Improvement of the documentation process in a small software company

INF5181 Process Improvement and Agile Methods in Systems Development

Marion DISTLER - marioncd@student.matnat.uio.no

9th semester

15/11/2012
Table of Contents

1. Introduction 3
   1.1. Context description 3
   1.2. Method 3
   1.3. Issues 3
   1.4. Goals 4
2. The Baseline Process 4
   2.1. Elements of the baseline process 4
   2.2. Descriptive model of the baseline process 6
3. Target Process 8
4. Implementation of the target process 10
5. Measurement and control 11
   5.1. Measurement plan 11
   5.2. Action plan 14
6. Discussion 14
   6.1. Underlying rationale of proposed changes 14
   6.2. Risks of proposed changes 15
References 16

Table of Figures

Figure 1. Legend of the graphical method 6
Figure 2. Baseline Process Model Diagram 7
Figure 3. Target Process Model Diagram 9
1. Introduction

This System Process Improvement plan is aiming to improve the documentation process in a small software development company. This company will be referred along this report as X Inc.

1.1. Context description
X Inc. is a small French software development company that specializes in highly technical and strong added value software programs. These software programs are mainly based on intensive calculation and optimization problems. The company is well implanted in its sector and has a faithful customer base all around the world. It has currently 6 software programs with several thousand licenses in use in more than 30 countries.

Aside from the business support employees, X Inc. has 5 employees involved in the development process including 3 developers. Their development process is defined but not formally, it is also not based on a well-known method. The company gathered throughout the years a main library containing core functions (mathematical optimization, 3D effect, …) that can be reused in other projects. One person is in charged of this library: it is the senior developer. Usually, every developer is in charged of the development of one software program and of the maintenance of two other programs.

1.2. Method
The method used to do this project was based on the PROFES method [1]. However not all the steps have been performed and some of the steps performed were not done in as many details as prescribed by the method. The steps performed were roughly:

- Step 1. Verify commitment
- Step 2. Identify product quality needs
- Step 4. Determine current process capability
- Step 5. Set product improvement goals
- Step 6. Determine necessary process changes
- Step 7. Describe process changes
- Step 8. Set metrics for the processes and product improvements
- Step 9. Prepare improvement implementation

For the measurement of the process goals, the Goal Question Metric [2] was used.

1.3. Issues
The SPI should address two main issues:

- The need for a more formal technical documentation of the software programs and their main library
- The need for quicker release of end-user documentation of the programs
These problems arises quite recently into the firm, indeed their way of documenting was still meeting the needs of the company until a few years ago. But due to their success, the intensification of the customer demands on new releases and new programs, the effort available for the user manual was reduced drastically and some user manuals were never finished even 2 years after the releases of the program.

Moreover due to the limited size of their developer staff, most “documentations” of the programs or libraries were communicated orally between the developers. But now with the possibility of growing, the problem of formation of new employee, adaptation of the development process for developing teams, and the loss of important information in case of turnover in the team is a real threat to the company.

1.4. Goals
This SPI plan is mainly focusing on formally defining and systematizing the documentation process. However the goals of this plan are beyond the sole improvement of the documentation. The aim of the proposed changes is to better document the products in the development phase without adding too much effort for the developer in order to gain efficiency in other tasks as maintenance, reusing of code or user documentation writing, and in order to prepare for the arrival of new employees.

Therefore the goals are:
- Accessibility for all developer to a written technical documentation for the main library and every new program to be made from now on.
- Reducing the effort of maintenance of software by 10%
- Reducing the effort necessary to write the user manual by 10%
- No increase in the effort during the implementation process

2. The Baseline Process

This section will depict the current development process of the company that integrates the documentation process. This information has been gathered during an interview with the marketing & functional manager of the company. No real process has never been implemented formally but things settled during the years into the following process.

2.1. Elements of the baseline process
The system development model of the X Inc. contains several elements that will be listed below.

