

Expected Influence of Ethics on Product Development Process

Stig Larsson

*The Department of Computer Science and Electronics,
Mälardalen University, Västerås, Sweden
stig.larsson@mdh.se*

Abstract: Product development efficiency and effectiveness is depending on a process being well executed. The actions of individuals included in the processes are influenced by the ethical and moral orientations that have been selected by each individual, whether this selection is conscious or not. This paper describes different ethical choices and the expected effects they may have on the development process exemplified by the product integration process for software products. The different frameworks analyzed are utilitarianism, rights ethics, duty ethics, virtue ethics and ethical egoism. The expected effects on the goals for product integration may be debated. This is a result in it self as it triggers discussions about ethical considerations and increase the awareness of the influence of

moral decisions. Our conclusion is that the adherence to specific moral frameworks simplifies the alignment of actions to the practices described in product development models and standards and through this supports a more successful execution of product development projects. This conclusion is also confirmed through a comparison between the different directions and several codes of ethics for engineers issued by organizations such as IEEE as these combine features from several of the discussed ethical directions.

Keywords: Product development, ethical directions

1 Introduction

The application of different ethical approaches in product development organizations is likely to influence the effectiveness and efficiency of product development (Martin & Schinzinger 2005). This is based on the assumption that actions performed by individuals involved in product development depend on the moral values that generally govern all areas of life. Ethical considerations can be investigated from different viewpoints; organizational, management, group and individual. The analysis here is concentrated on the choices made by the individual developer.

Throughout history different ethical theories have been formulated and expressed. To analyze these, a categorization is needed and in this paper we follow the classification made in (Martin & Schinzinger 2005). Five different moral frameworks have been selected and for each of those one or two different versions are described and analyzed from a product development perspective. The five frameworks are utilitarianism, rights ethics, duty ethics, virtue ethics and ethical egoism.

Numerous standards and reference models are available defining the processes needed to develop a product (Crissis et al 2005), (ISO/IEC 2002), (ANSI/EIA 1999), (ISO/IEC 1995). We have selected to investigate the Product Integration Process and concentrate on the case where the product is primarily based on software. This selection has been made as it highlights communication between different engineering disciplines and it relies on trust between co-workers.

The rest of the paper is organized as follows. Section two introduces the Product Integration Process which is used as an example of a part of the product development process. In section three, the different

ethical directions are applied to the actions by individual software developers in the product integration process, and the consequences are discussed. Section four contains a comparison between the different moral orientations with the IEEE Code of Conduct (IEEE 2005). In section five, the organizational influence on behaviour is discussed and section six contains a conclusion as well as proposed future work in this area.

2 Example Process: Product Integration

The endeavor of developing a product can be described as the execution of a set of processes. Different standards and reference models describe the requirements on these processes (Crissis et al 2005), (ISO/IEC 2002), (ANSI/EIA 1999), (ISO/IEC 1995). Several of these are collections of experiences that form what is considered to be best practices for the difference processes. In this paper, the example process is the Product Integration Process. Product integration involves several different groups of engineers and efficient communication as well as a high degree of trust is crucial for successful execution.

The Product Integration Process for software products represents the activities to combine software components to a product and to ensure that this product has the expected functions and qualities. The goal is to deliver a product that fulfills the expectations of the customer. It is expected that projects that have a Product Integration Process that follows the practices described in different reference models will have a higher probability to deliver on time with expected quality.

In this paper, we have selected a reference model from the Software Engineering Institute, the Capability Maturity Model Integration (CMMI) (Crissis et al 2005). CMMI version 1.1 defines 25 process areas, and for each process area there are a number of practices that, if performed, represents an indication of maturity. It is expected that this also increases the performance of the development organization.

Three goals are defined for the Product Integration process area: (i) prepare for product integration, (ii) ensure interface compatibility and (iii) assemble product components and deliver the product.

The three specific practices for the first goal are connected to each other. The basis is the integration sequence strategy and the integration sequence that build on this strategy. Besides the product components, the test components and equipment need to be included in the integration as the project progresses. This leads to the second practice which is to establish the environment for the integration. The build-up of this environment needs to be included in the integration planning based on the decided integration sequence. The different engineering disciplines such as software development, integration and test need to have a close cooperation to ensure that the plans are realistic and that all needed equipment is included in the planning. This cooperation is also needed for the third practice which is to establish the procedures and criteria for product integration.

The second goal describes the need to ensure that the different parts fit together. This can be achieved through two practices. The first is the review that is needed to make certain that the descriptions of the interfaces are complete, while the second is the need to manage the interfaces throughout the project life-cycle.

