Moral Intelligence for IT Production

P.P. van den Bosch

Moral Inteligence for IT Production

Find more info at: Ethics4industry.com

With business cases taken from

Pieter .P van den Bosch: *Moral Intelligence for IT production*
Master Philosophy of Science, Technology and Society

Supervised by
Prof.dr. Philip Brey (Department of Philosophy, UTwente)
Dr. Aimee van Wynsberghe (Department of Values, Technology and Innovation, TUDelft)
Dr. Luís Ferreira Pires (Faculty of Electrical Engineering, Mathematics & Computer Science, UTwente)

Location
UTwente, Enschede
The Netherlands

August 21, 2017

Preface

Innovative technology has always fascinated me. When I was six years old, my father programmed a Yahtzee Dice Game in Visual Basic on our home computer. It was fascinating for me to see the aspects of design and seeing code being written right there on the screen. Later, he gave my brother and me a book that taught us how to program that same game. We saw how our words were being translated into actions, as if it was magic. That was the turning point when I knew what to do in my life—something creative that included designing and writing software.

I was creating technology on a very small level in a safe, a controlled environment then. Twenty five years later, I present this Master's thesis as a result of my bachelor's background in Business Engineering (Technische Bedrijfskunde, HBO education), the courses that I took during the Master 's degree of Business Administration, and finally finishing the Master 's degree Philosophy of Science, Technology and Society at the University of Twente.

Many IT projects I have worked on involved engineering, analysing business contexts, and collaborative work with industry leading companies, which inspired me to write this thesis. The most challenging aspect was how to connect three different domains with each other—namely, philosophy and ethics; software engineering; and business. This ultimately led to a multidisciplinary framework for management, ethicists, engineers, and consultants to incorporate ethics throughout their design.

Perhaps the most interesting aspect of technology creation is connecting the dots that ultimately create technology, such as merging different perspectives, views, systems, and the incorporation of corporate dynamics. Future innovations and breakthrough discoveries may all depend on how well these elements integrate and are supported throughout development. This, of course, is a big challenge for the world.

With this Master's thesis, I hope to contribute a complete new methodological framework to support ethical development throughout design processes.

Firstly, I would like to express my sincere gratitude to my advisor Prof. Brey for his continuous support, guidance, patience, motivation, and knowledge during my Master's thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Aimee van Wynsberge for her insightful comments and encouragement, but also for the hard questions that led me to widen my research from various perspectives.

Finally, my deep and sincere gratitude goes to my family for their continuous and unparalleled love, help, and support. I would not be writing a Master's Thesis if it wasn't for them. My (adoptive) parents brought me to the Netherlands where I could partake in higher education that has culminated in me finishing my university Master's degree. I am sincerely thankful for this.

Pieter van den Bosch

ABSTRACT

Moral Intelligence for IT production

Intelligence is the ability to adapt to change
-Stephen Hawking

There is a need to consider values in designs and one major approach that has been proposed is Value Sensitive Design (VSD). Value Sensitive Design is a theoretically grounded approach that accounts for human values in a principled and comprehensive manner throughout technology design processes. It employs an integrative and iterative tripartite methodology, consisting of conceptual, empirical, and technical investigations. It was developed by Batya Friedman and Peter Kahn at the University of Washington from the late 1980s to the early 1990s (Friedman, Kahn, & Borning, 2008).

However, a major problem has been formulated for VSD: "Perhaps the most apparent challenge of engaging in VSD outside the academic laboratory is the inevitability of confronting competing values within varied design contexts" (Zimmer & Manders-Huits, 2009, p6). In other words, VSD somehow lacks the methods of confronting competing values in different design contexts. Several problems have been addressed within the operationalisation of VSD, which will be dealt with in this thesis.

Following up on these problems, the following research question has been formulated:

"How can Value Sensitive Design be extended towards the IT industry and integrated in the design and production process of IT companies?"

To answer the research question, two distinct sub-questions were formulated:

- How to embed ethics in VSD and in production so that values can be operationalised?
- How to operationalise VSD within a corporate IT setting and look at these design contexts?

Why is this thesis so important, you may ask? Contemporary VSD literature provides insights into the problems during VSD operations and the issue of confronting competing values during the practice of VSD (Borning & Muller, 2012; Davis & Nathan, 2013; Friedman, 1999; Grunwald, 2001; Lanzo, 2003; Maedche, 2017; Manders-Huits, 2011; Pesch, 2015; Spiekermann, 2015; van den Hoven, Vermaas, & van de Poel, 2015; Zimmer & Manders-Huits, 2009).

In order to solve such problems, a novel theoretical framework is developed to support the incorporation of ethical values into the theory and methods of VSD. This theoretical framework can be used to build a monitoring and reporting learning system, which I refer to as a moral framework that can be iteratively used throughout the development of a technology. This moral framework here refers to a "Moral Intelligence framework" (providing pragmatic steps, logic, and problem solving) that contains three unique components that explicate several important methods.

Step 1) A Moral Intelligence Capture component, which allows any person to start capturing ethical issues by choosing from a selection of capture methods to gather ethical data on different levels of technology. Its major function is to allow learning and data collection (intelligence).

Step 2) A Moral Intelligence Design framework that deals with the varied design disciplines within the IT industry and links this to captured values and tooling. Its major function is to allow ethical monitoring during development and iterative improvements within teams throughout design.

Step 3) A Moral Intelligence Deployment framework that allows for the integration of Moral Intelligence within important corporate contingencies that may speed up the process of adopting Moral Intelligence within corporate cultures. The deployment framework includes seven distinct systems that are related to the incorporation of Moral Intelligence within industry contexts. Its major function is to report ethical improvement back to all organisational levels within a company.

These components are described in detail herein and will form pragmatic steps for companies to embed ethics within their production methods. The foundation of Moral Intelligence has been built on parts of the tripartite methodology of VSD (I prose that Moral Intelligence can also be a standalone operational theory, not necessarily dependent on VSD, but still embodying its tripartite methodology.) This is done intentionally so that Moral Intelligence can grow as a separate theory but also fall back on VSD theory when necessary. Promoters of VSD may also see this as a benefit to adopt more theories of Moral Intelligence, without feeling their core VSD values are threatened.

Therefore, I will promote Moral Intelligence as a theory that could be linked with, and as an iterative progression of, VSD. If VSD requires more ethical reasoning than Moral Intelligence, that is a decision for the VSD participants.

A strong focus on innovation within the IT industry context

Following up on the proposed Moral Intelligence Framework, several methods of design were made explicit to allow for a unique integration of VSD into building technology processes (Van Wynsberghe, 2012). Industry standards — e.g. Model Driven Development (MDD), Agile Development, and Scrum — are scrutinised and the best parts are taken together to form a pragmatic and integrative approach. Qualitative interviews have also been provided from three IT companies (InterDC, El Niño BV, CAPE Groep BV) situated in Enschede, The Netherlands, to strengthen the foundation of the steps to be taken. These companies provide a better argument for the developed VSD approach and also aim to provide practical knowledge for a possible match between VSD, Moral Intelligence, and design disciplines. This investigation of industry standards gathers and unifies the disciplines and experiences that are already available within current industry practices, which may not always be taken into account by ethicists. This especially includes the roles of the engineer, consultant, and ethicist, which are unified by presenting a Moral Intelligence 'disciplinary matrix' to allow improved teamwork during a collaborative project to collect specifications or to build a product together.

