

# An Implementation to Reduce Internal/External Interruptions in Agile Software Development Using Pomodoro Technique

Mintra Ruensuk

Department of Information Technology  
Faculty of Business and Technology  
Stamford International University  
Bangkok, Thailand  
Email: mintra.ruensuk@stamford.edu

**Abstract**—Nowadays, in order to deliver software product in time. A developer needs to focus on their work seriously. Although, there are many distractions occurring during work time, the product must be delivered by the deadline. Developers may need to study on how to improve their focus and productivity. This paper aims at the implementation of using the Pomodoro Technique for reducing internal and external interruptions in Agile software development with case studies of a developer who is a team member of an agile team. The implementation of the study indicates that Pomodoro Technique can be adapted to fit for the developer based on his/her tasks. Developer can simply construct his/her rules such as categorize internal and external interruptions, extend Pomodoro in some certain cases to maximize the use of the Pomodoro Technique. The result and discussion of this study shows the statistics of developer's Pomodoro. It illustrates that the developer's productivity dramatically increases because the number of distractions decrease. The result of this study also shows that the developer and his/her colleagues satisfy the result of using the Pomodoro Technique.

## I. INTRODUCTION

Time management method is used widely to improve people's focus on doing interested things. It helps people to finish a task in time by improving their own productivity. In Scrum, which is one of the methods in Agile software development [1], there are people who are working as a team. Team is responsible to analyze, prioritize, define, and estimate the work. Planning session is used to collaborate among team members to become with the plan of that given sprint.

During the day in the active sprint, each member is responsible to manage his/her own given tasks. The story point of the task illustrates its complexity. However, the task may not be finished in time because people may get interrupted by any interruptions e.g. request for help from team members. The problem occurs when the task can not be done in the given time block. Time management method is needed in this case to help individual people improving their own process.

### A. Pomodoro Technique

Pomodoro Technique [2] was developed by Francesco Cirillo, a CEO of XPLaps in Germany, in the late 1980s. It is the time management method which uses a fixed block of times to

improve productivity to archive defined tasks. This idea helps to enhance focus and concentration by reducing interruptions, including both internal and external distractions.

Moreover, Pomodoro Technique is also useful for refinement the estimation process. It shows clearly how much time you have spent to complete a task as well as time you have wasted for interruptions. There is a time block for you to clarify the cause of problems and find a way to improve the next task.

The idea is basically you have 30 minutes called "Pomodoro". For the first 25 minutes is the time that you concentrate with a given task [3]. Then you have 5 minutes to take a break. A Pomodoro cannot be interrupted or split up. Whenever your Pomodoro is interrupted, it is categorized as void. In your 25 minutes, you are not allowed to do a task other than a given task. Even thinking of things other than work will be considered as internal distractions. In your 5 minutes break, you can do anything you want, including talking about the relax topic, go to another room, or even go to the toilet. It does not matter which activities you would like to have during your break as long as it does not call for any significant mental effort such as watch movies or television series, write important emails, start talking about related works with colleagues.

For every four Pomodoros, you are allowed to have 15-30 minute break. It is an opportunity for you to refresh up your mind by doing things which are not complex. You may use this time block to review what you did for last four Pomodoros and how to make your next Pomodoro better.

Type of interruptions is listed as follows:

1) *Internal interruptions*: This situation happens while you are working and you interrupt yourself by thinking of things other than work, the need to go somewhere else, and so on. Because people may have a little ability to concentrate things continuously in a block of time, people will lose focus by wasting their time unconsciously. Especially working in front of a computer as a programmer, it is very easy to lose the focus by opening a web browser and go to the desired web page.



Fig. 1. Kitchen timer [2]

2) *External interruptions*: This type of interruptions happens when you may interrupt by others or something that you have no control on it. Especially in Agile software development, we do normally work as a group of people. You may be interrupted by your colleague to do something urgently. Then your Pomodoro will be broken. Another example is that the phone ring, the email, and so on. You may need to have a solution to manage this type of interruptions. For example, if you are often interrupted by your colleague, then simply tell them beforehand that you are currently in the Pomodoro and you will contact them back after you have done this Pomodoro.

