Documentation of Software Development Processes in a Tech company.

Chaymaa IJERMKI

Concordia University

Background

Context

"One of the cornerstones to any quality program is documented processes" (Westfall, 2001, pp.1). Documentation of processes is a major step when conducting a project since it allows workers to access a written or digital database that encompasses the mechanisms followed in order to conduct a task. However, this database could be out-of-date or directed towards certain employees more than others, or the workers could not even be aware of its existence. CI Company is a tech company that is having a lack in the documentation of software development processes that its engineers and developers are experiencing with. For this matter, they tried a process of documentation that is created and maintained for the new code-base in order for new hires and interns to easily find documentation of certain features; however, this system is recent and did not use to exist in order to keep track of the features and work invested in old systems. In addition, the company hired a technical writing team to create technical documentation of the coding. However, this initiative ended up being more based on the business value that it would bring to the company. The priorities were directed towards a documentation that is geared for business partners, rather than an internal documentation for software engineers and developers. CI Company is left with the same issue even after its several attempts of resolving it.

Moreover, according to a subject-matter expert, most team members have little experience in the company. Thus, they do not have enough knowledge of the old systems, and this is a risk that can lead to a misuse of the company's coding mechanisms that could potentially lead to breaking the system. Furthermore, the SME added that there is no lack of

collaboration within the workers. The software engineers and developers are happily willing to share their knowledge and expertise; however, the experts among them are very over-worked, so they do not have enough time to share their expertise. In addition, the company tried to find alternatives for this issue by building expertise within teams in order for each one of them to be knowledgeable in a certain area. However, since the company is using the Agile method, they are faced with the incapacity to shuffle teams at the end of every sprint.

Research

In order to get enough information about the subject matter, I did some online research based on key words, such as 'software knowledge documentation', 'documentation of software processes', and 'software system documentation'. Furthermore, some papers were very useful to understand this issue more in depth, such as Linda Westfall's 2001 paper on *How to create useful software process documentation*, but also Hopinks and Jernow's 1990 paper on *Documenting the software development process*.

The Subject Matter Expert that I interviewed is a software engineer that has been working with the company for one year. He is relevant because despite his short experience, he was able to learn about the processes that most engineers need thanks to his social skills since he befriended old engineers of the company that gave him the information needed. The questions that I asked him were about the importance of this technical documentation, the difficulties that engineers go through for not having documents that explain these processes and the solutions that they turn to in order to conduct certain tasks.

Results

Restate the request

CI Company is a tech company that is specialized in advertisement. Their request is: I

need you to create a training program for software engineers and developers for them to be able to write, in a minimal amount of steps, a complete documentation of the software development processes that are confined within the company in order to operate the company's systems while saving precious resources.

Business need

The business need is to generate revenue by documenting the coding processes. New hires and interns will be able to find the data needed faster and complete their tasks rapidly and effectively. The company will then be able to meet the deadlines or even present the product to their clients before the deadline, which would allow the company to generate thousands of dollars in revenue by being first in the market and winning against the competitors. Thus, the company will gain new customers from the competitors and keep the current ones, which will generate more money for CI Company.

Performance problem

1) Current scenario

A Tech company has just hired a freshly-graduated computer science student. His valuable knowledge and small experience as an intern allowed him to enter the market of software engineering right after his graduation. As soon as he is introduced to his team members and desk, he is informed of a highly-anticipated project that the company has been conducting for the past six months that is expected to launch the next month, but the planning of the project has been badly-managed and they need to present the final product to a very important client. Therefore, the amount of work that is ahead of the team is too high for the tight schedule they were assigned. The new hire needs to start working on his task as soon as possible in order to progress. He is asked to conduct a stress test in order to determine how

many advertising requests a software can take per second. The new hire roams around the company looking for documentation on how to conduct this test, he asks other workers that are unaware of where he could find that information. Back to his desk, he grabs a notepad and goes to the most experienced engineer on his team who then turns him down because he needs to finish his share of the work before the next meeting. The new hire goes to a less experienced engineer who takes some time to explain the basics of the stress test and advises him to try to understand the rest on his own, but not take too much time, since the deadline is near. After doing some research, the new hire is able to conduct the stress test but lacks time to document the steps he went through in order to run it, but it does not matter, he remembers the process.

2) Ideal scenario

A Tech company has just hired a freshly-graduated computer science student. His valuable knowledge and small experience as an intern allowed him to enter the market of software engineering right after his graduation. As soon as he is introduced to his team members and desk, he is informed of a highly-anticipated project that the company has been conducting for the past six months that is expected to launch the next month, but the planning of the project has been badly-managed and they need to present the final product to a very important client. Therefore, the amount of work that is ahead of them is too high for the tight schedule they were assigned. The new hire needs to start working on his task as soon as possible in order to progress. He is asked to conduct a stress test in order to determine how many advertising requests a software can take per second. The new hire asks one of his team members where he could find the documentation needed to run the test. He is introduced to a database that informs and answers all of the questions needed to perform the stress testing.

Due to his high enthusiasm and desire to give a good impression, the new hire finishes his share of work right on time and goes to the team leader in order to ask him for more tasks to do.