Changing the IT industry with the these novel methods

This thesis takes active leadership in creating pragmatic steps for VSD implementation within the IT industry by connecting the ethical and engineering worlds through creation of a pragmatic framework. While the VSD literature covers some models of development, I argue that many do not fit the progress of engineering in practice because disciplines also have to be taken into account, which is harder to describe. Essentially, what is missing is an approach that goes in-depth within the disciplines found in an organisation and that effectively tries to link with VSD. In response, an intuitive and iterative three-step model is presented to ease this adoption. Moral Intelligence Capture, Design and Deploy allows for the ethical specification of all steps required throughout development. This method is evaluated with leading industry standards and has been cross-examined with three real business cases in this thesis.

Moreover, it is extremely important to look at design disciplines and to make these more explicit when integrating VSD. While models of development are largely discussed in theory, real-life situations and disciplines are harder to describe because of the context. This is apparent when we think about specific cases of coding, development methodologies, and corporate contingencies such as corporate culture.

In other words, there is an important gap between the VSD philosophy and its practitioners. Although some organisational factors have been analysed, no specific recommendation has been given, whereas this thesis develops an important deployment framework that can be used by ethicists, developers, consultants, engineers, businesses, and management.

To support further innovation for VSD and Moral Intelligence research, a website (Ethics4industry.com) and component platform (capture, design & deploy) showcase the components developed during this thesis as suggestions to take active leadership in Moral Intelligence and to execute the recommendations herein. The methods, cases, and theory are presented on this website and a mission, vision, and roadmap have been published to support ethical actions and as practical reporting tools. With the right leadership and vision (executing learning, monitoring and reporting), the developed components can be adopted by corporations to further build a Moral Intelligence system that can be scaled and implemented within small to large enterprises.

Research Highlights

- Throughout this thesis, new important building blocks for VSD are developed for integration within the IT industry.
- This thesis examines industry standard design methodologies to find a possible match for VSD and important engineering disciplines: e.g. VSD is extended for use by engineers, IT consultants, ethicists, and many more disciplines.
- Central for the improvement of VSD is Moral Intelligence, introduced as a core component of VSD, which allows for learning, monitoring, and reporting ethical issues.
- A full Capture, Design and Deployment method for VSD is developed for incorporation within a corporate IT setting.
- Three business cases are used to provide insight into the field of engineering to uncover possible matches between VSD and engineering.
- This thesis aims to bridge the gap between philosophical problems, the engineering domain, and management theory that are involved when adopting VSD in corporate settings.

Keywords: VSD; ATE; NEST Ethics; Moral Intelligence; Ethical Assessment; Model Driven Development (MDD); SCRUM Analysis; Agile Development; Ethics; Organisational Theory; Continuous Integration (CI); Software Development

CHAPTER 1

Introduction to Value Sensitive Design

Action is the foundational key to all success.

(Pablo Picasso, n.d.)

Welcome to today's design challenges and the problems for the next revolution in IT design. What are these problems and challenges? Well, technology seems to have virtually no limits, which forces us to think ethically about certain changes technology can bring.

If you want to check someone's personal information, Google may be a good starting point or you can try your luck with Facebook. If you want to build an invasive application as a software engineer, the risks can be found in the privacy scandals that surrounded Facebook in 2013 (Kelly, 2013) and the NSA scandals ("Edward Snowden: Leaks that exposed US spy programme," 2014), which include use without explicit permissions, the GPS functions, background data mining, or hidden third-party services that are broadly available now. People using smartphones may not be aware what other functions are active in the background. These misuses of technology can heavily conflict with a person's highly appreciated values of privacy or consent.

Another example of the influence of technology and the concerns that it brings is the Apple VS FBI case (Kharpal, 2016), which marked one of the highest-profile clashes in the debate over encryption and data privacy between the government and a technology company. Law enforcement authorities, e.g. the FBI, say that encryption used by the likes of Apple makes it harder for them to solve cases and stop terrorist attacks. Technology firms such as Google and Microsoft have kicked back, saying that encryption is key to protecting user data from hackers. Apple objected to the FBI's demand that Apple build a back door into the iPhone because they believed it was wrong and would set a dangerous precedent. Luckily for Apple, a hearing was postponed after the government reported it had found a third party that could unlock the iPhone, which resulted in the FBI dropping the case.

These examples highlight that while IT holds great power, there are also many cases of intrusive applications that can conflict with people's privacy or consent. Scandals like the ones listed above raise important ethical questions, such as whether engineers and other individuals have a responsibility to develop products with due consideration for the consequences of the product's intended and un-intended user features. Do engineers have a responsibility to consider their duties to design good products? Should engineers be trained to reflect on their professional role as designers or developers?

This raises the important question: how can we make "good" IT design really work in a corporate setting? As these questions are being raised more frequently about technology and ethical values by media or academia, society and developers may require a better design methodology to produce more ethically-valued products throughout the production process.

Fundamental questions like these will be answered in this thesis. To acknowledge these problems, firstly a very important question that we should ask ourselves is "can technology have embedded values?" Is there a relationship between the technologies we use every day and the values we hold dear as a society?

According to the embedded values approach, the answer is "yes", which holds that computer systems and software are capable of harbouring embedded or "built-in" values (Brey, 2009). For example, computer programs can be supportive of privacy and the free flow of information, or instead go against the realisation of such values. This shows how the technologies we use daily can have a direct impact on the values we appreciate in society.

Values can, for example, through their incorporation in technology, shape the space of action of future users, i.e., they can affect the set of affordances of and constraints to users (Maedche, 2017). Every design, artefact, and system is shaped by the values, ideas, and world views of the designer and builder. That applies to architecture, software engineering, product design, synthetic biology, material science, and civil engineering. A design is a consolidated set of choices made by designers, developers and engineers. Via their designs for systems and artefacts, they come to have an incredible impact on the lives of others: cables, code, search and reach algorithms, standards, ontologies, authorization matrices, menus, voting procedures, aggregation mechanisms, recommender systems, reputation systems (Maedche, 2017, p298).

Another way to think about these artefacts is to see them as social-technical artefacts. They affect us not only by their physical or material properties, but also by properties they acquire as systems and devices embedded in lager material and social networks and webs of meaning (Nissenbaum, 2010, p6).

When we see how companies treat this relationship and the terrible consequences that can occur, it may demand a "call for action", e.g. taking pragmatic steps to better intervene during the

technology engineering process to minimise the impact on people's values.

1.1 Introduction to Design for Value

In order to shape these environments in which we function as moral beings in a responsible way, we need to express or ''design in'' our shared moral values. 'Design for Value' recognizes design as a far richer process since it can realize our functional requirements and our moral values. Design for Values integrates design with our values and allows an active value-driven steering of and intervention in technological development. Design and designers can be frontloaded with moral and social values so they can realize these values and be held accountable (van den Hoven et al., 2015,p3). Values should therefore be seen as a sort of supra or non-functional requirements for which we can and ought to design. Moreover, it will become more and more important in the future to design systematically for moral, legal and social requirements (Maedche, 2017).

The practice of Design for Value is characterized by a diversity of approaches, theoretical backgrounds, designed values, and application domains.

Introduction to Value Sensitive Design

Value Sensitive Design (VSD) represents a pioneering endeavour to proactively consider human values throughout the process of technology design (van den Hoven et al., 2015). VSD is primarily concerned with values that centre on human wellbeing, human dignity, justice, welfare, and human rights. Value-Sensitive Design connects the people who design systems and interfaces with the people who think about and understand the values of the stakeholders affected by the systems (Friedman, 1999).