## II. AN IMPLEMENTATION

Aiming to improve individual productivity of a developer in a software process, an implementation of using the Pomodoro Technique in a real software development environment is conducted. It also attempts to adapt the use of the Pomodoro Technique to fit with any given task and any practitioner.

### A. Notification tool

The notification tool may need to notify you when you have finished a Pomodoro. Originally, Pomodoro was invented to use with an analog tool called kitchen timer as illustrate in figure 1. On the other hand, the digital tool can be used. It is much more suitable for those who are working in front of computer. The tool allows user to adjust the block of times if it is not the default 25 minutes as same as break time. Figure 2 shows tools which can be used on PCs. Figure 3 indicates native application of Pomodoro which can be used on iOS platform.

### B. Record the Pomodoro

In order to get useful information to analyze in the future, you may need to keep recording the history of a Pomodoro. Developer experimented by using paper to record it. Note it down whenever internal/external interruptions occur by using his/her own code to distinguish interruptions. Interruptions are categorized as codes as follows:

*I* is internal interruption

*E* is external interruption

Moreover, interruption can be organized into sub category which is useful to distinguish more clearly. Developer's personally list of sub-interruptions are listed as follows:

- Internal interruption

1. *ISF* is the time when the need to access Facebook is occurred

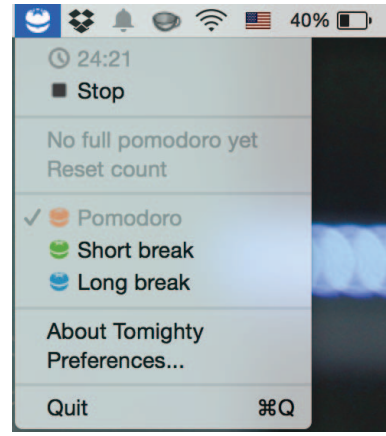


Fig. 2. Tomighty application on Mac OS X



Fig. 3. Pomodoro application on iOS

2. *IST* is the time when the need to access Twitter is occurred
3. *IO* is the time when thinking of work which is not related to current task
4. *IM* is the time when the need to leave a work space is occurred
5. *IP* is the time when missing somebody
6. *IX* is applicable for any other internal interruption

- External interruption

1. *EB* is the time when the boss needs help or asks
2. *ET<sub>x</sub>* is the time when a team member interrupts, where *x* is the name of the team member e.g. *ETL*
3. *EP* is the time when the phone is ringing
4. *EO* is applicable for any other external interruption

### C. Advanced Pomodoro

Obviously, the concept of the Pomodoro Technique helps people to improve their focus and productivity. However, after using it for a period of time. Developer may notice the limitation of it. For example, for some tasks require a block of time to focus continually which cannot be broken down into smaller task. Developer may already have a deep focus during his/her Pomodoro. When a block of time is finished. By concept, he/she needs to stop it right away. Sometimes, this is an interruption in the bigger point of view.

Developer is allowed to adapt the Pomodoro Technique based on his/her habit. From the experiment, in order to work with complex tasks require time and deep understanding.

TABLE I  
AN EXAMPLE OF A POMODORO RECORD

Date	C No	S Point	S Time	Pom	Min	Status
21/11/14	ET-122	1	10:30	IO ISF ISF IST	25	D
				IP EP IO		
			11:00	ISF ETL	13	V
			11:40	IO IST IX	25	D
24/11/14	ET-142	3	13:00	IO IO	60	AEV
			10:30	ISF IO	25	D
			11:00	IO	25	D

where D = Done, V = Void

Developer may extend his/her Pomodoro by ignoring the notification alarm. *AEV* is used to indicate as an extended Pomodoro which produces a void Pomodoro.