Activities:
- Defining next task to performs: determine which task is to be performed next and if it is related to a functionality that will be integrated into the main library or not.
• **Software implementation**: the developer in charge of the program code the tasks related to the software by creating new functions and using the library functions. If needed the developer can ask for help the senior developer on the main library
• **Function implementation**: the senior developer in charge of the main library codes the new function
• **Review**: the output of the development is reviewed during a meeting and validated. The functionality of the code is tested but the code it-self and the comment are not reviewed
• **Integration in the main library**: the function is put into the main library
• **Integration into the final software**: all the different tasks are integrated into the final software
• **User manual writing**: based on the final software and the requirement specification determine in the beginning of the project, the user manual is written

**Artifacts:**

• **Requirement specifications report**: written by the functional manager in the beginning of the project, it compiles all the functionalities of the software determined at that time. It is not very extensive
• **Architecture design report**: written by the developer this report is the draft of the architecture of the software
• **Next task to perform related to a new functionality for the main library**: function determined by the technical manager and the developer as needed to be integrated into the main library
• **Next task to perform related to a functionality specific to the current software**: task specific to the current software and to be developed by the developer in charge of the program
• **Piece of software related to the tasks**: code of the tasks
• **Function related to the tasks**: code of the function to be incorporated into the main library
• **Validated piece of software related to the tasks**: code of the tasks was validated by the technical manager and is ready to be integrated in the final software at the end
• **Validated function**: code of the function validated by the technical manager and is to be integrated into the main library. This function will then be used by the developer in another task for the current software
• **Updated main library**: main library containing the new coded function
• **Final software**: final version of the software containing all the tasks integrated into one program
• **User manual**: end-user manual containing documentation of the main functionalities of the software and troubleshooting

**Roles:**

• **Marketing manager**: in charge of the user documentation
• **Functional manager**: in charge of the main functionalities of the software. He wrote the requirement specification
- **Developer**: in charge of the programming of the software.
- **Technical manager**: in charge of the good progression and quality of the project
- **Senior developer**: in charge of maintenance the main library of the company and the programming of new functions to be integrated into it

There are only 5 employees implicated in the development process, however some of the employees can have more than one role. For example the person being the marketing manager is also the functional manager and on certain project the senior developer and developer can be the same individuals.

**Tools/techniques/methods:**
- **Meeting**
- **In-code commenting documenting**, the developer put comments inside the code in order to be able to do maintenance on it afterwards

### 2.2. Descriptive model of the baseline process

In this section, the baseline process will be modeled using a graphical method. The legend of this graphical method can be found on Figure 1.

![Figure 1. Legend of the graphical method](image)

The SPI focuses on the documentation process, however documentation is spread across the entire software development process. In the first part of the system development process, some of the technical documentation is done by different actors. These documentations are: a marketing report, a requirement specification report and an architecture design report. It is also in this part of the process that the effort estimation is made and the Go/noGo decision is taken. This part of the process, being not implicated in the SPI, will not be developed here.

Figure 2 represents the second part of the development process that focuses more on the implementation. Two flows are done in parallel, once the next tasks to perform are defined they are classified into two categories: the ones that are related specifically to the software and the one corresponding to functionalities that will be integrated into the company’s main library. The company main library contains all the technical functions used in their software for the past 20 years and are used in most of their software. One person handles it: the
senior developer. As shown in the figure, one developer implements one program from top to bottom. This person is also in charge of its maintenance afterwards.

Figure 2. Baseline Process Model Diagram
3. Target Process

For a small company, with a limited number of developers at disposal and a great pressure to release software as quick as possible, proper documentation can be disregarded in profit of oral communication. However as the company grows, this communication can be a waste of resource, knowledge and effort. The purpose of this software improvement plan is to improve the documentation processes in the development process in order to achieve the goals displayed in Section 1.4. This section will present the changes that need to be done in order to achieve these goals.

The main changes of the process are:

- **Planning and tasks defining meeting.** A new meeting involving every stakeholder (functional manager, technical manager, senior developer and developer) in the development process is made at the beginning of the software implementation. This meeting will output a Work Breakdown Schedule of what the project consists of and will determine all the new functions that need to be implemented and also integrated into the main library.
- Setting up of a documentation software program of type Doxygen [3] or Ndoc [4] that creates code documentation from comments within the code. It will be used to compile the technical documentation of the software and of the functions contained in the main library.
- Implementation of a in-code documenting template and function naming rules allowing the compilation of formal technical documentation by the above mentioned program.
- Updating of the requirement specification after every completed tasks if necessary.
- Implementation of a user manual template

The changes made in the process are visible in green on Figure 3.

The baseline process already separates the implementation into tasks, however it is done with the flow and without documenting the choices or schedule made. In order to have more perspective and a better understanding of the project, we compile this decision of tasks defining into a planning and tasks defining meeting at the beginning of the implementation. This meeting should involve everyone concerned by the project because all their competences in the different parts of the project are needed in order to produce the output document: the work breakdown schedule.