Value Sensitive Design emerged in the 1990s as an approach to the design of information and computer systems that account for human values in a principled and comprehensive manner throughout the design process. VSD emphasises the moral perspective (e.g. privacy, security, trust, human dignity, physical and psychological wellbeing, informed consent, intellectual property) and also accounts for usability (e.g., ease of use), conventions (e.g., standardisation of technical protocols), and personal predilections (e.g., colour preferences) within a graphical interface ("VSD: projects," 2011).

Central to a Value Sensitive Design approach are analyses of direct and indirect stakeholders; distinctions among designer values, values explicitly supported by the technology and stakeholders; individual, group and societal levels of analysis; the integrative and iterative conceptual, technical, and empirical investigations; and a commitment to progress, not perfection.

VSD Theory

VSD is related to values in a two-fold way. First, it starts from the observation that design and technology may impact upon values: (new) technology developments may enhance, threaten, or transform existing values. For example, the advent of social networking sites has changed existing conceptions of privacy, especially for younger generations. Secondly, values are fostered and built into design by means of VSD; it seeks to identify values considered to be of importance for a target group, say a certain company, society, or user group, and subsequently safeguards these values by designing them into technology (Manders-Huits, 2011, p4).

VSD researchers

Batya Friedman is currently a professor in the Information School, Adjunct Professor in the Department of Computer Science, and Adjunct Professor in the Department of Human-Cantered Design and Engineering at the University of Washington where she directs the Value Sensitive Design Research Lab. Batya pioneered VSD, an approach to account for human values in the design of information systems. First developed for human-computer interactions, VSD has since been used in information management, human-robotic interactions, computer security, civil engineering, applied philosophy, and land use and transportation.

Her work has focused on a wide range of values, which include privacy in public trust, freedom from bias, moral agency, sustainability, safety, calmness, freedom of expression, and human dignity; along with a range of technologies, such as web browsers, urban simulation, robotics, open source tools, mobile computing, implantable medical devices, computer security, ubiquitous computing, and computing infrastructures. She is currently working on a multi-lifespan information system design and on methods for envisioning new ideas for leveraging information systems to shape our futures. Voices from the Rwanda Tribunal is an early project in this multi-lifespan information system design program. In 2012, Batya was awarded the SIG-CHI Social Impact Award ("Batya Friedman | Information School | University of Washington," 2016).

Peter H. Kahn, Jr. is a Professor in the Department of Psychology and Director of the Human Interaction With Nature and Technological Systems (HINTS) Lab. The HINTS Lab seeks to address two world trends that are powerfully reshaping human existence from a psychological stance: (1) the degradation, if not destruction, of large parts of the natural world, and (2) unprecedented technological development, both in terms of its computational sophistication and pervasiveness. He received his Ph.D. from the University of California, Berkeley, in 1988.

His publications have appeared in journals such as Child Development, Developmental Psychology, Human-Computer Interaction, and Journal of Systems Software, as well as in proceedings such as CHI, HRI, and Ubicomp ("Peter Kahn," n.d.).

VSD and other engineering domains

VSD has been embraced by a broad range of domains. Dr. Aimee van Wynsberge, has been investigating ethics and value incorporation within the field of robotics (Van Wynsberghe, 2012). Prof. M. J. den Hoven has been investigating ethics and information technology with research interests in the Technology, Computer Ethics, Value Sensitive Design, Responsible Innovation and Privacy topics ("TU Delft: Prof. M.J. (Jeroen) van den Hoven," n.d.). Prof . Sarah Spiekermann is author of the book "Ethical IT Innovation: A Value-based System Design Approach". Her main research interests are electronic privacy, disclosure behaviour, and ethical computing (Spiekermann, 2015). Alan Borning is a professor in the Department of Computer Science and Engineering at the University of Washington and an adjunct professor in the Information School. His current research interests are in human-computer interactions, particularly as applied to tools for civic engagement and for improving public transit information systems, and in using and extending Value Sensitive Design in this work ("Alan Borning - CROW 2016," 2016).

Methods of VSD Technical, Conceptual and Empirical Investigations

Design occurs through a tripartite investigation consisting of three phases: conceptual, empirical, and technological. These investigations are intended to be iterative, allowing the designer to modify the design continuously. Examples are given of these tripartite investigations in the illustration below.

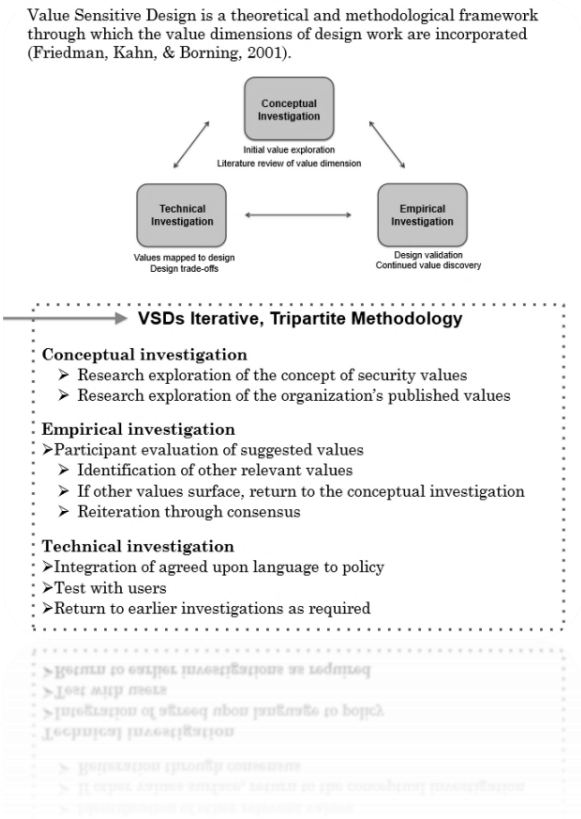


Figure 1.1:
http://cec.nova.edu/research/fall_2013_poster_session/documents/Diane%20Blitstein%20Solomon.pdf

Conceptual investigations

In short, conceptual investigations involve no costly empirical analyses, but instead thoughtful consideration of how stakeholders might be socially impacted by one's technological designs. It focuses on the philosophically informed analyses of the central constructs and issues under investigation. This includes questions like, What are values? Whose values should be supported in the design process? What do values such as privacy mean? (Friedman, H. Kahn Jr, & Borning, 2001).

Technical investigations

Technical investigations focus on how existing technological properties, features, and underlying mechanisms support or hinder human values (Friedman, H. Kahn Jr, & Borning, 2001).

Emperical investigations

Empirical investigations examine how stakeholders understand individual values in the interactive context, e.g. understanding, context and experiences (Davis & Nathan, 2013, p16). More in-depth details and discussions about these investigations will be covered throughout Chapter 4.

Examples of VSD projects in practice

Currently, VSD is mainly practiced in Research and Development (R&D) environments. The vsdesign.org website displays several projects in which VSD research has been carried out. Fewer projects, however, focus on the IT industry.

Only one published project showcased on the vsdesign.org website that was carried out in the IT industry was the Microsoft COOP project.