#### D. Result template

Table I indicates the example records of using Pomodoro Technique. In Scrum, the tasks are normally broken down into a card. A card has its story points which indicates a complexity or how long will take to finish the card (it depends on the culture of a company). A story points, in this case, is used to indicate the number of time in an hour to finish the card. The parameters are listed as follows: date, card number, story point, start time, Pomodoro record, duration, Pomodoro status.

### III. RESULT AND DISCUSSION

This study was experimented by a software developer who is responsible to analyze, implement, and test the given tasks. The developer is one of the members of an agile team. Every team members, including Product Owner were noticing that the developer would like to start this study. They were prepared to evaluate this study after a period of time.

The idea of this study was basically to study the consequence of using Pomodoro technique. It mainly focused on velocity of a task, developer's satisfaction and team's satisfaction. The study has been conducted continuously for four sprints, which was around two months. There were two projects involving during this study. Based on an assumption of using Pomodoro Technique, the productivity of practitioners should be increased. This study attempted to compare effectiveness of work before and after using Pomodoro Technique along with stakeholder's satisfaction. The developer (practitioner) was responsible to record time he/she spent to finish a card. Tempo Timesheets for the JIRA, a plugin integrates with the JIRA for tracking the time, is used to collect time information of a story card.

Figure 4 shows information of a card which has been done before applying the Pomodoro Technique. The card has 1 story point which was planned as 4 hours to complete this card. As can be seen from this example card, the developer has spent her 3 hours and 5 minutes to complete the task. Likewise, figure 5 also indicates the information of a card, whereas this card has been done during the use of Pomodoro Technique. Within the same number of story point, this card can be done with the less number of hour spending to complete a task comparable to the previous card.



Fig. 4. Example of time information before using Pomodo Technique

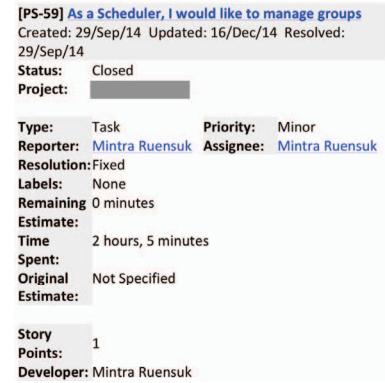


Fig. 5. Example of time information after using Pomodo Technique

Not only time information was collected, three keys were collected as well, which were number of internal/external interruptions, type of internal/external interruptions, and the status of a Pomodoro whether it was done, void or extended.

Figure 6 illustrates a comparison of planned story point versus worked time in hours. As can be seen from the chart, Sprint 0-0 and Sprint 0-1 were categorized as sprint which did not use the Pomodoro Technique yet. Sprint 1 to Sprint 4 indicated time during the experiment. It can be seen clearly that number of hours in the Sprint 1 to Sprint 4 gradually decreased where number of story point per each sprint were not dramatically changed.

Figure 7 indicates the comparison of a number of internal interruptions and number of external interruptions occurred in each sprint. During the first sprint of Pomodoro, number of