Then in order to better the technical written documentation of the programs and library without increasing too much the effort of the developers, in-line commenting is kept but with the use of a unified template of commentating in the code and function naming. Moreover a documentation software program is introduced. It must be runned after every task in order to compile and keep an up-to-date technical documentation. The technical manager should review the documentation along with the functionality of the code during the review meeting.
Marion DISTLER

INF5210 - Improvement of the documentation process in a small software company

Figure 3. Target Process Model Diagram
In the baseline process the requirements specification is done once when defining the project but it is not updating along the way if changes occur. In the new process, during the review meeting, the technical manager needs to update the requirement specifications document if any change has been made to the functionalities of the software.

The user manual documentation activity is also changed. When interviewing the marketing manager, we identified that the main causes of problem during this activity are the lack of up-to-date documentation of the software and the fact that he has to do everything from scratch for every program. Therefore in order to facilitate the writing, the activity will performed using a defined template and will have as input up-to-date technical documentation and requirement specifications report.

4. Implementation of the target process

This section will display how the changes of the process should be introduced into the company.

1. Meeting announcing the changes in the process

**When:** This task should be performed at the beginning of the implementation of the target process. Indeed in such a small structure starting changes without informing the interested employees will be quite certainly noticed and will arise suspicion or even distrust in the future process from the employees. This should take a couple of hours of time, therefore 1 day effort.

**Who:** Everyone involved in the old and new process, therefore both 5 employees involved in the development process

**How:** The functional/marketing manager, which is director of the firm, should call all employees in a meeting along side the technical manager and explains the schedule of changes along with the changes that will be made.

2. Choosing the documentation software

**When:** This step should be performed after the starting meeting. It should be performed in no more than 1 week.

**Who:** The functional manager and the senior developer

**How:** The senior developer should research open-source and proprietary documentation software programs that fit the needs of the developers. Then it should show its selection to the functional manager that will decide which software to purchase and implement.

3. Determining and formalization of the in-code commenting documentation template

**When:** This task can be performed during or after choosing the documentation software. It should take 2 days time.
Who: The Senior Developer and the Technical manager  
How: The senior developer formalizes a template for the in-code commenting documentation. This should also include a template for naming functions. Once done, he should meet with the technical manager in order to validate the templates.

4. Determining and formalization of the User Manual Template  
When: This step can be performed in parallel with the two previous ones. It should take 2 days efforts  
Who: The marketing manager  
How: The marketing manager either reviews old documentation and extracts a template from them or finds an appropriate user manual template in software development methods. For example in many CMMI-SW process and documentation books, as the one from Susan K Land And John W. Walz [5], give templates for such deployable documents.

5. Prepare formation for the documentation software  
When: Once the documentation software has been chosen. This should take 1 day.  
Who: The senior developer  
How: The senior developer should install the documentation software and prepare a quick in-house formation for the other developer

6. In-house formation on the documentation software  
When: Once steps 5 has been performed. The effort estimation for this step is of 6 hours.  
Who: The senior developer and the developers  
How: The senior developer should form the two other developers on the chosen documentation software and introduce to them the new in-code commenting documentation template and function naming rules.

5. Measurement and control

This section will describe how to measure the performance of the SPI and how to decide if the SPI initiative was successful.

5.1. Measurement plan  
In this part, the Goal Question Metric method, as explained in [2], was used in order to describe the measurement plan.

4 goals have been defined in section 1.4. Based on these, there are also 4 goals for measurements and for every one of these goals, questions and metrics have been defined. For every measure introduced, the name N, the entity E, the unit U and the Range R will be stated.
**Goal 1:** Ensure the accessibility of one written technical document for every software program developed from now on and for the main library from the point of view of the developers

**Question 1.1:** How many software programs were developed since the SPI implementation and how many main libraries are available?

**Metric 1.1:** \( N: \) Number of entities = # of software programs + # of main libraries
\( E: \) Software programs and libraries \( U: \) dimensionless \( R: \ [1,2,3,\ldots, \infty) \)

**Question 1.2:** How many written technical documentation documents are available?

**Metric 1.2:** \( N: \) Number of available written technical documentation
\( E: \) Technical documentation \( U: \) dimensionless \( R: \ [1,2,3,\ldots, \infty) \)

**Question 1.3:** Have the developer access to written documentation for every programs or libraries in the scope of the SPI?