The actual combination of the different parts of a product is described in the third goal and is supported by four practices. The first is a preparation that includes checking that the delivered components adhere to the criteria for integration that has been established. The second is the actual assembly activities. These activities should follow the selected strategy and integration sequence. After the compilation, the product should be evaluated with specific care in testing and evaluation of the interface interactions. The final activity is the packaging and delivery to the customer.

3 Applying Ethical Directions on Product Integration

In this section, different ethical theories are related to the goals and practices for Product Integration as described in the CMMI and given in table 1. The analysis is done from the view of a software engineer responsible for the development of a specific function in a software product. We suggest that a conscious

decision by the individual to base actions on a specified set of ethical rules shape how successful interactions with co-workers will be, how well different tasks are performed and eventually how professional the development is executed.

For each of the theories, the question if it supports each of the goals is considered. An indication is given for each of the goals if the theory can be expected to support or oppose the intentions with the goal. There are also indications if our analysis is inconclusive. The conclusions in this section can be debated, and this is probably the most important result as this triggers the discussion regarding ethical considerations in the development of software products.

3.1 Act-Utilitarianism

For each situation, actions should bring the most good for the most people, and both immediate and long term effects should be considered.

Prepare for Product Integration. The idea that we should maximize the good for all people does neither help nor oppose the preparation. As an example, if the developer synchronizes the integration sequences ad-hoc in a successful way, this benefits most people as the goal of the project is fulfilled, but does not fulfill the goal of the process. On the other hand, the existence of this goal in CMMI indicates that there are benefits for the project and resulting product in having a strategy for the integration sequence.

Ensure Interface Compatibility. Also this goal may be supported or not. In general, the number of errors found in later stages of product development will be reduced if the interface compatibility is ensured. On the other hand, in the short term, this leads to additional work for all involved engineers, which may already have ensured interface compatibility in the development of the function.

Assemble the Product Components and Deliver the Product. The goal to maximize benefit is normally supporting the goal to assemble the product and bring it to the market. Of course there are exceptions where products do harm to many persons, but generally act-utilitarianism supports this goal.

3.2 Rule-Utilitarianism

For each situation, a set of chosen rules that should bring the most good for the most people is to be applied. If the rules are carefully selected, all three goals should be supported. However, one alternative to a well working product integration selected by some organization is to test extensively before a product is released. This may be a way to maximize the benefits for most people, but does not ensure that the goals for product integration are fulfilled.

3.3 Liberty Rights Ethics

The freedom to act for each individual should be respected. For the engineer, this could mean that as long as the result of the work is leading to the common goal, the means to that goal is a free choice for the engineer.

Prepare for Product Integration. The preparation requires that a strategy for the sequence of integration is selected and implemented. This is supported implicitly by the liberty rights, as it does not prescribe how the engineer meets the requirement on delivery on a specific time. Also the build up of environment and the specification of rules is supported, as this makes it easier for the engineer to understand the constraints for the development of functionality.

Ensure Interface Compatibility. Ensuring interface compatibility requires review. This can be considered as an infringement on the freedom to act for the engineer and that the results delivered are not respected. The conclusion is that this goal is not supported by liberty rights ethics.

Assemble the Product Components and Deliver the Product. The assembly and delivery of the components and the product depend on the result of the engineering work. Of course, the quality of the product is important, but the procedures to achieve it are not prescribed for this goal. Hence, the engineer is free to do what is required within the constraints, and consequently the goal is supported.

3.4 Welfare Rights Ethics

Transferred to a product development context, welfare rights ethics can imply the support from the organization to the individuals that need assistance to be able to perform the task.

Using this interpretation leads to an inconclusive result regarding the impact on all three goals for product integration. The needs for the individual may increase the focus on achieving the goals. One effect of this could be that engineers needing assistance to perform their task would always get it, and this would lead to better fulfillment of the goals. On the other hand, it might lead to sub-optimization and divert the work from the organization's goals. The influence will hence depend on the possibility for support within the resource constraints given for different parts of the organization.

