A simple research and attempt to measure software engineering process

**Yuxiao Hu**

**19321099**

**Introduction**

For years, the manager in companies are trying to seek an appropriate way to precisely measure the contribution of certain employee or team and make decisions according to the result. Many companies use some common standard such as KPI (Key Performance Indicator) or OKR (Objectives and Key Results) to value employees. For software engineering, the measurement is more complex than other work. People can’t be judged simply by the hours worked on or the quantity of work such as LOC (line of code) or the number of commits to repository, these criteria are too shallow and the focus should lie on the efficiency of the code. But how to measure the efficiency of code? Some optional choice might be the memory usage, the running time or the number of line for a certain method, etc. In this report, I will try to express my understanding toward this topic and show some other people’s research toward this topic as well as ethic issues that may affect the measure to the software engineering.

**Body**

With the development of the Internet, companies are trying to bring advanced technology into the workspace to enhance the communication inside and outside the company as well as to help employees improve the quality and efficiency of their work. Bughin and Chui (2013) argued in their report that 90% of executives whose company use technologies gain measurable business benefit from employees, customers and partners. The benefit is not only related to the interaction between people, but also lies on single employee’s efficiency towards its own work. The success of a company can’t be depended on the management of executives merely, the effort of each employee should be taken into account as an inevitable factor. The productivity or efficiency of single employee should be cared and measured in order to form an effective team. The manager should both care the single person and the whole team’s productivity in order to rearrange the resources and optimize the development of project. The team’s productivity is not just the simple combination of each single employee, a good team would make ‘1+1>2’ in the real situation. In addition, the people around employee would make influence to his/her states and the states would change accordingly to the surroundings (Pan *et al*,2012). As in the evaluation towards single employee’s productivity, its own performance and the effect of the surrounding should both be taken into account.

Since employee’s productivity is important to company, there should be some approaches or standards which can be used to measure the staffs inside a company. However, some people claimed that software engineering is too complex to measure. Nortal (2018) pointed out in their article that classical measurement like working hours, line of code, bugs fixed, function design and defect rate all have some disadvantages. Working more hours but produce equal or less output, adding more meaningless lines of codes in order to ‘increase’ the quantity of output, in the real development situation, it would be easy for skilled and cunning employees to fake the productivity and affect the evaluation result. Fabulich (2016) argued in his article that if people want to measure the productivity for a software engineering project, then the output of project needs to be valued which is hard or impossible to implement. Instead, he pointed out that the satisfaction can be an optional benchmark to measure the productivity indirectly since the good productivity can enhance the satisfaction and the good satisfaction can improve productivity inversely in general. The satisfaction and the productivity would form a virtuous circle for an employee and the manager should try to take advantage from this circle. Using this kind of measurement, the productivity can be measured by the degree of the satisfaction which is more operable. The company where Fabulich is working for uses the question like ‘How likely are you to recommend working at this company?’ to ask the employees and collect the results to do the analysis. In addition, his company would also try to pay more attention to the complaints from employees in order to improve the productivity via the improvement in satisfaction. The attitude of the employee towards its own work should be paid more attention which is the emphasis of Fabulich’s work. Although Fabulich gives people a brand new view to do the measure, he also admits that the satisfaction measure only takes the largest factor of the whole evaluation since the measure of satisfaction is also indirect and subjective, some basic and classic factors like bugs counts are also taken into consideration in order to complete the whole performance measure. The approach mentioned above gives people some inspirations to the measurement of productivity, they are reasonable and practical to certain extent but the measurement is still not objective and accurate enough.

Some researchers tried to use technology and mathematics to measure the productivity in a more accurate and detailed way. The point is that, although software engineering process itself is hard to measure, people can divide the whole project and the components inside can be measured individually to reflect the whole project’s progress. Yano *et al.* (2015) pointed out using a wearable device to measure the activity of employee since they believe the equation ‘physical activity = happiness = productivity’ holds true, the physical activity can be used as an indicator of certain person’s working state since people ‘s active and mind state would affect physical indexes. Another research (Anchor, 2012) also claimed that people who are happy have 37% higher work productivity and 300% higher creativity then those unhappy people which supports Yano’s research. Fenton and Martin (1999) used an advanced and combined software matrix to produce decision-making system and track the software development process, making future predictions, evaluations and trade-offs during the process. Grambow, Oberhauser and Reichert (2013) also introduced an ontology-based multi-model holistic approach to measure the software engineering process, it unified three classic process reference models (CMMI, ISO 15504 and ISO 9001) but also has its own features to expand the evaluation. Dittrich, Gunes and Dascalu (2013) described a network graph for identifying the expert in certain project as well as to compare the performance in project and help manager do the comparison.

