Using OCR to read handwritten texts in search for NFRs in Agile Software Engineering

Richard R. Maiti
Department of Computer Science
Berea College
Berea, KY
maitir@berea.edu

Aleksandr Krasnov
Department of Computer Science
Berea College
Berea. KY
krasnova@berea.edu

Deanna Marie Wilborne
College of Engineering & Computing
Nova Southeastern University
Fort Lauderdale, FL
mw1313@mynsu.nova.edu

Frank Mitropoulos
College of Engineering & Computing
Nova Southeastern University
Fort Lauderdale, FL
mitrof@nova.edu

***Abstract*— Non-Functional requirements (NFRs) are ignored in agile software engineering and Functional Requirements (FRs) often take center stage during agile software development due to the nature of agile software. Research shows neglecting NFRs can be expensive. The Capture Elicit Predict (CEP) methodology which is an extension of the NERV and NORMAP methodology utilized OCR to extract requirement texts from images. Additionally, the CEP methodology utilized the NFR Locator and utilized the Chung's NFR framework to categorize NFRs. The extension of the CEP methodology includes the extraction of NFRs from handwritten text that are written by stakeholders and team members during the beginning stages of agile software development, NFRs can be identified early. Research has shown that NFRs are helpful and compliment FRs during the beginning stages of agile software engineering. This research shows that using NFRs from handwritten texts from requirements 3 x 5 cards can be beneficial in the agile software development process. The OCR recognition of handwriting was 74.13%. The accuracy of the NFRs that were captured were 92%.**

Keywords—Agile Software Engineering, Capture Elicit Prioritize, CEP, Functional Requirements, Non-Functional Requirements, NFRs, FRs, NERV, NORMAP, Optical Character Recognition, OCR

INTRODUCTION

Agile Development Process (ADP) such as Scrum and XP ignore Non-Functional Requirements (NFRs) due to management's focus on Functional Requirements (FRs) (Ramos et. al, 2018). Short iterations of ASD and quick delivery of software often does not consider important NFRs such as security and other NFRs (Wang et. al, 2018). Research has shown that NFRs are often not considered during the agile software development process (Nguyen, 2009). NFRs are system behaviors that are starting to gain precedence and given the same importance as Functional Requirements (FRs). NFRs describe the characteristics of a system whereas FRs describe the functionality of a system (Ameller, 2012). In order for a software system to be affluent its entire life, the FRs and NFRs should both be considered (Slankas & Williams, 2013). Stakeholders and software developers should agree to include both FRs and NFRs in order for the software system to be successful (Danylenko and Lowe, 2013) (Poort et. al, 2012). Software systems are becoming more complex and are being deployed on multiple devices increasing total complexity and the possibility of unintended behaviors – it is, therefore, important to take NFRs seriously in the beginning of agile software development (Yin & Jin, 2012) and during the entire lifetime of the software system. Data has shown that not considering NFRs has resulted in a failure rate of greater than 60% (Fabio et. al, 2013) (Bajapi & Gorthi. 2012).

The nature of Scrum, a framework for agile software development, relies on developing software quickly and therefore FRs are only taken into considerations (Farid & Mitropoulos, 2012). However, NFRs are now being taken into consideration due to research that shows NFRs can be equally beneficial as FRs (Saadatmand et. al, 2012) (Affleck et. al, 2012) (Farid & Mitropoulos, 2012) (Liu et. al, 2012). Research shows that the consideration of NFRs can dramatically reduce software defects and increase reliability of software during the lifetime of the software system (Cao et. al, 2013).

Historical NFRs have been shown to be beneficial in predicting NFRs in agile software development (Maiti et al., 2018). Historical data can be used to predict NFRs based on a decision tree (Maiti et al., 2018). NFRs that appear multiple times in an iteration can be beneficial in predicting NFRs on next iterations of agile software development (Maiti et al., 2018). The predicted NFRs data can be beneficial for agile development team members for developing secure code (Maiti et al., 2018).