Project Description: Microsoft COOP

The CodeCOOP is a groupware application designed to help Microsoft Research employees who participate in engineering activities to (a) share information (including code) and (b) build community. An internal web application, the CodeCOOP, supports two types of content: a code repository (including the ability to post and search code) and conversations (including the ability to ask questions, reply to questions, as well as search prior questions and answers). In addition, the CodeCOOP implements an email digest to manage the information flow of requests. This project represents one of the first efforts to extend the theory and methods of Value Sensitive Design to an industry setting, thus providing a proof-of-concept for the viability of this approach within industry ("VSD: projects," 2011).
Source: http://www.vsdesign.org/projects.shtml#consent_online

1.2 Extending VSD towards the IT industry

VSD happens to be mostly practiced in the lab instead of being widely used in the IT industry (Miller, Friedman, & Jancke, 2007). From vsdesign.org sources, it may be concluded that VSD is not well known in industry perhaps due to the minimal amount of completed industry projects or that the VSD methodology is not yet promoted and adapted to fit with the IT industry yet. VSD is mainly used for small research teams of R&D projects, such as robotic pets & the elderly, robotic pets & preschoolers, or a research project on urban areas called UrbanSim ("VSD: projects," 2011). VSD is also often seen as costly and manipulated by managers to increase the efficiency of a project (Zimmer & Manders-Huits, 2009).

In this thesis, an approach for expanding the VSD theory to use in the IT industry is proposed. Information technology is the use of any computers, storage, networking, and other physical devices, infrastructure, and processes to create, process, store, secure, and exchange all forms of electronic data (Rouse, n.d.). I will also include all IT activities that involve software building, particularly taking software development companies as an example.

The software industry consists of that part of computer programming that is traded between software-producing organisations and corporate or individual software consumers (Campbell-Kelly, 2003). Core activities include programming services and selling software products.

According to some researchers, VSD is not successful when it comes to the implementation and operationalisation of values for applications in technical designs (Zimmer & Manders-Huits, 2009). I argue that this can be due to the fact that product development and design in an IT context are often guided by a (regulatory) production methodology and, within the approach of VSD, this crucial analysis and implementation is lacking. I will show why this analysis is key and how it can be carried out as part of the VSD approach. For all the benefits that VSD provides, it is still somewhat difficult to standardise VSD methods because they does not do a good enough job of creating a regulatory (ethical) framework.

For example, engineers have been taught to work using their own "workflow" and have some degree of freedom but are mostly guided by the industrial context, such as production methodology, tooling, and the company structure. This means their procedures often don't include ethics or, even worse, fail to look at values that may be at stake outside the scope of the project. Also, certain types of development or design methodology that engineers work with may limit their ability to reflect on their actions, whereas other types do not. In other words, to be effective, a better regulatory framework may be required that can guide or lead people.

Essentially, during this thesis, we may ask ourselves whether guiding design is really enough to promote VSD. I aim to develop a method that not only steers ethics in IT development but also takes a leading role in its operationalisation by creating a methodologically rich approach that takes active leadership by developing the appropriate ethical tools and managerial instruments. The research question that will provide guidance for developing these tools is stated next.

1.3 Research question for extending VSD towards the IT industry

I formulate my research question as follows: *How can Value Sensitive Design be extended towards the IT industry and be integrated in the design and production process of IT companies?*

In order to extend VSD outside the labs, one may first ask what current problems VSD faces when extended outside labs? Contemporary literature may provide very interesting insights into VSD problems and

challenges. This will provide a roadmap for VSD improvement. After knowing the current situation, an approach may be developed to extend VSD outside lab settings.

Sub-question one
What are the current problems within VSD methodology when extending VSD outside labs?

Sub-question two
What approach can be developed to overcome problems so that VSD can be better extended towards the IT industry?

In the next chapter, I will first answer question one by scrutinising contemporary VSD literature. Then I will propose a research framework that will answer question two because many steps may be required to answer this follow up question.

CHAPTER 2

Challenges to Value Sensitive Design

Coming together is a beginning,
Keeping together is progress,
Working together is success.

(Ford, n.d.)

VSD was introduced in the previous chapter, as was the suggestion to extend VSD towards the IT industry. To develop an approach, a sub-question was formulated:

What are the current problems within VSD methodology when extending VSD outside labs?

Contemporary VSD literature indicates that VSD still has some remaining challenges in its practices. Literature on "Next Steps for Value Sensitive Design" (Zimmer & Manders-Huits, 2009) chart interesting continuing problems for VSD and its extension outside labs (Miller et al., 2007). Zimmer & Manders-Huits have a very powerful statement on VSD: "Perhaps the most apparent challenge of engaging in Value-Conscious Design/VSD outside the academic laboratory is the inevitability of confronting competing values within varied design contexts" (Zimmer & Manders-Huits, 2009, p6).

According Zimmer & Manders-Huits, the first problem is the inevitability of confronting competing values within a varied design context. I argue that the design context can be seen as a regulatory framework for developers addressing tooling and production methodologies that often don't match the integration of values. In other words, it is essential to build support for values and in order to do this, we need to confront values with the specific design context. But, what are these design contexts? These remain to be determined and are not made explicit by Zimmer & Manders-Huits.

Secondly, several important problems have been addressed, which I have categorised as operational problems (e.g. confronting, competing values in practice) and philosophical problems (e.g. the theory for VSD). Operational problems are defined as the processes or activities required so VSD can function within an organisational context and philosophical problems including examining what a good ethical approach would be.

In practical operationalisation, questions are more for designers, technical staff, management, and everyone involved within the corporate and practical context (e.g. daily operations). The philosophical issues need to be supported by ethical theory and often come to the surface when operational problems are discovered by a team, which forces ethical reflection due to bad consequences.

2.1 Operational problems with VSD

The below operational problems have been formulated by Zimmer & Manders-Huits. Some may sound similar. I have included them intentionally to chart VSD trends and to provide a holistic view of the VSD situation. Operational problems are defined as the processes or activities required to make VSD applicable within an organisational context.

The first problem we encounter, according Zimmer & Manders-Huits, is:

"The difficulty to properly explicate and translate ethical considerations to workable requirements and specifications for the other project participants actually building the system" (Zimmer & Manders-Huits, 2009, p4).

Often the values that are discovered can't be easily put into a design context even if specific requirements have been formulated. Technology design is constructed in the literature as an ambiguous term. What is the design processes to which VSD refers? How can the values be operationalised? How will ethical considerations be incorporated into design processes? This general question seemingly conflicts with a workable design context.

The problem is how to operationalise values so they can be incorporated into technological design.

It is also problematic to information on supporting values throughout the full development process. For a design team, it is very often difficult to put ethical considerations on the roadmap and continuously worked on them. One example is a project management team who did not share the same commitment or appreciation of the demands as a previous team. This can lead to a hostile working environment.

The creation of a hostile environment (Zimmer & Manders-Huits, 2009, p4-5)

The creation of a hostile environment also contributes to reasons why some VSD projects may fail. A hostile environment may be created due to an existing corporate culture. Some companies have strict

guidelines and protocols for product development. Some companies stimulate openness and fairness towards employees, while other companies may create a hostile environment with not much room for innovation or awareness of the product's risks. A hostile environment may be created by managers, a team member, protocols, or the lack of the right resources. According to Sarah Spiekermann author of ethical IT innovation, Value creation or destruction through IT regularly plays out not only on the individual level but also at the corporate and societal level (Spiekermann, 2015,p15).