Among all the different kinds of measure approaches, what interests me most is the way provided by Connor, Finlay and Pears (2014). Since the development of certain project depends on the usage of various kinds of tools. Their method is to track the developer’s communication data during the development of project, making dynamic decision-making tree model and adjust it with the new data. In this way, the insights towards collaboration and development activities can be derived from the data and the future of project can also be predicted to help manager make decisions. To verify their method, they used the data from Jazz repository, provided by IBM, which is an open source repository that recorded the data of a large number of software project development processes and artefacts. Developers’ interaction and communication can be extracted from the repository. The project output would be reflected by the user comments on social network and the accuracy of model can be calculated by comparing the prediction and user feedbacks. Connor and his team used the changing communication data as a stream to feed the model and predict the project’s future outcomes as well as the productivity. In addition, the links between team communication and the project’s output can be derived using the model.

Unlike other researchers who use the matrix of source code to build the decision model, Connor’s research extends the model by adding the data mined from the developers’ communication stream and form a new matrix as a new variable. As I mentioned above, the communication between developers could make a difference to the productivity and the output, people’s mind state and concentration would be affected by the environment. The model would also discard the old part of the stream once the update of model has been completed in order to save the memory usage. In building the social network graph, data are derived from the communication networks and different roles have been assigned to the people related, including committers, creators, commenters and subscribers. Each role has its own unique influence to the project which can be derived from the name and as the existence of ‘subscribers’, other developers and customers can comment on the project to enrich the content of the network. Connor pointed out that once the social network graph has been built through the data extraction, some necessary variables can be calculated to form the decision tree, which includes:

1. In Social Network Centrality Metrics:
2. Group In-Degree Centrality (the number of edges towards certain node)
3. Group Out Degree Centrality (the number of edges outwards certain node)
4. Group InOut-Degree Centrality
5. Highest In-Degree Centrality
6. Highest Out-Degree Centrality
7. Node Group Betweenness Centrality(for every pair of vertices, there should be at least one shortest path between the vertices no matter it’s calculated by the minimized number of edges or the sum of weights, the betweenness centrality is the number of these shortest paths that pass through certain vertex(Wikipedia, 2019))
8. Edge Group Betweenness Centrality
9. Group Markov Centrality(same as random walk closeness centrality which describes the average speed by walking randomly through the network to reach a certain node(Wikipedia, 2019))
10. In Structural Hole Metrics:

1.Effective Size and Efficiency

(where structural hole can be understood as the gap between two peoples who have complementary information source (Wikipedia,2019))

1. In Basic Network Metrics:
2. Density
3. Sum of vertices

3.Sum of edges

1. In Additional Basic Count Metrics:
2. Number of work items the communication metrics were extracted from
3. Number of change sets associated with those work items

The tree would use the data in the Jazz repository to train and verify its prediction, success or failure are used to indicate certain project’s result according to the variables calculated before. Connor’s team uses Hoeffding tree as its model prototype as we need to deal with dynamic data while the basic decision-making tree only supports static data. The improvement is achieved by using the Hoeffding bound which can be used not only consider the accumulated instances but also make assessments to the information that would result from the future. The bound can be calculated using the formula:

where R represents the random variable of assessed criterion, n is the number of observations and delta means a confidence parameter.

The result of Connor’s method shows that the accuracy of prediction stabilized at around 64% after approximately 100 samples has been imported. The accuracy of prediction doesn’t seem so high but it shows that the model only uses limited calculated variable to build up the decision tree and improve the accuracy gradually. The data mining of this stream can indicate the productivity of employees through the social network graph and located the success or failure using part of the connections between nodes inside the graph.

However, since different roles would all contribute to every progress, it may need more time to identify each person’s individual contribute. The author also pointed out it would better if the number of samples could be larger and more software metrics could be combined to improve the accuracy of prediction. From another view, the collection of these communication data is also under debate.

