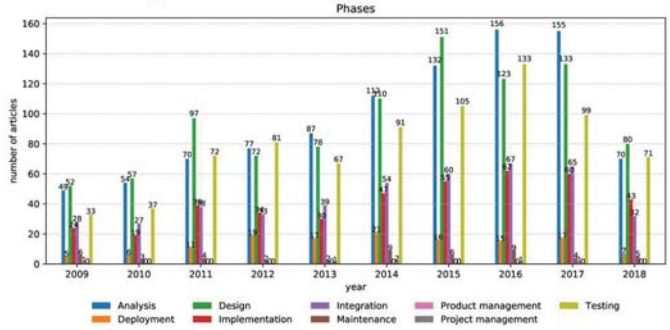


Fig. 9. Article distribution per year and properties in Phases property group.

The third property type as exhibited in Fig. 10 is "Critical Systems", which includes Business Critical, Mission Critical and Safety Critical. The finding shows that Software Development for ALE/ELE and Software Development and apps for Older People do not specifically identify themselves as business, mission, and/or safety critical key words.

Next, Fig. 11 shows how different properties are related between each other in terms of how often they occur together in the same article. These graphs can be used for guiding the drilling down process and selection of articles that need to be analyzed manually. The darker an edge is, the more articles that have the connected keywords. Also, it shows that some properties are not often encountered with others (e.g. Prototyping and Deployment on Fig. 11.). The two figures clearly depict that the iterative and prototyping software development methods are intensively used in the engineering of software for ALE/ELE. These terms are highly connected with the software development lifecycle.

Finally, Fig. 12 exhibits how authors from different countries collaborated. This graph lucidly shows that there is collaboration among authors from different countries and continents. In most cases, we attribute this to geographical location, smaller language barriers, or both.

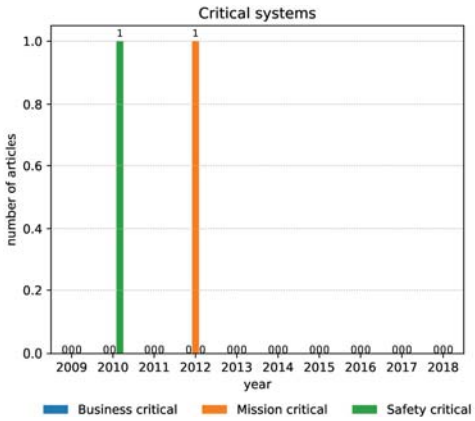


Fig. 10. Article distribution per year and properties in Critical Systems property group.

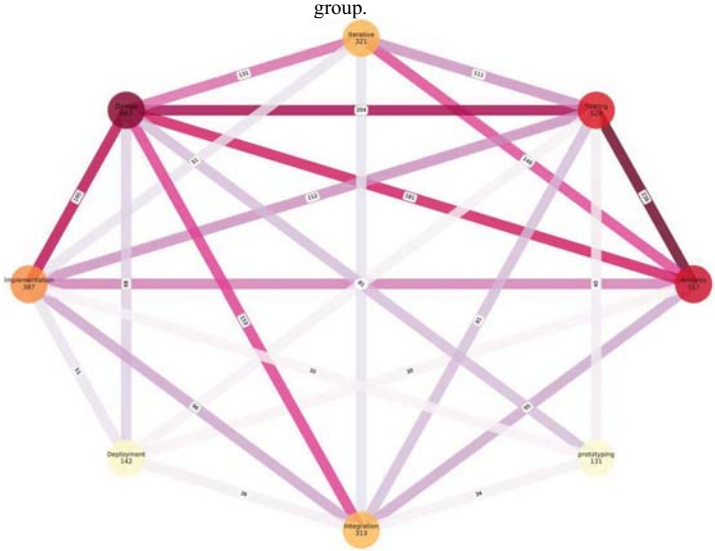


Fig. 11. Graph visualization with circular layout relevant articles by properties. Node labels show the property and number of articles that contain it and edge label shows the number of papers that have the properties it connects.