The rejection of ethical concerns to expedite the project (Zimmer & Manders-Huits, 2009, p4)

In sum, these four questions show that there are operational concerns regarding the operationalisation of VSD. During this thesis, an approach will be developed to solve these concerns.

2.2 Philosophical problems with VSD

Artefacts are believed to have a kind of morality. According to Verbeek, artefacts have moral relevance given their role in mediating one's experience and practices. He describes technology mediation whereby technology helps to shape human actions and perceptions and create new practices and ways of living (Verbeek, 2008, p92). Designers materialise morality; consequently, technology design is inherently a moral activity. Choosing values over the other design factors can therefore be morally problematic.

Sara Spiekermann discusses in her work that we must built machines that actively embrace, embed and foster values. She begins with mentioning that knowledge, freedom, autonomy, security, health, friendship and dignity are undoubtedly important for everyone. She began with mentioning the knowledge value, outlining how building information for the machine age requires ethical conduct at all stages of the knowledge creation process. For people to thrust machines and machine operations, data and information must be collected in a legitimate way. Data quality and transparency is important for data aggregation (Spiekermann, 2015,p152).

Extending this line of thought, three challenges have been introduced related to VSD in an industrial context versus an academic one: (2.1) Confronting competing values, (2.2) identifying the role the values advocate, (2.3) the justification of a value framework (Zimmer & Manders-Huits, 2009). I will explain these problems step-by-step.

Confronting competing values

Rather than benefiting from working within the academic sphere, committed to the primacy of designing for moral values, design contexts outside academia often include stakeholders whose goals might come into conflict with the protection of these values. Zimmer & Manders claim that, for a fully engaged and pragmatic application of VSD to be successful, it must ensure that values are not only discovered and clarified, but also consciously and deliberately built into design, even if such an embedding of values conflicts with other design objectives. This means, however, that values also generate conflict during their selection.

For all the benefits VSD promotes, it lacks ethical theory to make choices about these value conflicts (will be further explained also in Chapter 4). However, when you participate in VSD, it can also be seen as an ethical (moral) activity, as explained earlier. Furthermore, during a brainstorm session with van Wynsberge (see Appendix H), several reasons why VSD includes ethics were discovered. First of all, decisions to resolve value conflict are ethical reasoning. This manifests itself throughout design: what is a good (technical) design and what is bad (technical) design for users? Moreover, these values manifest in the resulting technology artefact, which forces us to think about the accountability of designers and the technology itself. Who will be responsible for an autonomous robot when bad things happen? Will that be the designer or the robot?

While all this seems like an operational problem due to practical situations, I also refer this problem to VSD core methods and its evaluation. To allow a better confrontation of competing values, we have to allow for a framework for the discovery of ethical issues within the technological artefact first. Unfortunately, VSD lacks ethical theory about design or the interpretation of competing values. How should we deal, for example, with the designer's versus the user's values? How can we make a balanced choice?

This raises some questions: what method could be designed for confronting competing values? How can we gather values based on ethical theory? How to use them in a technology context? All of these questions will be answered in Chapter 4 and later chapters.

Identifying the role of the values advocate (Zimmer & Manders-Huits)

In VSD, ethical judgment is required, which is often provided by a person, such as a values advocate. However, there is no clear role for this value advocate in software development. Zimmer & Manders envision an array of roles that the values advocate can take within a particular design context. It is suggested that a leadership role is most preferred.

I suggest, however, looking at a design team first and identifying whether the values advocate is more than one person. Therefore, we first need to know which roles can be clearly identified in a design team. Is it simply a programmer, designer, or are there also other important roles? Furthermore, what are the dynamics of a values advocate within this design team? And, what is each member's individual contribution? These are important questions related towards the ethical role of VSD.

The justification of a value framework

According to the contemporary literature (Zimmer & Manders-Huits, 2009) and also an interview with Dr. Van Wynsberge, VSD lacks ethical theory for the interpretation of values. What is good design in the first place and how will values fit? While VSD focuses on values, it does not mention ethics in general so, importantly, an ethical framework for VSD is missing. It does not include a normative framework as consequentialism, virtue ethics, or deontology ethics do. Each company deals with design in a different way and current production methods and methodologies are robust systems that don't allow VSD to be used in an effective manner. These problems ultimately lead to a more general question that I have formulated to account for these issues:

How we can embed an ethical approach in VSD and have VSD supported throughout production

A research strategy is now required, which will be explained in the next section. This research strategy will answer question two of the previous chapter "What approach can be developed to overcome problems so that VSD can be better extended towards the IT industry?"

2.3 The Research Design & Approach

Now that VSD problems have been formulated, I argue that engineers have been taught to work using their own "workflow" with some degree of freedom, but are mostly guided by *production methodology, tooling, and company structure* that can cause most operational problems (1.1-1.4) mentioned earlier. In essence, this specifically means that procedures often don't include ethics considerations. In addition, certain types of development or design methodologies limit engineers to be flexible and to reflect on their actions, whereas other types do not.

To successfully bridge this gap, I explicate a design context. I will explain several IT standard design methodologies and analyse how to hook VSD into these to create a new adjusted approach to bridge the gap between important VSD theory and its operationalisation (1.1, 1.2, 1.3, 1.4).

To embed ethical reasoning in VSD that will provide a good basis for the problems 2.1, 2.2, 2.3, 2.4, I will develop a new approach that allows for gathering values and the usage of ethical theory in VSD.

How can Value Sensitive Design be extended towards the IT industry and be integrated in the design and production process of IT companies?

Research strategy to solve the above research question:

1 How to operationalise VSD within a corporate IT Setting?

In order to answer this, the below questions need to be addressed:

1.1) How to solve the difficulty of properly explicating and translating ethical considerations into workable requirements and specifications for project participants who are actually building the system?
1.2) How to solve the problem of operationalising the values so they can be incorporated into technological design?
1.3) How to solve the creation of a hostile environment?
1.4) How to solve the rejection of ethical concerns to speed up the project?

2 How to embed ethics in VSD and in production so that values can be operationalised?

In order to answer this, the below questions need to be solved:

2.1) How to embed an ethical approach in VSD and in production?
2.2 How to confront competing values?
2.3) How to identifying the role of the values advocate?
2.4) How to select the justification of a value framework?

2.4 The scope of extending VSD towards an industry framework

Now that the sub-questions have been formulated, an approach needs to be developed to solve these problems. Because many problems have been addressed (e.g. theoretical, operational, and philosophical), an integrative approach is suggested with an organisational context that covers the above aspects. In the following chapter overview, this approach is explained in detail.

Analysis framework by chapter

In this thesis, three dimensions will be covered to extend VSD towards the IT industry. These dimensions are **Methodology and methods** (reflection on the VSD philosophy to provide a basis for 2.1, 2.2, 2.3, 2.4), **Production** (deals with the embedment within corporate settings 1.1, 1.2, 1.3 and 1.4) and concrete **Interventions** (suggested tools and synergy of all the mentioned problems).

Chapters 3-4

During Part I, "Methodology and Methods'", I will focus on investigating multiple ethical theories to create VSD methods that offer a pragmatic approach to include ethics into design. Real cases of companies could be very useful when developing the methodology and methods. To build a stronger case for this thesis several interviews are also taken into account of IT

companies situated in Enschede, The Netherlands. Operational problems were earlier discussed but I argue that more background info may be required e.g. that of interesting IT companies to link methodology and methods to the practice. Therefore, several interviews were held to scrutinise ethical culture at IT companies and to investigate how VSD may be incorporated.