**Metric 1.3:** \( N: \) Availability ratio = \( \frac{Number \ of \ available \ written \ technical \ documentation}{Number \ of \ entities} \)
\( E: \) Documentation \( U: \) dimensionless \( R: \ [0,1] \)

**Goal 2:** Decrease the amount of effort used in the maintenance process from the point of view of the technical manager

**Question 2.1:** What used to be the average maintenance effort?

**Metric 2.1:** \( N: \) Old average maintenance effort = \( \frac{\sum_{\text{maintenance}}(# \ of \ developer \times # \ of \ days)}{\# \ of \ maintenance} \)
\( E: \) Maintenance process \( U: \) man.day \( R: \ [0, \infty) \)

**Question 2.2:** What is the average maintenance effort after SPI plan implementation?

**Metric 2.2:** \( N: \) New average maintenance effort = \( \frac{\sum_{\text{maintenance}}(# \ of \ developer \times # \ of \ days)}{\# \ of \ maintenance} \)
\( E: \) Maintenance process \( U: \) man.day \( R: \ [0, \infty) \)

**Question 2.3:** Is the average effort for the maintenance process decreasing?

**Metric 2.3:** \( N: \) Decreasing percentage of the maintenance effort
\( % \ Decreasing = \frac{Old \ average \ maintenance \ effort - New \ average \ maintenance \ effort}{Old \ average \ maintenance \ effort} \times 100 \)
\( S: \) Maintenance process \( U: \) dimensionless \( R: \ (-\infty, +\infty) \)
**Goal 3:** Decrease the amount of effort used in the user manual documentation writing from the point of view of the marketing manager

**Question 3.1:** What used to be the average user manual writing effort?

**Metric 3.1:**
- **N:** Old average manual writing effort \(= \frac{\sum_{\text{manual}}(\# \text{ of writer} \times \# \text{ of days})}{\# \text{ of manual}} \)
- **E:** User manual writing
- **U:** man.day
- **R:** \([0, \infty)\)

**Question 3.2:** What is the average user manual writing effort after SPI plan implementation?

**Metric 3.2:**
- **N:** New average manual writing effort \(= \frac{\sum_{\text{manual}}(\# \text{ of writer} \times \# \text{ of days})}{\# \text{ of manual}} \)
- **E:** User manual writing
- **U:** man.day
- **R:** \([0, \infty)\)

**Question 3.3:** Is the average effort for the average user manual writing decreasing?

**Metric 3.3:**
- **N:** Decreasing percentage of the user manual writing effort
- **S:** User manual writing
- **U:** dimensionless
- **R:** \((-\infty, +\infty)\)

**Goal 4:** Analysis of the global effort used in the development process from the point of view of the developers

**Question 4.1:** What used to be the average development effort?

**Metric 4.1:**
- **N:** Old average development effort \(= \frac{\sum_{\text{programs}}(\# \text{ of developer} \times \# \text{ of days})}{\# \text{ of programs}} \)
- **E:** Development process
- **U:** man.day
- **R:** \([0, \infty)\)

**Question 4.2:** What is the average development effort after SPI plan implementation?

**Metric 4.2:**
- **N:** New average development effort \(= \frac{\sum_{\text{programs}}(\# \text{ of developer} \times \# \text{ of days})}{\# \text{ of programs}} \)
- **E:** Development process
- **U:** man.day
- **R:** \([0, \infty)\)

**Question 4.3:** Is the average effort for the development process not increasing?

**Metric 4.3:**
- **N:** Evolution of the development effort
- **S:** Development process
- **U:** dimensionless
- **R:** \((-\infty, +\infty)\)
5.2. Action plan  
In order to decide if the SPI initiative was successful are not, one needs to look at the effort spent on different tasks on future project started with the new process. In order to be considered successful the following condition needs to be met:

- **Condition 1:** Metric 1.3, Availability ratio = 1
- **Condition 2:** Metric 2.3, Decreasing percentage of the maintenance effort ≥ 10%
- **Condition 3:** Metric 3.3, Decreasing percentage of the user manual writing effort ≥ 10%
- **Condition 4:** Metric 4.3, Evolution of the development effort ≤ 0%

If condition 1 and 2 fails, it means that the technical documentation process has been wrongly implemented or is not followed correctly by the developers. The technical manager needs therefore to better ensure, during the review meeting, that the technical documentation is up-to-date and according the template.