Table 1. Relation between Product Integration goals and ethical directions

	Goal 1: Prepare for product integration	Goal 2: Ensure Interface Compatibility	Goal 3: Assemble Product Components and Deliver the Product
Act-Utilitarianism	Inconclusive, depends on situation	Inconclusive, depends on situation	Support, as it benefits a number of users
Rule- Utilitarianism	Support, but depends on the set of rules chosen	Support, but depends on the set of rules chosen	Support, but depends on the set of rules chosen
Liberty rights ethics	Support, as long as the defined areas of work are respected	Oppose, conflicts with the right to work without interference within the defined limits	Support, as long as the defined areas of work are respected
Welfare rights ethics	Inconclusive, depends on the possibilities for support from different parts of the organization	Inconclusive, depends on the possibilities for support from different parts of the organization	Inconclusive, depends on the possibilities for support from different parts of the organization
Duty ethics	Support, As long as the organization has a policy that supports the goal	Support, As long as the organization has a policy that supports the goal	Support, As long as the organization has a policy that supports the goal

	Goal 1: Prepare for product integration	Goal 2: Ensure Interface Compatibility	Goal 3: Assemble Product Components and Deliver the Product
Virtue ethics (MacIntyre 1984)	Inconclusive, depends on situation	Inconclusive, depends on situation	Support, as it benefits a number of users
Virtue ethics (Florman 1978)	Support, as professionalism is stressed	Support, as professionalism is stressed	Support, as professionalism is stressed
Ethical egoism	Inconclusive, depends on amount of work required and expected additional future work.	Oppose, for the development of a specific function, this is only additional work	Oppose, for the development of a specific function, this is only additional work as the function has already be tested
Community-oriented self-realization ethics	Inconclusive, depends on situation	Inconclusive, depends on situation	Support, as it benefits a number of users

3.5 Duty Ethics

What duty ethics imply depends on the rules and guidelines developed and used in the organization. For development organization, it is often expressed as policies, indicating the expected behavior from the developers. This leads to a common conclusion for all three goals for product integration, i.e. it depends on the policy for product integration in the specific organization. However, the general idea of having a policy would be supporting the goals as long as they are a part of it.

3.6 Virtue ethics (MacIntyre 1984)

Based on Aristotle, the virtue ethics described by MacIntyre express professions as valuable social activities. The target for the engineer would be to produce goods that can be internal or external, to adhere to standards of excellence, and to contribute to progress of the society. Internal goods can be personal (meaningful work), or public (medicine or electric power). External goods are earned through activities and include money, power and prestige.

Prepare for Product Integration. The preparation may be support as the standards of excellence is aimed for, but there may be a conflict with the aim of producing external goods as power and prestige.

Ensure Interface Compatibility. Again, this goal is basically supported, but there may be a conflict in the notion of meaningful work. Checking interface compatibility may be perceived by engineers to be unnecessary work, as they adhere to standards of excellence.

Assemble the Product Components and Deliver the Product. As for act-utilitarianism, the progress and the delivery of internal goods to the public are generally considered as good, and support this goal.

3.7 Virtue ethics (Florman 1978)

The virtue ethics described by Florman put the emphasis on the loyalty to the employer and on professionalism, but emphasis is on desirable features rather than on expected behavior. An engineer that does the job well is a morally good engineer. The two virtues are thus loyalty and competence. To act professionally is to work according to identified and described good practices. The CMMI is a collection of good practices that has been collected from a large number of successful product development organizations. The fulfillment of the goals described can be considered to be supported. However, if the

organization has selected a different model, with contradicting goals for product integration, this conclusion is invalid.

3.8 Ethical Egoism

Ethical egoism focuses on long term solutions that would maximize the benefit for the individual performing the actions. Hence, care for others is not in focus, and for an engineer developing functions for a product may not even care about the final product results. The goal would be to make sure that the individual contribution is observed as excellent.

Prepare for Product Integration. The result of the analysis is inconclusive as it depends on the amount of extra work that is the result of reaching this goal.

Ensure Interface Compatibility. To ensure the compatibility when integrating is additional work for an engineer doing development of a specific function.

Assemble the Product Components and Deliver the Product. Also for this goal, the individual engineer developing a function only sees additional work. This activity is perceived unnecessary as the individual functions have been tested in the development work.

3.9 Community-oriented self-realization ethics

Emphasis in this direction is on the commitments that individuals make, based on their self-interest, balanced with an understanding that self-realization depends on the relationships in the society. The commitments reflect what the engineer care about and govern the actions in development projects.

Prepare for Product Integration. The decisions in determining the strategy may be supported, but may also conflict with the interest of the engineer if the requirements limit the freedom for the developer.

Ensure Interface Compatibility. If this goal is to be supported by this ethical choice, the commitment from the engineer must be to follow the goal. Otherwise, the activities leading to ensuring interface compatibility will be considered unnecessary and not in the self-interest of the engineer.

Assemble the Product Components and Deliver the Product. The commitment of the individual engineer is often directed towards development of functionality in products that will contribute to society. This supports the goal of assembling and delivering the product.