Chapter 3 answers: What is the current ethical culture and ethical practice at IT companies situated in Enschede? What may be developed to improve ethical culture?

Chapter 4 answers: what could be a moral framework that integrates with VSD to justify the value framework to ultimately help solve the problems (2.1)(2.2)(2.3)(2.4)?

Chapters 5-6

Part ll, Production (Embedding and Integration), translates the right basic set of tools and instruments for VSD. In this part, embedding and the integration/balancing of values will be a key focus. I aim to develop a production method to create a VSD community that could operate in an IT setting.

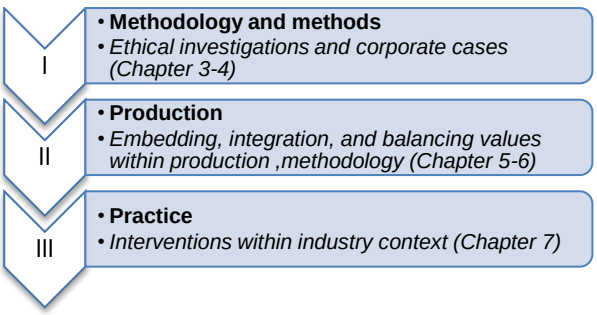
Chapter 5 answers: what design context could be compatible with VSD to answer (1.1-1.4)?

Chapter 6 answers: how to integrate VSD outside the academic laboratory and confront values within design contexts and apply (2.1, 2.2 2.3)? What kind of tools may be developed?

Chapter 7

Finally in Part lll, a list of interventions, tools and instruments are suggested to benefit practice. The list of interventions, tools and instruments may be used by anyone interested in incorporating VSD within an IT context. I argue that the dimensions covered in the thesis will form the basis of a solid VSD production methodology for industry. Below is an overview of the chapters and their pragmatic approach.

Table 2.1 Overview of the framework used in this thesis to extend VSD towards the IT Industry



2.5 Practical cases that will be used throughout the thesis

In order to provide both useful theoretical and practical knowledge, three companies will be interviewed (qualitative interviews) to provide cases of practical inquiry. These companies are all very active in the IT industry but in different branches (e.g. one data storage vs. two software developers). They will be scrutinised and interviewed about their design context, its flexibility in development, and their available resources for a VSD methodology that can provide a better understanding of practices within the industry but also their view on incorporating ethics or VSD.

The aim of using practical cases is that companies and VSD can mutually benefit from each other by allowing industry knowledge and VSD methodology to come together. These three companies may produce the following benefits for building a stronger implementation case:

1. They may be used to develop a long-term plan for VSD supported by organisational theory and vice versa.
2. They may help to get more insights of the current IT industry standards and the adoption of VSD practices within these standards.
    a. E.g. they could help to explicate VSD processes throughout organisations.
    b. Allow for a critical reflection on VSD methodology, e.g. by extending the roles of participants.

Table 2.2 VSD Industry Extensions

Extending VSD to industry by cases		
Company name	**Company type**	**Production method**
CAPE Groep BV[1], situated in Enschede, The Netherlands	Supply chain integrator, IT consultant	Model Driven Development (MDD)
InterDC[2], situated in Enschede, The Netherlands	Data warehouse and webhosting company	Agile development
El Niño[3], Situated in Enschede, The Netherlands	Software development company	Continuous integration (CI)
TriMM[4], Situated in Enschede, The Netherlands	Web-development company	Scrum

[1] Capegroep.nl
[2] InterDC.nl
[3] Elnino-ict.nl
[4] Trimm.nl

2.6 Conclusion and reflection

Several core problems were formulated in this chapter that chart a possible roadmap for extending VSD outside the labs. As an answer to the core problems, during the thesis, Methodology and Methods, Production and Practice will be developed to form the basis steps to extend VSD towards the IT industry.

The research design proposed in this chapter forms the steps and building blocks to create a new holistic VSD for IT approach answering the philosophical problems and the operational problems.

Qualitative interviews are used so companies and VSD can mutually benefit from each other by allowing industry knowledge and VSD methodology to come together. The next chapter will illustrate what ethical concerns are raised by IT companies and why ethical issues may be also included in VSD.

CHAPTER 3

A Case for Building Moral Intelligence

If you are working on something exciting that you really care about, you don't have to be pushed. The vision pulls you.

(Steve Jobs | Values.com, n.d.)

3.1 Charting ethical culture within Tech companies situated in Enschede

In the previous chapter, operational and philosophical problems were discussed and it was explained that VSD lacks ethical theory for the interpretation of values (Zimmer & Manders-Huits, 2009). How do these questions relate to practice? In order to get a better understanding on where VSD and ethics would fit within the practice of IT companies, several IT companies situated in Enschede were interviewed at the beginning of this research. Because the companies are not familiar with VSD, questions were asked about ethics to provide useful insight in the current ethical culture and ethical practice at IT companies situated in Enschede (see Appendix A). Their insights may lead to possible understandings of ethical conflicts.

The first company that was interviewed is the data warehouse, "InterDC", situated in Enschede. Its core activities are providing server storage, infrastructure, and web hosting.

CASE I
Interview Summary: Roland Kamphuis, CEO, InterDC, Enschede (Data Warehouse, small-sized company).
The following are the questions asked:.

1. What are the potential areas where you could include ethics in your IT company?

InterDC stresses the importance of keeping the data of their customers secure and the importance of data protection for both the digital and the physical world (e.g. the physical servers stored in the data warehouse). "It is important that the personal data does not become publicly known or found freely on the web," CEO, Roland Kamphuis, emphasises.

Having the access to ethics and knowing what to do in specific cases would help the data centre to steer development by providing guidelines to safely use software and to explicate how customers could behave more responsibly.

Furthermore, Roland mentions that the delegation of this responsibility and creating awareness is important. "We need to ask ourselves, what does our software do for customers? And how do you control your information?" The CEO mentions that he could envision great benefits in the creation of an operational guide for escalations and security: "How do you create safety or create applications that do not steal private information?"

2. What are the positive and negative effects of incorporating ethics in different IT development economics in your company?

For InterDC, good ethical software development costs a lot of money and therefore time and investment, which is mentioned as a negative aspect. On the positive side, good marketing and Corporate Social Responsibility (CSR) can boost the quality of the company and also contribute to creating a better environment (e.g. having less waste and reducing their CO^2 footprint).

3. What are the main decision-making aspects when deciding between the incorporation of ethics or not in production?

Roland mentions that they have to be sure that the cost will be covered and that a return of investment is the most important aspect.

4. What are the general implementing problems/challenges managers, staff, and/or developers face when implementing ethics/VSD or an ethically justified methodology?