If condition 3 fails, it means that the user manual template is not relevant enough for the company or that the requirement specifications documentation is not done well enough. So the marketing manager needs to determine which is the cause of the failure. If it is the first one, he should find or make a better-suited template. Otherwise he needs to remind the technical manager to update the requirement specifications throughout the development process in order to have this documentation up-to-date with the software functionalities at the end of the development process.

If condition 4 fails, it means that the process is too heavy for the size of the team and that the improvements in effort in some areas are not good enough to compensate for the extra effort asked by the new process. In this case the managers should interview the developers and decide which part of the process should be lightened.

6. Discussion

In this part will be discussed the underlying rationale of this system process improvement plan. Afterwards the main risks in implementing this SPI plan will be outlined.

6.1. Underlying rationale of proposed changes  
The company is in a delicate position. On one hand the increase in demands of their software programs and the acceleration of the product life cycle is good on a financial point-of-view. But on the other hand, it stretches the organization to a breaking point. Indeed the amount of work per person with the ongoing organization is critical, but they don’t have enough work of one type of tasks to employ a new full-time worker and have trouble finding part-time worker with the needed competences. Moreover, the company will definitely grow in the next few years and it needs to prepare for it. Therefore, it is clear that X Inc. needs to formalize, rationalize and
improve efficiency of its processes in order to face the challenges of today (critical amount of effort, no documentation…) and prepare the future (arriving of new employees).

It is very hard to implement well-known methods into small companies since they are usually ill adapted or too heavy. The aim of the proposed changes is to better document the products in the development phase without adding too much effort for the developer in order to gain efficiency in other tasks as maintenance, reusing of code or user documentation writing. The choice of using template documents for the in-line commenting along with documentation software and the user-manual writing arises from this consideration. The new formalized documentation will allow the developer to be more independent of one another when doing maintenance or reusing part of the code they did not write themselves and will therefore reduce the effort used by the other developer to explain his previous work. These documents will also be important tools when new developers will be hired and will have to be formed.

As the demands for quicker new products and new releases will grow in their customer base, as it is now, the company will have to drop their system of “one program-one developer” and introduced developing teams. The proposed target process with its work breakdown schedule will be also more adapted to that than the baseline one.

Moreover an important concern of the company needs to be the knowledge management. As of now only one developer takes care of the main library and, since no written documentation is available, he holds almost all the knowledge of it. However this library is a very valuable component of the company as it holds part of their key competences and concurrence advantages. If ever this senior developer were to leave the company, this asset will partly or entirely disappear with him. Documenting the library is therefore also a way of securing one of the company’s assets.

Concerning the user manual writing, a better and more throughout process could be done in order to increase greatly the quality of this document. However, with the current organization, this will take too many resources away from other core processes of the business. The proposed changes have been selected so that the time required making the user manual decreases not to improve its quality. Indeed having a user manual released not to long after the software release is expected and the minimum in order to keep the customer. However great quality user manual is not the aim of the company and not one of the asset they want to develop, especially since their customer based is very specific and usually quite proficient with basic informatics knowledge.

6.2. Risks of proposed changes
Changing processes and therefore implementing a System Process Improvement plan always caries risks. Here are the main one identified for this SPI plan:

1. *Non-use of the newly done technical written documentation.* Once of the incentive of this plan is that the excess of effort used to do documentation during the implementation will be compensated by the amount of effort not lost during reusing and maintenance
of code. This is due to the fact that instead of asking for information about the functions orally with another developers, only one will be searching the documentation. However if the employees stick with their habits and keep spending time explaining to each other functions, it will just double the effort. This risks could be reduced by closely managing the team and make sure that the senior developer, the one being the most solicited in this kind of interaction, can properly refer the other developers to the new written documentation.

2. The employees do not agree with the new process. If the employee do not agree with the changes made or do not understand the need for these changes, there is a risk that they will not be motivated to properly performed the implementation steps and the new activities of the target process. This will result on a lot of money and effort spent on a SPI plan for nothing, since the implementation and use will be faulty and there the goals will not be reached. This risks can be limited if during step 1 of the implementation, the starting meeting, the management takes close attention on properly informing the employees on the reasons of the changes and what they have to gain with the new process.

References