Incorporating NFRs from handwritten text from developers can be beneficial in developing agile software. Research has shown that recognizing handwritten text is still a challenge (Cao et. al, 2011). However, grouping handwriting from several writers with the same character sets has shown significant improvements in recognition (Alvaro et. al, 2013). Software development meetings can be informal at times where electronic sources to capture NFRs may not be available or electronics such as smart phone, tablets and laptops can fail. The nature of agile software development is to capture requirements on 3 x 5 cards. Capturing important metadata such as NFRs from 3 x 5 index cards can be beneficial in developing agile software.

RESEARCH GOALS AND RESEARCH QUESTIONS

Reseach Goals

This research extends Capture Elicit and Prioritize (CEP) methodology to capture NFRs from handwritten texts (Maiti & Mitropoulos, 2017a). NFRs are left behind until the later process of agile software engineering due to the steps of agile software engineering which takes FRs. Taking NFRs as well as FRs in the beginning stages has been proven to have an impact on producing reliable software. Extending the CEP methodology (Maiti & Mitropoulos, 2017b) to include OCR to recognize NFRs from handwritten texts by taking the handwritten 3 x 5 card requirements from a senior college project.

Research Questions

This research answers the following question:

RQ: Can OCR be utilized to capture NFRs from 3 x 5 index cards? If so, how accurate is the information captured?

BRIEF LITERATURE ON OPTICAL CHARACTER RECOGNITION

Optical Character Recognition of handwriting is still a challenge and research has shown that grouping known characters is more reliable than writing recognition that is not supervised (Cao et. al, 2011). The k-nearest neighbor classifier is applied to writer identification texts in handwritten document images which shows an error rate of 1.5% from 650 writers on 1500 pages of handwritten data (Cao et. al, 2011). In most cases, there is not sufficient handwritten data available to train to recognize each data set for each writer (Cao et. al, 2011). There are different adaptation techniques that are used in OCR such as Maximum Likelihood Linear Regression (MLLR) and Maximum A Posteriori (MAP) (Cao et. al, 2011). MAP adaption is better for more amounts of training data set where as MLLR is better suited for adapting to multiple images of the same character sets (Cao et. al, 2011). In the research conducted by (Cao et. al, 2011) trained the writer identification system using 259 writers. The data shows the accuracy of recognizing handwritten OCR improved by adding more pages of handwritten text to the training set (Cao et. al, 2011).

OCR is also utilized to teach children how to write (Alvaro et. al, 2013). The children write handwriting into a stylus with the appropriate letter and a program recognizes which letter has been written to provide feedback to the user (Alvaro et. al, 2013). The software has 185 samples of the letter and greater samples have reduced the errors in recognizing the character (Alvaro et. al, 2013). OCR was utilized as a tutorial for children students and to provide immediate feedback on the written handwriting (Alvaro et. al, 2013).

METHODOLOGY

In previous research the Capture Elicit Prioritize (CEP) methodology extended NERV and NORMAP (Maiti & Mitropoulos, 2015) (Maiti & Mitropoulos, 2017a)(Maiti & Mitropoulos, 2017b)(Domah, 2013)(Farid, 2011). The CEP, NERV and NORMAP methodologies utilized the EU eProcument requirements document (European Dynamics S.A. vol. 1, 2005) (European Dynamics S.A. vol 2, 2005). CEP was successful in identifying 56 out of 57 requirement sentences and successfully elicited 98.24% of the baseline. This is an improvement of 10.53% over the NORMAP and 1.75% over the NERV methodologies (Maiti & Mitropoulos, 2017a)(Maiti & Mitropoulos, 2017b). CEP methodologies NFRs count was 86 out of 88, an improvement of 12.49% over NORMAP and 4.55% over NERV (Maiti & Mitropoulos, 2017a)(Maiti & Mitropoulos, 2017b).