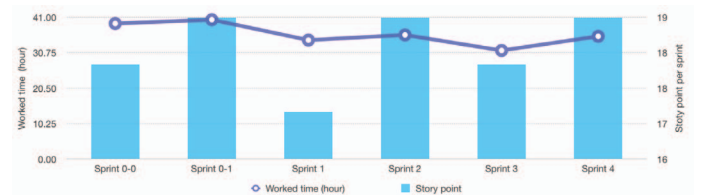


Fig. 6. The 2-Axis chart of story point and worked time in hour

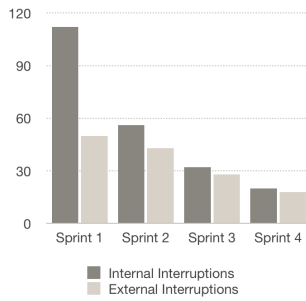


Fig. 7. Number of internal/external interruptions

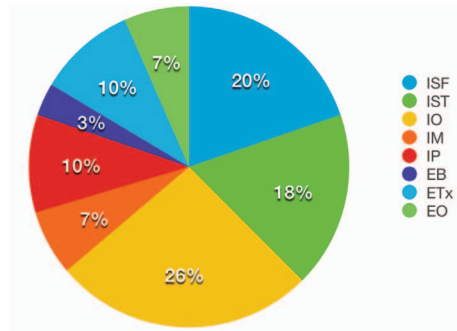


Fig. 8. Internal/external interruptions by sub-interruption

internal interruptions were the highest. After using Pomodoro Technique for a sprint, developer can adapt his/her own process to fit with the Pomodoro Technique clearly because number of internal interruptions have dropped dramatically since Sprint 2. Likewise, the number of external interruptions also decreased gradually since Sprint 1.

Figure 8 shows sub-interruption of both internal and external interruptions. Almost 40% of sub-interruption went to internal interruption which related to social network. Another 26% was also categorized as internal interruption. External interruptions took just around 20% of it.

Figure 9 displays the result of Pomodoro by its type where Done means the developer can finish his/her Pomodoro, Void means a Pomodoro was broken, and finally AEV mentions extended Pomodoro (which can be categorized as Void). There are 75% indicating the good Pomodoro(s) whereas a quarter of it showing unsuccessful Pomodoro. However, AEV is used to declare that developer introduced his/her own idea to match with his/her process.

The study also aims to receive feedback from developer and stakeholder. The team and developer were asked to give feedback about the use of Pomodoro by a survey after Sprint 4 had ended. The retrospective meeting of Sprint 4 was used to discuss the benefits and drawbacks of using Pomodoro among the team members. Developer satisfied with the work and the increase of productivity. The score of taking the survey was 4.76 out of 5. Although, the team members are blocked by the developer sometime when he/she was asked for help, the team members still satisfied the implementation of Pomodoro Technique. Only one improvement was to increase

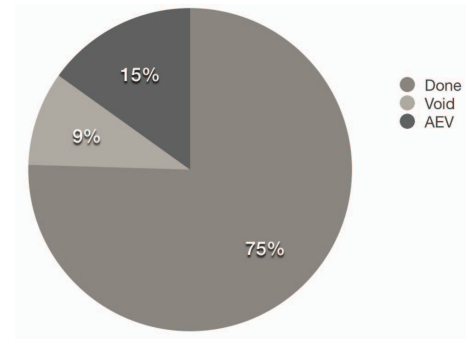


Fig. 9. The result of Pomodoro by its type

the communication with the team. The result of the survey indicated that the rate of stakeholder's satisfaction was good which averaged 4.48 over 5 (standard deviation = 0.29).

#### IV. CONCLUSION

The Pomodoro Technique is a time management tool that was originally aimed to optimize time we have spent on personal work and study. Recently it has been used widely especially in software development. This study experiments how Pomodoro Technique is used in the real software development environment. The study intends to focus on factors to increase developer's productivity as well as stakeholder's satisfaction. There are interesting metrics which are time, number of internal/external interruptions, type of internal/external interruptions, and result of a Pomodoro.

Based on experiment, the metrics reveal that number of internal interruptions significantly decrease after using Pomodoro Technique for a sprint. Although, number of time taken in each sprint slightly decrease, the developer's satisfaction dramatically expand. Stakeholder's satisfaction of the use of Pomodoro Technique is excellent. However, the communication among team members is also an issue.

In the future, using Pomodoro Technique for the whole team may be considered since it is interesting to study how Pomodoro Technique will be applied for every team members and how they can communicate if everyone has its own time bracket.

#### ACKNOWLEDGMENT

The authors would like to thank Dr. Paul Janeczek, CEO of Think Blue Data co, Ltd., and the team for the collaboration.

#### REFERENCES

- [1] 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," 2001. [Online]. Available: <http://www.agilemanifesto.org/>
- [2] F. Cirillo, "The pomodoro technique (the pomodoro)," *Agile Processes in Software Engineering and*, vol. 54, no. 2, 2006.
- [3] A. Nyström, "Agile solo-defining and evaluating an agile software development process for a single software developer," 2011.