4 Comparison between IEEE Code of Conduct and Different Moral Directions

To follow a code of conduct is considered to be one of the criteria for a profession to be mature (Ford & Gibbs 1996). For software engineering, the IEEE Code of Ethics (IEEE 2005) and ACM Code of Ethics and Professional Conduct (ACM 2005) are examples that have been developed and is also pronounced to be a sign of maturity (McConnel 2004). In Table 2, the ten guidelines included in the IEEE Code of Ethics are compared to the ethical approaches that can be considered to be the basis for them. Note that an approach that does not insist on but still does not contradict the statement is not indicated below. The interpretation in this section can and should be discussed as this most likely would increase the knowledge and awareness about the influence of the ethical directions on the software engineering discipline.

Table 2. Relation between IEEE Code of Conducts and ethical directions

According to the IEEE Code of Conducts, the members should agree:	Act-Utilitarianism	Rule-Utilitarianism	Liberty rights ethics	Welfare rights ethics	Duty ethics	Virtue ethics (MacIntyre)	Virtue ethics (Florman)	Ethical egoism	Community-oriented self-realization ethics
1. to accept responsibility in making engineering decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment	X	X			X	X	X		X
2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist Develop an integration plan based on the strategy	X	X	X	X	X	X	X		
3. to be honest and realistic in stating claims or estimates based on available data	X	X	X	X	X	X	X		
4. to reject bribery in all its forms	X	X			X	X	X		
5. to improve the understanding of technology, its appropriate application, and potential consequences	X	X			X	X	X		
6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations	X	X			X	X	X		
7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others	X	X	X	X	X	X	X		
8. to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin	X	X	X	X	X	X	X		
9. to avoid injuring others, their property, reputation, or employment by false or malicious action	X	X	X	X	X	X	X		X

According to the IEEE Code of Conducts, the members should agree:	Act-Utilitarianism	Rule-Utilitarianism	Liberty rights ethics	Welfare rights ethics	Duty ethics	Virtue ethics (MacIntyre)	Virtue ethics (Florman)	Ethical egoism	Community-oriented self-realization ethics
10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics	X	X	X	X	X	X	X		

5 Organizational influence

Many organizations explicitly select a set of guiding principles that are intended to ensure that employees base decisions on ethical principles common for the organization. However, observations made in industrial settings indicate that the influence on behavior is limited. Probable reasons for this include inadequate communication of principles and abstract definitions, but also organizational changes such as mergers, acquisitions, and lay-offs would make it difficult to convey an ethical direction to the whole organization. The individual selection will eventually determine the taken action.

6 Conclusion and future work

The influence on the effectiveness and efficiency in the workplace in general and on product integration in particular from the ethical codes followed is substantial. In most organizations, there is a mixture of different moral orientations which makes the analysis difficult. From our compilation and the reasoning above we conclude that the impact from different ethical theories is difficult to determine theoretically. An indication that a combination of several directions probably would give the best result is found through examination of different ethical codes for engineers. The gain from making ethical choices explicit is that it facilitates rational discussions and understanding of optimal choices in team work situations where different ethical attitudes always exist, but remain un-explicated. Examples are team members that are supposed to share their knowledge, information, results, resources etc with each other, but who might follow the line of ethical egoism.

Future work should include investigations in different organizations with and without explicit ethical policies. This would increase the understanding of the influence this has on individual behavior and on product development efficiency.

References

- ACM (2005) Code of Ethics and Professional Conduct, www.acm.org/constitution/code.html, (link valid April 2005)
- ANSI/EIA-632-1999, "Processes for Engineering a System", Government Electronic and Information Technology Association, Electronic Industries Alliance.
- Chrissis, M.B., M. Konrad, S. Shrum (2003) CMMI, Addison-Wesley, Boston, MA
- Florman, S.C. (1978) "Moral Blueprints: On regulating the ethics of engineers", Harpers 257
- Ford, G., N.E. Gibbs (1996) "A Mature Profession of Software Engineering.", SEI, CMU, CMU/SEI-96-TR-004
- IEEE (2005) Code of Ethics, www.ieee.org/portal/pages/about/whatis/code.html, (link valid April 2005)
- ISO/IEC 12207:1995, "Information technology – Software life cycle processes", ISO/IEC
- ISO/IEC 15288:2002, International Standard, "Systems engineering – Systems life cycle processes", ISO/IEC
- MacIntyre, A. (1984) After Virtue, 2d ed. South Bend, University of Notre Dame Press
- Martin, M.W., R. Schinzinger (2005), Ethics in Engineering, Fourth edition, McGraw-Hill, New York, NY
- McConnel, S. (2004), Professional Software Development, Addison-Wesley, Boston, MA