The Capture Elicit Prioritize (CEP) captures potential NFRs by using OCR on requirement images NORMAP (Maiti & Mitropoulos, 2015) (Maiti & Mitropoulos, 2017a) (Maiti & Mitropoulos, 2017b). In the elicit step, the NFR Locator plus (NFRL+) takes sentences from requirement documents that are placed in distinct categories utilizing the *k*-NN classification algorithm [3]. The Chung's NFR framework is used to categorize the NFRs utilizing a set of keywords that are trained to locate NFRs (Maiti & Mitropoulos, 2017a) (Maiti & Mitropoulos, 2017b). The αβγ-framework was used to prioritize the NFRs, the flexibility of this framework allows agile members to substitute any parts of the framework with other processes (Maiti & Mitropoulos, 2017a) (Maiti & Mitropoulos, 2017b). This research uses the prototype research method.

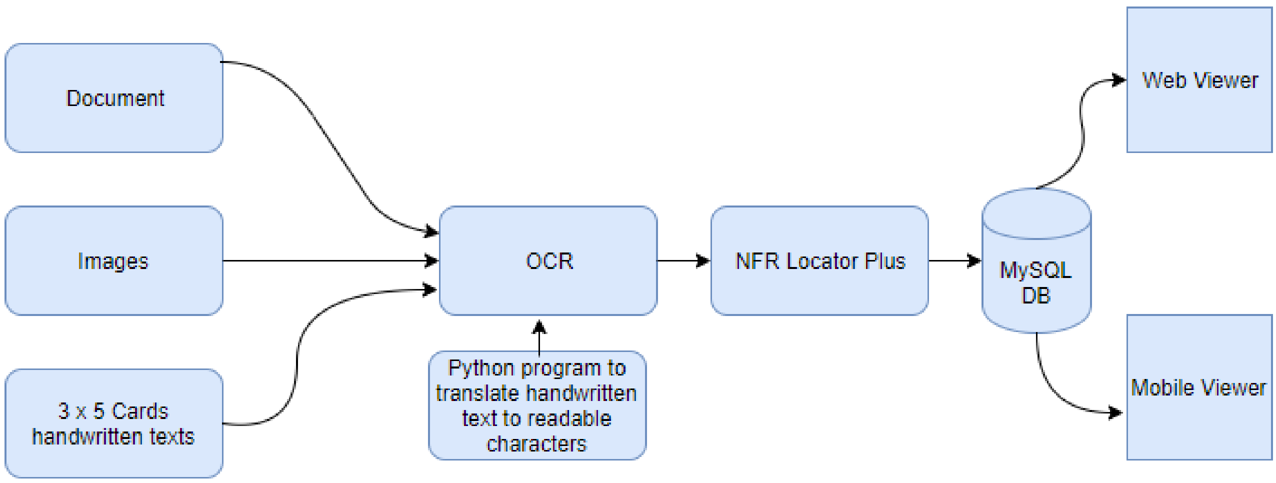

Figure 1. CEP methodology with readable hand written text

This research takes a college senior project and captures handwritten requirements as a set of 3 x 5 index cards. As shown in figure 1 above, the handwritten text is integrated in the capture part of the CEP methodology and OCR along with the Python program to recognize handwritten characters and translate to text for the NFRL+. The handwritten requirement texts are extracted from the 3 x 5 cards using OCR and the NFRL+ is utilized to identify potential NFRs from the 3 x 5 index cards. The potential NFRs are validated using past NFRs data.

RESULTS

This section covers the results of the NFRs captured from handwritten 3 x 5 requirements card. The first step involved taking pictures of the 3 x 5 requirements cards. In this step an android smart phone was used to capture the handwritten texts. There were several 3 x 5 index cards with requirement information written. Each card was photographed, individually, and the data was downloaded on to a laptop PC. There were several 3 x 5 index cards that contained requirements for a website.