V. DISCUSSIONS

Software Development methods can be classified as either a heavyweight or a lightweight method [33]. The heavyweight methods, Plan-driven methods, or Traditional Software Development Methods (TSDMs), usually focus on comprehensive planning, complex documentation, and big design up-front. Conversely, the lightweight methods, also known as Agile methods, concentrate (1) more on individuals and interactions than processes and tools, (2) more on working software than comprehensive documentation, (3) value customer collaboration more than contract negotiation, and (4) focus more on responding to change than following a plan [34].

TSDMs, which invests significant time of system development in requirements engineering, have been increasingly employed in Safety/Mission-Critical software engineering such as ALE/ELE. This is due to the fact that these systems have stringent requirements validation and verification from industries. On the other hand, Agile software methods, which pays a considerable attention to working software without clear upfront definition of requirements, has not at least popularly and widely been used in such software engineering endeavors. Accordingly, the present research was intended to review the software methodologies that are in use for ALE/ELE software engineering by employing an NLP toolkit, which implements the PRISMA model.

Hence, an increasing trend over different search keywords over the last decade (see Fig. 6) was observed. In all searched keywords (Software Development for Ambient and assisted living, Software Development for Enhanced Living Environment, Software for Older People, and Apps for Older People) showed increasing trends from year to year except Apps for Older People, which slightly dropped in 2017. As depicted in Fig. 8, this research has also uncovered Iterative development and prototyping software development methodologies are the highly sought-after methodologies in engineering of software for ALE/ELE. In addition, agile development methodology is on the rise in the field.

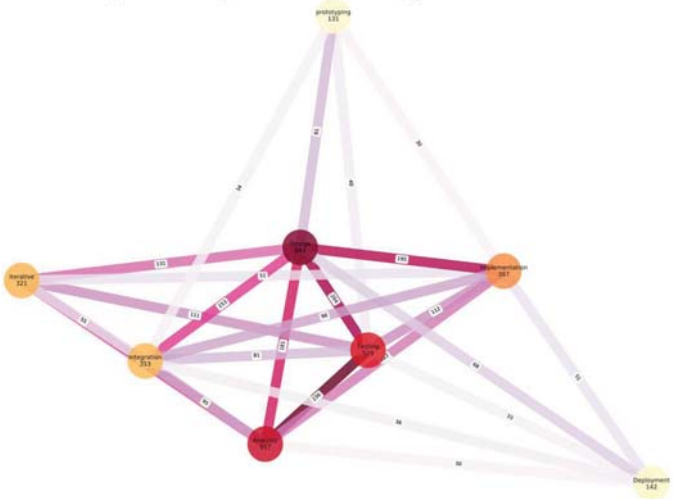


Fig. 12. Graph visualization with spring (i.e. Fruchterman-Reingold) layout relevant articles by properties.

In this paper, the NLP toolkit was found invaluable to identify software development method for ALE/ELE research focuses for the last decade. It was also applied to simplify the review process in several previous works [26]. The NLP toolkit can be used by many researchers to ease the review processes of the vast amount of research publications in our discipline, Software Engineering. By being able to reuse intermediate results and allowing tweaking and fine-tuning of keywords and properties, the researcher can test different alternatives of keywords and properties very quickly. The toolkit also provides ability to fine-tune the graph plotting thresholds, so they can show appropriate number of edges. These default parameters were empirically determined based on extensive analysis with over dozens of different use-cases.

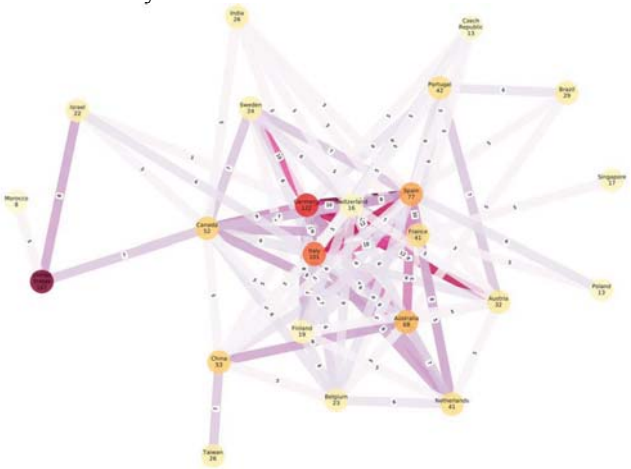


Fig. 13: Graph visualization of relevant articles by countries. Node labels show the country and number of publications from it, while edge labels show the number of papers that were published by authors with affiliations from the countries it connects.