Employee’s communication and social network can be considered as personal privacy and if these data are used to predict the future of project and calculate the productivity, the employees would feel uncomfortable to be supervised by tools. The laws for employee’s privacy are still under debate. Some companies would ask employees to set up account for work purpose and track the flow or detailed content of the communication (HRP, 2017). As a result, the employees’ communication, whether is work related or not would be supervised by the company without notification which would impinge on privacy rights to certain extent. This kind of supervision would affect the behavior or mental state and as mentioned before, the employee’s state could affect the productivity of the project. The collection and the use of employee’s communication information should be paid more attention and consideration to protect the privacy.

**Conclusion**

In this report, I have talked about several issues towards the measure on productivity. The ways introduced above include both directly and indirectly approaches. The software engineering itself is vague and complex but certain ways can be used to measure and compare it from different perspectives. The measure could give manager the guidance to the future work and compare the performance among employee. In addition, the privacy issue towards measure has been introduced shortly in the last part which points out the privacy issues of data collection. To conclude, the productivity is affected by various factors (like developer communication, developer’s state) and the relevant data can be collected appropriately and feed into advanced model to evaluate the effiency.

**References**

1. Anchor, S. (2012), ‘*Positive Intelligence*’ Harvard Business Review (Jan./Feb. 2012). Available at: <https://hbr.org/2012/01/positive-intelligence> (Accessed: 23 Oct, 2019)
2. Bughin, J. and Chui, M. (2013) ‘*Evolution of the networked enterprise: McKinsey Global Survey results*’. Available at: <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/evolution-of-the-networked-enterprise-mckinsey-global-survey-results> (Accessed: 23 Oct, 2019)
3. Connor, A.M., Finlay, J. & Pears, R. (2014) "*Mining developer communication data streams*", Proceedings of the Fourth International Conference on Computer Science and Information Technology (CCSIT 2014)
4. Dittrich, A., Gunes, M.H., and Dascalu, S. (2013) "*Network Analysis of Software Repositories: Identifying Subject Matter Experts.*" Complex Networks. Springer Berlin Heidelberg. pp. 187-198.
5. Fabulich, D. (2017) ‘*You can’t measure software engineering productivity, so measure job satisfaction instead*’. Available at: <https://redfin.engineering/measure-job-satisfaction-instead-of-software-engineering-productivity-418779ce3451> (Accessed: 24 Oct, 2019)
6. Fenton, N. E., and Martin, N. (1999) "*Software metrics: successes, failures and new directions.*" Journal of Systems and Software 47.2 pp. 149-157.
7. Grambow, G., Oberhauser, R. and Reichert, M (2013) ‘*Automated Software Engineering Process Assessment: Supporting Diverse Models using an Ontology.’* Int'l Journal on Advances in Software, 6 (1 & 2). pp. 213-224.
8. Human resource consultants (2017) “*Employee’s privacy laws in the workplace*”. Available at: <https://hrpgroup.ie/employees-right-to-privacy-in-the-workplace/> (Accessed: 26 Oct, 2019)
9. Nortal HQ (2018) ‘*The myth of developer productivity*’. Available at: <https://nortal.com/blog/the-myth-of-developer-productivity/> (Accessed : 24 Oct, 2019)
10. Pan, W., Dong, W., Cebrian, M., Kim, T., Fowler, J.H., Pentland, A.S. (2012), "*Modeling Dynamical Influence in Human Interaction: Using data to make better inferences about influence within social systems*" in IEEE Signal Processing Magazine, vol. 29, no. 2, pp. 77-86, March 2012.
11. Wikipedia (2019) ‘*Betweenness centrality’*. Available at: <https://en.wikipedia.org/wiki/Betweenness_centrality> (Accessed: 30 Oct, 2019)
12. Wikipedia (2019) ‘*Random walk closeness centrality*.’ Available at: <https://en.wikipedia.org/wiki/Random_walk_closeness_centrality> (Accessed: 30 Oct, 2019)
13. Wikipedia (2019) ‘*Structural holes*’ Available at: <https://en.wikipedia.org/wiki/Structural_holes> (Accessed: 30 Oct, 2019)
14. Yano, K. *et al.* (2015) ‘*Measuring Happiness Using Wearable Technology —Technology for Boosting Productivity in Knowledge Work and Service*’. Available at : <http://www.hitachi.com/rev/pdf/2015/r2015_08_116.pdf> (Accessed: 23 Oct, 2019)