The first step involved utilizing OCR to translate the handwritten texts to readable characters. The script (Krasnov, 2018) was written in Python and required training to recognize the letters and numbers. The script is available for download (Krasnov, 2018). One python program was used to recognize the numbers. The numbers were used for numbering the requirements in sequence as given by the client. Another python program was written to recognize the characters. In a requirements gathering setting, there may be multiple stakeholders with different handwritings. In the case described here, we are dealing with one set of handwritten text. As shown in figure 2 below, the training data was used to improve the recognition of letters. There were several sets of training data that was used to improve the recognition of characters (European Dynamics S.A. vol. 1, 2005) (European Dynamics S.A. vol. 2, 2005). The algorithm used for training the set of characters was the k-nearest neighbor. The accuracy of character recognition was 74.13 %. Past OCR research has shown that the more training improves the recognition (Alvaro et. al, 2013).

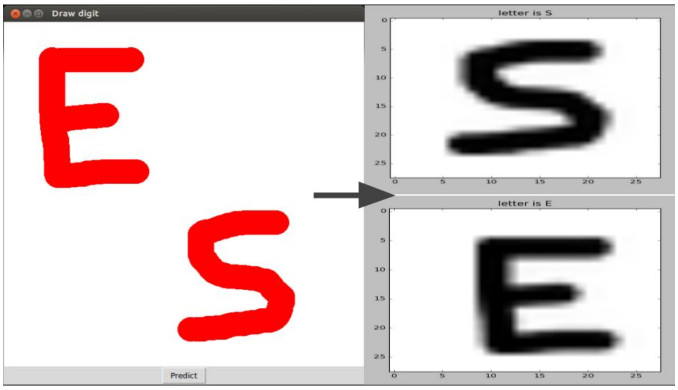

Figure 2. Training data set

TABLE I. CEP METHODOLOGY NFR RESULTS DATA

NFR	**Occurrences**
	Number
Accessibility	4
Accuracy	5
Confidentiality	10
Configuration	0
Documentation	0
Efficiency	0
Interoperability	2
Legal	0
Performance	3
Reliability	0
Scalability	1
Security	3
Usability	6
User Interface	4

The next step, is to utilize the NFRL+ locator to determine if there are any potential NFRs in the requirements that were gathered. Table I CEP Methodology NFR Results Data above, shows NFRL+ was able to pick up potential NFRs. For accessibility, NFRL+ picked up the following words: “listing”, “available”, “already”, “added”. In table II NFR keyword by category below, the words that NFRL+ picked up for all the NFRs are shown. For NFR accuracy NFRL+ picked up words such as “log” and “login”. For confidentiality there were several key words that NFRL found such as “data”, “information” and “update”. For interoperability the key words found were “available” and “service”. The words “time”, “week” and “maintain” were found for performance. For scalability the word “available” was found. For security, the “http” keyword was found by NFRL+. Some of the words that NFRL+ picked up were not correct. Such as the word “logo” for accuracy does not belong to the NFR accuracy.

TABLE II. NFR KEYWORD BY CATEGORY

NFR	**Key NFRs** *Words*
Accessibility	listing, available, already, added
Accuracy	logo, logo, log, Already, login
Confidentiality	Data, common, information, updates, information, update, update, update, information, information
Configuration	-
Documentation	-
Efficiency	-
Interoperability	available, services
Legal	
Performance	time, week, maintain
Reliability	
Scalability	Available
Security	http, http, http
Usability	-
User Interface	-

In table III below NFR Results, the validation was done by taking the keywords and validating it with previously validated NFRs. For accessibility, “already” is not a NFR. NFRL+ was looking for the keyword “read” and instead picked up the word “already”. For accuracy, NFRL+ picked up the word “logo” twice which is an incorrect NFRs.

TABLE III. NFR RESULTS

NFR	**Key NFRs**
	Words
Accessibility	3 out of 4correct
Accuracy	3 out of 5 correct
Confidentiality	9 out of 9
Configuration	N/A
Documentation	N/A
Efficiency	N/A
Interoperability	2 out of 2 correct
Legal	N/A
Performance	3 out of 3 correct
Reliability	
Scalability	1 out of 1 correct
Security	1 out of 1 correct
Usability	N/A
User Interface	N/A

The total correct NFRs were 23 out of 25 which an accuracy of 92%. NFRL+ did not pick up any NFRs for the configuration, documentation, efficiency, reliability, usability and user interface.