Tasks in ideal and current performance.

1) Tasks in current performance

- Refer to more experienced engineer or developer when information about the system is needed.
- Use the chatbot of the company to inquire about the steps to conduct certain tasks.
- Search online for software development processes.
- Send an email to a senior software engineer or developer to ask about the systems of the company.

2) Tasks in ideal performance

End Result: Document coding processes.

Main task 1: Create an informative database

- *Supporting tasks:*

- Gather data about old systems.
- Identify data types
- Choose a form for the database. (Files, website, reports…)
- Identify the hardware and software needed.
- Create a template for the database. (Tables, queries…)
- Add information to the database.
- Implement the database.

Main task 2: Create a process to document newly added code and its processes.

- *Supporting tasks:*

- Identify the hardware and software needed.
- Gather the data needed.
- Choose a form for the process.
- Add the information to the process.
- Implement the process.

Main task 3: Manage the database and newly added process.

- *Supporting*
- Test of the database and the process.
- Develop the database and the process constantly (progress reports, debugging, adding new features…)
- Conduct assessments and evaluations.
- Allow the database and process to be uploaded.

Learners

- High maintenance learners: Knowledgeable engineers and developers with expertise. Their age range would be high, but there are young engineers or developers with a lot of expertise within the company. They know a lot about the systems and the features used in it. However, despite their knowledge, they remain the learners that may not be suited for documentation since they are not aware of the challenges that the new hires go through when they just enter the company, nor their way of thinking or the knowledge base they have as new hires. Therefore, they will have a hard time documenting the processes in a way that is suitable for less-knowledgeable engineers or developers.

- Medium maintenance learners: Software engineers and developers that are established in the company for a long period of time and have experience and knowledge about the systems, but not as much as the senior engineers and developers. They also have a hard time documenting coding processes and will not able to remember the mechanisms that new hires and interns would need to get more insight into the company's systems.
- Low maintenance learners: Software engineers and developers that have slightly more knowledge than interns and new hires. They are low maintenance because they are the most recent employees and still have in mind the difficulties that they went through as new hires. Thus, they will be good at documenting the coding processes in a way that is the most suitable for the interns and new employees. This sample of engineers will be able to learn how to have access to old processes and how to currently conduct technical documentation in a more efficient manner.

1) In the learning environment.

- Approaching deadlines for other projects while the engineers need to document the processes of the code they currently work on. Therefore, the learners are tied to a busy schedule.
- Software engineers not open to face-to-face interaction.
- Software engineers prefer problem-based activities.
- Training program must include the three large groups of software engineers and developers.
- The budget is not specified.

2) On the project.

- Needs to be within the office to avoid transportation.
- Needs to be online or in a digital format since it is the most preferred medium for software

engineers and developers.

- Training program has to exceed the previous solutions to the issue of technical documentation that the company used.
- Use of simplified English.

Requirements

Objectives

1) Business objective.

Given the implementation of a documentation process, CI Company will be able to keep their current customers and gain an additional 15% new customers yearly from the competitors.

2) Learning objectives.

- Given a need to use an old software system, create an accessible and user-friendly database that contains information about the old systems.

- Given the hiring of inexperienced and uninformed interns, create a process to document newly added code and its processes that is accessible in a maximum of ten steps.

- Given the use of the database and newly added process by several engineers and developers, manage these resources on a weekly basis.

Evaluation

- Level 2: Self-assessment done by software engineers and developers in order to evaluate their knowledge of the documentation process after the training.

Here is a sample of the instrument:

	Yes	**No**
I find it easy to gather data about old systems.		
Adding information to the database is unchallenging.		
I know all the steps needed in order to conduct technical documentation.		
The hardware and software needed for documentation is easily accessible.		
I can update the database in a minimal amount of steps.		

I am able to write clear coding processes.		
When I document, I do take into consideration interns, new hires and less-experienced workers.		
I use multimedia (screenshots, videos, etc.)		
I use the template provided for consistency.		

- Level 3: Self-assessment. The learner will be able to individually assess whether or not the learning objective has been achieved and whether he was able to use what he learned in his workplace.

The project involves documenting coding processes and looking for answers concerning old software systems, as well as updating out-of-date information in the databases. The target audience: a small team that contains the three levels of learners. I chose self-assessment in order to test the learners and their ability to apply their acquaintance in a larger-scale project and allow them to evaluate those new skills and judge if their needs are met.

This is the instrument used for the self-assessment done by the learner:

	Yes	No
Were you able to find the processes for old software systems in less than ten steps?		
Were you able to document the software processes that you used while coding in less than ten steps?		
You had difficulties following the steps needed for technical documentation.		
You can update the database without technical issues.		
You did not have to contact another colleague in order to write technical documentation.		
You had enough time in your day to do project-related tasks and technical documentation.		

References

Westfall, L. (2001). How to create useful software process documentation. *The Westfall Team.* 1-13

Hopinks, J. S., & Jernow, J.M. (1990). Documenting the software development process. *Proceedings of the 8th Annual International Conference on Systems Documentation – SIGDOC 90, 14*(4), 125-133. doi: 10.1145/97426.98000