VI. CONCLUSION AND FUTURE WORKS

In this paper we used NLP toolkit to speed up the process of surveying scientific articles and trend analysis Meta studies. By leveraging NLP, it facilitates a robust and comprehensive eligibility analysis of papers, so the user can focus on reading a small number of potentially relevant papers. The toolkit was able to analyze the abstracts of nearly seven thousand papers automatically and visualize different trends of interest.

In this research, based on the obtained data, we can conclude that almost all of the searched keywords and properties have an increasing trend except critical systems. The processed articles show that the research community are more interested in researching Software and Apps Development for Enhanced living environments by using iterative and prototyping software development methodologies. The finding also lucidly exhibits that agile software development methodology is being adopted in the engineering of software for elderly people. In addition, the research community in the area are not designating the research articles by business, mission and/or safety critical keywords. Furthermore, substantial research articles have invested in the requirements engineering (analysis) phase of the software development life cycle for elderly people. However, it is

imperative that the research community and software development industry and practitioners exert relentless efforts at least to tap on the "sweet Spot" of agile development methodologies, which brought success from small to large to very large software development projects, in the software development for ambient and enhanced living environment.

Researchers in software engineering discipline can benefit by employing NLP Toolkit to minimize the daunting work of reviewing the voluminous publications. For future work, we will expand keywords and increase the years to investigate into more research works and corroborate the current findings.

ACKNOWLEDGMENT

This work was supported by FCT project UID/EEA/50008/2013 (Este trabalho foi suportado pelo projecto FCT UID/EEA/50008/2013).

This article/publication is based upon work from COST Action IC1303 - AAPELE - Architectures, Algorithms and Protocols for Enhanced Living Environments and COST Action CA16226 - SHELD-ON - Indoor living space improvement: Smart Habitat for the Elderly, supported by COST (European Cooperation in Science and Technology). More information in www.cost.eu

REFERENCES

[1] J. Hatcliff, A. Wassyng, T. Kelly, C. Comar, and P. Jones, "Certifiably safe software-dependent systems: challenges and directions," *Proc. Futur. Softw. Eng. - FOSE 2014*, pp. 182–200, 2014.

[2] E. S. Grant, "Requirements engineering for safety critical systems: An approach for avionic systems," *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 991–995, 2017.

[3] TechoPedia, "What is a Mission Critical System?" [Online]. Available: https://www.techopedia.com/definition/23583/mission-critical-system. [Accessed: 10-Dec-2018].

[4] P. A. Laplante and J. F. Defranco, "Software Engineering of Safety-Critical Systems: Themes from Practitioners," *IEEE Trans. Reliab.*, vol. 66, no. 3, pp. 825–836, 2017.

[5] R. Kasauli, E. Knauss, B. Kanagwa, J. K. Balikuddembe, A. Nilsson, and G. Calikli, "Safety-Critical Systems and Agile Development: A Mapping Study," *2018 44th Euromicro Conf. Softw. Eng. Adv. Appl.*, pp. 470–477, Aug. 2018.

[6] L. Provenzano and K. Hänninen, "Specifying software requirements for safety-critical railway systems: An experience report," in *Int. Working Conf. on Requirements Engineering: Foundation for Software Quality*, 2017, vol. 10153 LNCS, no. January, pp. 363–369.

[7] M. Vuori, *Agile Development of Safety-Critical Software*. 2011.

[8] B. Boehm and R. Turner, "Using risk to balance agile and plan-driven methods," *Computer (Long. Beach. Calif).*, vol. 36, no. 6, pp. 57–66, 2003.

[9] K. Aguanno, *Managing agile projects*. Multi-Media Publications, 2005.

[10] S. Jalali and C. Wohlin, "Global software engineering and agile practices: a systematic review," *J. Softw. Evol. Process*, vol. 24, no. 6, pp. 643–659, Oct. 2012.

[11] I. Sommerville, *Software Engineering*, 9th ed. Boston: Addison-Wesley, 2011.