CONCLUSION & FUTURE STUDIES

The research answered the following question:

RQ: Can OCR be utilized to capture NFRs from 3 x 5 index cards? If so, how accurate is the information captured?

This research examined whether OCR of handwritten text were beneficial in capturing NFRs. The accuracy of character recognition was 74.13%. For the NFRs that were captured, there is an accuracy rate of 92%. These were validated from previously captured NFRs. This is the first research that examined capturing handwritten text to extract NFRs. The OCR recognition needs to be improved. This can be done with more training with more sample handwritten data.

There are times where digital tool such as smart phones and laptops can fail. The failure can come from the battery life of digital devices or other mishaps that may occur. The failure can result in a loss of important data which could be costly to the project. It is cost effective to write down important design ideas on a simple 3 x 5 card. It becomes essential for developers to write down NFRs on non-digital devices such as a 3 x 5 card. The application can be used by developers and stakeholders to communicate requirements with the team. For future studies, the application can be extended to be an app on a mobile phone where multiple handwritten scripts can be captured and to have multiple agile teams to use historical data and to compare NFRs. Another future study could examine the benefits of having this additional metadata from agile development team during brain storming sessions and how beneficial and cost effective this additional metadata could be to the project.

Reference List

Affleck, A., Krishna, A., & Achuthan, N. R. (2013). Optimal Selection of Operationalization for Non-Functional Requirements. Proceedings of APCCM: The 9th Asia Pacific Conference on Conceptual Modeling, Darlinghurst, Australia, 69-78.

Alvaro, A., Dela, R., Cruz, R., Fonseca, D. & Samonte, M. (2010). Basic handwriting for kids using OCR as an evaluator. International Conference on Networking and Information Technology, Manila, Philippines, 265-268.

Ameller, D., Ayala, C., Cabot, J. & Franch, X. (2012). How do software architect consider non-functional requirements: An exploratory study. *The 20th International Requirements Engineering Conference*, Chicago, IL, 41-50.

Bajapi, V. & Gorthi, R. P. (2012). On non-functional requirements: A Survey. *International Conference on Computer Networks and Information Technology (ICCNIT'12)*, Abbottabad, Pakistan, 2012, 333-340. DOI=10.1109/SCEECS.2012.6184810

Cao, H., Prasad, R. & Natarajan, P. (2011). OCR-Driven Writer Identification and Adaptation in an HMM Handwriting Recognition System, *International Conference on Document Analysis and Recognition*, Beijing, China, 739 – 743.

Danylenko, A. & Lowe, W. (2012). Context-aware recommender systems for non- functional requirement. *The 3rd International Workshop on Recommendation Systems for Software Engineering*, Zurich, Switzerland, 80-84. DOI=10.1109/RSSE.2012.6233417

Domah, D. (2013). The NERV methodology: Non-functional requirements elicitation, reasoning and validation in agile processes (Doctoral Dissertation) Available from ProQuest Dissertation and Thesis database (UMI No. 3594275)

European Dynamics S.A. (2005). Functional Requirements for Conducting Electronic Public Procurement Under the EU Framework (Volume 1). [Online] Available:http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/functional-reguirements-vol1_en.pdf. (URL)

European Dynamics S.A. (2005). Functional Requirements for Conducting Electronic Public Procurement Under the EU Framework (Volume 2). [Online] Available: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/functional- reguirements-vol2_en.pdf. (URL)

Fabio, S., Lucena, M., & Lucena, L. (2013). STREAM-AP: A Process to systematize architectural patterns choice based on NFR. *TwinPeaks: The 3rdInternational Workshop on Twin Peaks of Requirement and Architecture*, Rio de Janeiro, Brazil, 27-34.