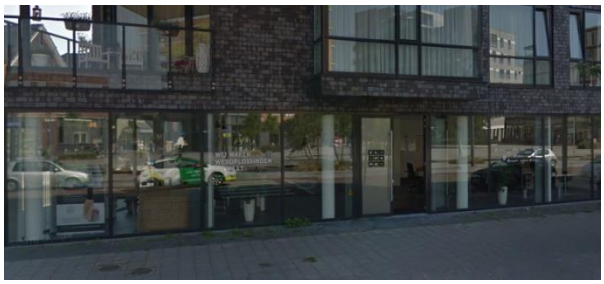
Roland mentions that plans and specifications can contradict the corporate process and the original production system design. For example, when you implement an intervention for safety issues, Roland specifically mentions that you have to be aware the new process could also have a negative outcome (trade-off). An example is protection of the data centre with loud alarms, although the sound raises awareness that something has gone wrong; the frequency of the sound and its vibrations can cause materials to vibrate that can cause the server hard disks to vibrate, which can damage stored data. In this case, it is a business process versus safety trade-off that has to be well evaluated before execution. Another example is that

often usability is important for customers and this is likely to be prioritised before ethics. Customers tend to find new software features more important than the possible ethical implications. Roland mentions that values are often discovered when something goes wrong (and not analysed in the prospective sense).

The scalability of technology and maintenance are also seen as being problematic. Roland mentions that making sure his costs are covered and that a return of investment is important to invest in ethical design.

CASE II

Interview Summary: Michael, CEO El Niño-ICT, Enschede (Software development company, mid-sized company).



1. What are the potential areas of including ethics in your IT company?

Michael, the CEO of El Niño-ICT mentions that good security in his IT company is essential. They have to encrypt their customer hosting data and save passwords accordingly to maintain the privacy of information. Employees cannot leak private data and this has to be ensured by the organisation. Their most important stakeholders are customers and partners and they have to make sure that their knowledge is also not leaked to competitors or any other third parties. Michael mentions when you have a third party hosting company, it is essential to be fair to customers. If this hosting party is has unusual activities and all you care about is the payment, this is obviously also not very ethical. You have to ensure your customers are also safe with third parties.

2. What are the positive and negative effects of incorporating ethics on different IT development economics in your company?

Michael mentioned that being unethical can be easy in the short term, but can lead to losing a lot of happy customers in the long term. In this way, it is better to have some ethical awareness. “We think a long-term investment is more important over a short one in this sense,” Michael says. “For example, if we sell open source, we can resell this for e.g. ten thousands Euros or we can be fair about the invested time. If a customer needs a blog site, we can use an open source system or build the system from the ground up. We are not forcing our customers and we are transparent about their options.”

3. What are the main decision-making aspects when deciding between the incorporation of ethics or choosing none to be adopted in production?

Michel mentioned that long-term relationship building is most important after economic factors.

4. What are the general implementing problems/challenges managers, staff, or developers face when implementing ethics or an ethically justified methodology?

“It is important to train employees to be safe with data and specify categories of knowledge. Knowing what is important for a customer or knowing what issues can be dangerous.”

For the implementation of a VSD methodology, we will look if it is easy to be quickly applied, measurable, practical, and the customer has to see that it is useful. It also depends on the type of software; Michael mentions that an ecommerce system (e.g. online store) is different than a sales software program. Within design, different information is required to look at a per software solution.

“Besides a checklist, a good test would be useful, usage with stakeholders, or a monitoring system that can benefit the IT world.” Michael mentioned the possibility of VSD to be integrated within their Continuous Integration (CI) development method.

“Each time a code is created, you can analyse your code and install plug-ins, and the server will check the code for unsafe functions. It will give errors and denotes which files need to be checked.” Michael mentions that it would be possible to develop multiple VSD systems that would fit within the CI method.

Analysis of the ethical concerns provided by these companies

The questions asked during the interview sketched some practical cases in which companies may be looking for a solution where ethics may help them. In the above cases, the companies also stressed that it is important to intervene correctly. InterDC mentioned the importance of delegating responsibility and that the creation of awareness is important. An important question that was raised is: how do you control your information? Roland mentioned that plans and specifications can contradict the corporate process and the original intended production. Also, the scalability of technology or maintenance is seen as very problematic. Michael of El Niño-ICT mentioned that it is important to train employees to be safe with data.

To sum up the concerns, I have created a schematic overview of the issues that were described by the interviewees.

Gathering daily problems

(1) Gathering ethical knowledge (e.g. to support maintenance)

The companies require a system that they can consult to obtain ethical knowledge. Companies may experience a great benefit when they learn how to incorporate ethical reasoning within their company. A learning system that helps companies may be a great way to introduce ethics within the corporate setting.

(2) Ethical issues should be applied quickly, be measurable, and should be practical.

The ethical interventions should be measurable and also be practical, according to these business cases. Measuring ethics means that a monitoring and reporting system would be essential to measure the effects of ethical interventions within the organisation.

(3) Scaling technology easily

Scaling technology is seen as problematic by Roland. He refers to the intervention that his company made with the alarm system. Therefore, scaling technology the right way when implementing ethics is essential.

As mentioned earlier, Sarah Spiekermann began with the knowledge value, outlining how building information for the machine age requires ethical conduct at all stages of the knowledge creation process (Spiekermann, 2015,p152). I will build further on this emphasis and suggest that knowledge in the form of ethical reasoning should be embedded within a company. Because ethical reasoning can be embedded in knowledge systems and different forms, I will refer to ethical reasoning from now on as “Moral Intelligence”. ‘Intelligence’ has been defined in many different ways, including one's capacity for logic, understanding, self-awareness, learning, emotional knowledge, planning, creativity, and problem solving (Legg & Hutter, 2007).

“Moral” intelligence implicates this same reasoning but within the category of making moral decisions. For my own conceptualisation of Moral Intelligence within a business perspective, Moral Intelligence allows for the organisational or individual capacity to understand right from wrong and to behave (plan, do, check, and act) based on the value that is believed to be right. This is the special part where ethics is required, e.g. during production. I argue that the notion of Moral Intelligence can defined in abstract form as the *theoretical knowledge, justifications, and the actions* that are involved in making ethical decisions. Moral Intelligence can make the same usage of understanding, logic, planning, creativity, and all the forms that are also present within other kinds of intelligence.

For a better understanding, I conceptualise Moral Intelligence as a complete learning system to provide logic and knowledge. Secondly, to promote intelligent planning, the system should incorporate monitoring and reporting. To promote creativity, the system should promote implementation for creating self-awareness and problem solving. This conceptualisation is described below.

Table 3.1

Moral Intelligence conceptualisation		
Intelligence factor	**Promoted function**	***Abstracted system***
-Logic and understanding, emotional understanding	Promotes providing ethical knowledge (e.g. to support maintenance)	*A learning system*
-Planning	Promotes ethical intervention. This should be measurable and practical	*A Monitoring & Reporting system*
-Self-awareness -problem solving.	Promotes the scaling of technology	*An Implementation system*

In an abstracted definition of a Moral Intelligence system, *learning* (e.g. by ethical theory, methods, practice) can be a primary component to promote logic and understanding. *Monitoring and reporting* can promote planning and problem solving will primarily promote implementation.

Having this model raises a further question to address: Who should have Moral Intelligence? I argue that Moral Intelligence can be beneficial for the organisation at large or individuals, such as programmers, designers, or managers to create awareness of the implications of technology. It allows for the incorporation of a system that guides and regulates ethical inquiry, but also creates a general scope to think about. To successfully implement Moral Intelligence, however, I suggest a design and deployment framework in the following chapters that deal with these in a more practical way on how to deploy learning, monitoring, and implementation. Companies may find a great benefit in having access to Moral Intelligence: e.g. companies may steer product design better to gain more quality, reduce negative feedback from their clients, the community, or it can be used to benefit the planet, people, or profit. I argue that, when looking at the big picture, Moral Intelligence is the key to ultimately solve the embedding of an ethical approach in VSD and confronting values more transparently.

