**Towards Understanding Emotional Response to Requirements Changes in Agile Teams**

Authors : Madampe, Kashumi and Hoda, Rashina and Singh, Paramvir

Conference : International Conference on Software Engineering, ICSE2020

**Introduction**

Agile software development accepts improvements to requirements even close to the deadline, "Welcoming changing demands, even late in development" as one of its principles. This is one of the main rule of agile which make him different from traditional way. It is not so easy for the organization to work like that they have to face many problems. However, it is argued that reacting to and embracing change requires not just addressing strategic problems but also coping with emotional factors, which is easier said than done. Although agile practices may provide a structure for managing change, the human aspects of reacting to change are still not understood. That is why this study worked on emotional level of the employees.

Emotions serve as motivators of action and have a relation with cognition, efficiency, and decision making. Emotion studies are becoming more and more centered in software engineering. Nevertheless, emotions of agile teams have not been studied in response to changes in requirements, leading to the question, "how do agile teams react emotionally to changes in requirements (RCs)? This is the main aim objective of this study. Therefore it is very interesting to research how they react to these changes emotionally.

**Methodology**

In this study, they (i) implement a novel hybrid method by using Grounded Theory and Sentiment Analysis,(ii) present three distinct stages by emotional responses, a description of emotions, and differences in emotional and sentimental polarity at each stage,(iii) highlight the need for agile teams to take emotional responses into account when applying agile concepts and practices to tackle change. They collected the data from participants in a manner were asked to fill out a pre-interview questionnaire to gather details about the demographics, project, and team. The interviews were then conducted face-to- face from 7 participants and online from 3 participants. Each interview lasted on average about 50-60 minutes. For data analysis a mixed approach was taken to evaluate the qualitative data obtained in three phases.

**Results**

Using a novel hybrid method of grounded theory and sentiment analysis, they identified three phases of change in requirements (RC), where participants documented agile teams that responded emotionally: receiving, implementing, and delivering an RC. The findings of this pilot study show that while agile is based on accepting change, agile teams are going through emotional ups and downs in reacting to changes in requirements. Hence it is important to consider emotional dimensions to understand and develop how agile teams react to changes in action.

**Second article**

**A Study on the Prevalence of Human Values in Software Engineering Publications, 2015 – 2018**

**Authors:** Perera, Harsha and Nurwidyantoro, Arif and Hussain, Waqar and Mougouei, Davoud and Whittle, Jon and Shams, Rifat Ara and Oliver, Gillian

**Conference:** International Conference on Software Engineering, ICSE2020

**Journal:** arXiv preprint

**Introduction**

When understanding of human dimensions of software increases, software that accounts for its principles is increasingly demanding by the public. Failure to account for human values in software ( e.g., equality and fairness) can result in user dissatisfaction and negative socioeconomic effects. It can result in adverse socio-economic impacts such as financial loss and damage to reputations. This study argues that work and practice in software engineering is actually paying inadequate attention to certain human values. It may be due to the lack of sufficient technological and methodological support for the software engineering principles. To provide evidence for this argument, they have reviewed research papers on software engineering (SE) as part of their broader approach to the analysis of human values to determine how much consideration the SE sector has paid to values.

**Methodology**

They listed publications from top-tier SE conferences and journals manually based on their importance to different values. This was focused on multiple raters manually classifying the paper abstracts. Classification based on abstracts is sub-optimal, instead of reading the entire article, but finds a correlation between the quality and the time required for the study. All papers had several raters, and Fleiss' Kappa was used to calculate inter-rater cooperation.

They have chosen to list papers from the last four years of conferences and journals that are widely regarded as top SE venues, namely, the International Conference on Software Engineering (ICSE), the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), the IEEE Transactions on Software Engineering (TSE), and the ACM Transactions on Software Engineering and Methodology (TOSEM). The pilot and the main analysis all used SVS as the scheme for classification. Totally there were seven rates in which 5male and 2female were employed with different levels of SE relevant work experience, varying from doctoral students to professors, including one rater from outside the field of software engineering.

**Results**

In this paper, they explored the existence of a wider spectrum of human values in software engineering research involving trust, equality and social justice. Using Schwartz Values Structure as their classification system, 1350 newly published papers (2015–2018) were categorized into top-tier SE conferences and journals. They conclude that the human ideals are regarded by only a small proportion of SE studies. Although safety as a value category and privacy as a particular value stand out as the key focus in SE study, few other human values are considered, such as Helpful, Environmental Protection and Social Justice.

**Third article**

**What Predicts Software Developers' Productivity?**

**Authors**

Murphy-Hill, Emerson and Jaspan, Ciera and Sadowski, Caitlin and Shepherd, David and Phillips, Michael and Winter, Collin and Knight, Andrea and Smith, Edward and Jorde, Matt

**Publisher :** [IEEE Transactions on Software Engineering](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32)

**Introduction**

It is important to improve the productivity of the software developers. Organizations have a range of choices to help their software developers become their most successful version, from changing office layouts, investing in new resources, to cleaning up the source code. Yet which choices have the greatest impact? This is big question for the organization how to increase the productivity Organizations such as require empirical advice on which variables to seek to exploit to increase efficiency in the best possible way. In a perfect world, they should invest in a number of factors that increase profitability, but time and resources are limited so we need to invest selectively. The aim is to understand the factors influencing the efficiency of software developers in comparison.

**Methodology**

It is a qualitative study conducted data through interview. To accomplish the main goal of this study, they conceived a survey consisting of a set of productivity problems, a set of productivity factors, and a set of demographic variables. In a survey 622 developers across 3 companies asked about these productivity factors and self-assessed productivity.

**Results**

The findings indicate developers are comparable in some respects to software workers, and differ in others from them. For example, job enthusiasm was the best predictor of productivity for developers, but that indicator predicted productivity for analysts to almost the same degree. Their results indicate that the factors most strongly correlating with self-rated efficiency were non-technical factors, such as work enthusiasm, social encouragement for new ideas and valuable input on job results. The results of this study also show that self-rated productivity of software developers is more closely linked to the variety of tasks and the ability to work remotely relative to other knowledge workers.