Faird, W. M. (2011). The NORMAP methodology: Non-functional requirements modeling for agile processes (Doctoral Dissertation) Available from ProQuest Dissertation and Thesis database. (UMI No. 3460458)

Farid, W. M. & Mitropoulos, F. J. (2012). Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes. Proceedings of IEEE: Southeastcon, Orlando, FL, 1-7.

Krasnov, A. (2018, February 27). OCR for handwritten translation. https://github.com/krasnova/OCR

Liu, Y., Zhiyi, M., Qiu, R. Chen, H., & Shao, W. (2012). An approach to integrating non-functional requirements into UML design models based on NFR-specific patterns. Proceedings from QSIC: The 12th International Conference on Quality Software, Shaaxi, China, 132-135. Y. Saito., K. Matsumoto, and A Modern Evaluation of non-functional requirements in a Request for Proposal (RFP). *The 22nd International Workshop on Software Measurements and the 7th International Conference on Software Process and Product Measurement*, Assisi, Italy, 2012, 106-111.

Maiti, R. R., Krasnov, A. & Wilborne, D. M. (2018). Agile Software Engineering & The Future of Non-Functional Requirements, *Journal of Software Engineering Practice* 2(1).

Maiti, R. R. & Mitropoulos, F. J. (2015). Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development. *IEEE, (Southeastcon '15)*, Ft. Lauderdale, FL, 1-8. DOI=10.1109/SECON.2015.7133007

Maiti, R. R. & Mitropoulos, F. J. (2017a). Capturing, eliciting, and prioritizing (CEP) NFRs in agile software engineering, *IEEE SoutheastCon*, Charlotte, NC, 1-7.doi: 10.1109/SECON.2017.7925365

Maiti, R. R. & Mitropoulos, F. J., (2017b). Prioritizing Non-Functional Requirements in Agile Software Engineering. In Proceedings of the SouthEast Conference (ACM SE '17). ACM, New York, NY, 212-214. DOI: https://doi.org/10.1145/3077286.3077565

Nguyen, Q. L. (2009). Non-functional requirements analysis modeling for software product lines. *ICSE Workshop on Modeling in Software Engineering, IEEE Computer Society*, Washington, DC, pp. 56-61. DOI= http://dx.doi.org/10.1109/MISE.2009.5069898

Poort, E.R. Key, A., With, P.H.N. & Vilet, H. (2012). Issues dealing with non-functional requirements across the contractual divide. *WICSA and ECSA: Software Architecture and European Conference on Software Architecture*, Helsinki, Finland, 315-319.

Ramos, F., Costa A., Perkusish, M., Almeida, H. & Perkusish, A. (2018). A Non-Functional Requirements Recommendation System for Scrum-based Projects. *The 30th International Conference on Software Engineering & Knowledge Engineering,* Redwood City, San Francisco Bay, California, USA, DOI=1 0.18293/SEKE2018-107.

Saadatmand, M., Cicchetti, A., & Sjodin, M. (2012). Toward model-based trade-off analysis of non-functional requirements. Proceedings from SEAA : The 38th Conference on EUROMICROS, Cesme, Turkey, 142-149.

Slankas, J., & Williams, L. (2013). Automated extraction of non-functional requirement in available documents. *1st International Workshop on Natural Language Analysis in Software Engineering*, San Francisco, CA, 9-16. DOI= 10.1109/NAturaLiSE.2013.6611715

Wang, W., Gupta, A. & Niu, N. (2018). Mining Security Requirements from Common Vulnerabilities and Exposures for Agile Projects, *2018 IEEE 1st International Workshop on Quality Requirements in Agile Projects (QuaRAP)*, Banff, AB . 6-9, doi: 10.1109/QuaRAP.2018.00007

Yin, B. & Jin, Z. (2012). Extending the problem frames approach for capturing non-functional requirements. *The 9th International Conference on Computer and Information Sciences*, Shanghai, China, 432-437.