[12] K. Schwaber, "Agile project management with Scrum," Microsoft Press, 2004.

[13] K. Schwaber and M. Beedle, *Agile software development with Scrum*. Upper Saddle River: Prentice Hall, 2002.

[14] M. Poppendieck and T. D. Poppendieck, *Lean software development : an agile toolkit*. 2003.

[15] M. Paasivaara, "Adopting SAFe to scale agile in a globally distributed organization," *Proc. - 2017 IEEE 12th Int. Conf. Glob. Softw. Eng. ICGSE 2017*, pp. 36–40, 2017.

[16] P. Pelliccione, E. Knauss, R. Heldal, S. Magnus Ågren, P. Mallozzi, A. Alminger, and D. Borgentun, "Automotive Architecture Framework: The experience of Volvo Cars," *J. Syst. Archit.*, vol. 77, pp. 83–100, 2017.

[17] C. Ebert, M. Paasivaara, D. J. Reifer, F. Maurer, and H. Erdogmus, "Scaling agile methods," *IEEE Softw.*, vol. 34, no. 6, pp. 98–103, Nov. 2017.

[18] European Cooperation in Science & Technology, "Action CA16226 - COST." [Online]. Available: https://www.cost.eu/actions/CA16226/#tabs%7CName:overview. [Accessed: 10-Dec-2018].

[19] D. Moher, A. Liberati, J. Tetzlaff, and D. Altman, "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement," PLos Medicine, Ottawa, Canada, Jul. 2009.

[20] D. Moher, L. Shamseer, M. Clarke, D. Ghersi, A. Liberati, M. Petticrew, P. Shekelle, and L. A. Stewart, "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement," *Syst. Rev.*, vol. 4, no. 1, p. 1, Dec. 2015.

[21] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering," Durham, UK, 2007.

[22] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Inf. Softw. Technol.*, vol. 85, pp. 1339–1351, 2017.

[23] D. Moher, L. Stewart, and P. Shekelle, "All in the Family: systematic reviews, rapid reviews, scoping reviews, realist reviews, and more," *Syst. Rev.*, vol. 4, no. 1, p. 183, Dec. 2015.

[24] M. D. J. Peters, C. M. Godfrey, H. Khalil, P. McInerney, D. Parker, and C. B. Soares, "Guidance for conducting systematic scoping reviews," *Int. J. Evid. Based. Healthc.*, vol. 13, no. 3, pp. 141–146, 2015.

[25] E. Zdravevski, P. Lameski, V. Trajkovik, and I. Chorbev, R. Goleva, N. Pombo, N. Garcia," Automation in systematic, scoping and rapid reviews by an NLP toolkit: a case study in Enhanced Living Environments" in Algorithms, Architectures and Platforms for Enhanced Living Environments (AAPELE), Springer, 2019

[26] A. Alla, E. Zdravevski, V. Trajkovik.: Framework for aiding surveys by natural language processing," in *Web proceedings of the ICT Innovations 2017 conference*, 2017.

[27] E. Zdravevski and A. Kulakov, "System for prediction of the winner in a sports game," in *Davcev, D., Gomez, J.M. (eds.) ICT Innovations 2009.*, 2010, pp. 55–63.

[28] E. G. K. Thiruvathukal and B. J. D. Hunter, "2011Namikawa_BTO_FE_SXRL_J.pdf," pp. 90–95, 2007.

[29] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," *Proc. 7th Python Sci. Conf.*, no. SciPy, pp. 11–15, 2008.

[30] J. J. Webster and C. Kit, "Tokenization as the initial phase in NLP," in *Proceedings of the 14th conference on Computational linguistics* -, 1992, vol. 4, p. 1106.

[31] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.

[32] S. Bird, "NLTK: The Natural Language Toolkit Steven," in *Proceedings of the COLING/ACL on Interactive presentation sessions*, 2006, pp. 69–72.

[33] J. J. Cho, "An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum," *All Grad. Theses Diss.*, p. 599, 2010.

[34] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for Agile Software Development," *Agile Alliance*, 2001. [Online]. Available: https://moodle2016-17.ua.es/moodle/pluginfile.php/80324/mod_resource/content/2/agile-manifesto.pdf. [Accessed: 01-Oct-2018].