3.2 Conclusion and reflection

As already explained during the interviews, it would be useful for CEOs of IT companies to understand how to obtain the right resources for ethical implementation. VSD already promotes the discovery of values during design; consequently, Moral Intelligence can possible be used in VSD to find ethical values and ultimately implement them in technology. Once Moral

Intelligence is cultivated — e.g. by knowing how to use ethical theory and gather ethical data — the next step would be to look further into the adoption of these methods in design processes.

A way to cultivate Moral Intelligence is by incorporating useful theory that can lay the foundations of ethical judgment that will support the processes of learning, monitoring, and implementation with education e.g. by an specialist ethicist. I argue that we need to create the right methods for cultivating Moral Intelligence to create a solid theory that can justify moral actions and form an important compass throughout design. In the next chapter, I will further build on the concept of Moral Intelligence for corporations and how to cultivate it in the next chapter.

CHAPTER 4

Cultivating Moral Intelligence for VSD

Without ethical culture there is no salvation for humanity. [5]

(Albert Einstein)

4.1 Introduction

During Chapters 2 and 3, I explained the relation between values and ethics and several problems companies face in a practical context. Several lines of thought were formulated: How to gather values based on ethical theory? How to incorporate them into a technology context? In Chapter 3, I conceptualised Moral Intelligence with three important core components: learning, monitoring, and reporting that may benefit the industry context.

In this chapter, I question VSD regarding its current methods and I suggest incorporating a Moral Intelligence framework that answers the question:

How to embed ethics in VSD and in production so that ethical values can be operationalised?

I strongly argue that we have to look at the core methods of VSD first prior to incorporating Moral Intelligence. To start developing my method, I will first explain the current components of VSD and how it is originally intended. After this, I design a method for cultivating Moral Intelligence in VSD for different levels of technology.

4.2 The contemporary methods of VSD

A typical VSD project starts with investigating data ("Value Sensitive Design | Public Sphere Project," n.d.). One may start with a Value, Technology, or Context of Use. Any of these three core aspects easily motivates Value Sensitive Design. Begin with aspect most central to your work and interests and continue to identify stakeholders.

Identify Stakeholders

Next, identify *Direct* and *Indirect* stakeholders systematically. Direct stakeholders are those individuals who interact directly with the technology or with the technology's output; indirect stakeholders are also impacted by the system, though they never interact directly with it.

Identify Harm and Benefits

Following up and identify possible Harms and Benefits for each stakeholder group. Systematically identify how each category of direct and indirect stakeholders would be positively or negatively affected by the technology under consideration.

- Map Harms and Benefits on to Corresponding Values. At times, the mapping between harms and benefits and corresponding values will be one of identity; at other times, it will be multifaceted (that is, a single harm might implicate multiple values, such as both security and autonomy).

Conceptual Analysis

- Conduct a Conceptual Investigation of Key Values. Develop careful working definitions for each of the key values. Drawing on the philosophical literature can be helpful here.

- Identify Potential Value Conflicts. For the purposes of design, value conflicts usually should not be conceived of as "either/or" situations, but as constraints on the design space. Typical value conflicts include accountability vs. privacy, trust vs. security, environmental sustainability vs. economic development, privacy vs. security, and hierarchical control vs. democratisation.

Technical Investigation

- Technical Investigation Heuristic Value Conflicts. Technical mechanisms will often adjudicate multiple if not conflicting values, often in the form of design trade-offs. It may be helpful to make explicit how a design trade-off maps on to a value conflict and differentially affects different groups of stakeholders.

- Technical Investigation of Unanticipated Heuristic Consequences and Value Conflicts. In order to be positioned to respond agilely to unanticipated consequences and value conflicts, when possible,

[5] *"The Need for Ethical Culture" celebrating the seventy-fifth anniversary of the Ethical Culture Society, founded by Felix Adler (5 January 1951)*

design flexibility into the underlying technical architecture is necessary to support post-deployment modifications.

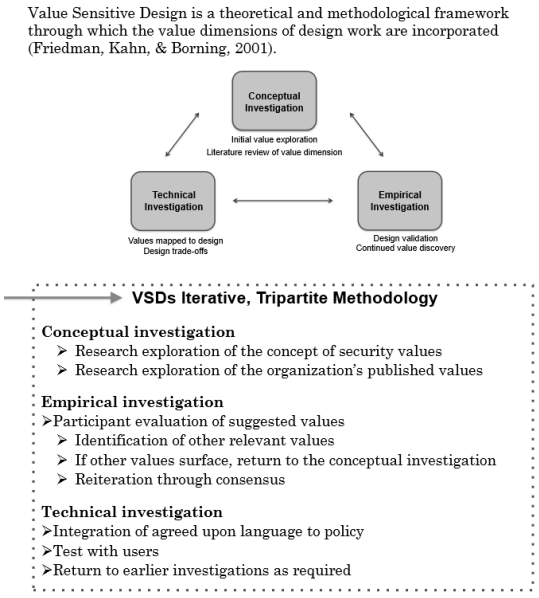


Figure 4.1: VSD

4.3 Analysis of the tripartite methodology of VSD

The above overview shows how values can be gathered. Unfortunately, I have several concerns regarding the conceptual, empirical and technical investigations. It is unclear how values are gathered by stakeholders as I will point out. How do you values and on the basis of what analysis? Is there a standardisation in this? Which pragmatic steps and actions need to be taken? Critical questions and actions are stated below regarding these investigation methods.

Reflection on conceptual analysis, as described earlier

- *Conduct a Conceptual Investigation of Key Values.*
  - *Develop careful working definitions for each of the key values. Drawing on the philosophical literature can be helpful here.*

My critique: for anyone performing conceptual analysis is to note it is crucial to know where and how to get the information on the topic of values. Questions that need to be explicated in the conceptual phase are: what are methods of gathering information? Is, for example, literature the only way? Or are there other useful methods that can be used?

VSD has incorporated the investigation of values into a range of standard social science methods, such as semi-structured interviews, surveys, observations, quasi-experimental designs, exploratory inquiries, and longitudinal case studies (Davis & Nathan, 2013).

Ten descriptions of investigation methods can be found in VSD literature:

(1) stakeholder analysis; (2) designer/stakeholder explicitly supported values; (3) co-evolution of technology and social structure; (4) value scenarios; (5) value sketches; (6) value-oriented semi structured interview; (7) granular assessments of magnitude, scale, and proximity;(8) value-oriented coding manual; (9) value-oriented mock-ups, prototypes, and field deployments; (10) ethnography focused on values and technology; (11) model for informed consent online; (12) value dams and flows; (13) value sensitive action reflection model; and (14) envisioning cards

Authors of the paper "Value Sensitive Design: Applications, Adaptations, and Critiques" were still unable to find pragmatic descriptions of all of these in the published literature on VSD (Davis & Nathan, 2013,p17).

As a result they only discuss three methods from this list that are reported upon in their literature and I argue that methods like these should be presented more clearly and step-by-step (instead of being fragmented) to have more people working with it. Some methods may also be not practical whereas others are.

Suggestions: Methods should also be easily accessed and not fragmented throughout various literatures. A great effort is already the combined work of various authors in "Handbook of Ethics, Values, and Technological Design" that discusses Value incorporation in Design on different domains. This handbook makes available work in ethics of technology and design research that provides an altogether different and more constructive perspective on the possibility to develop technology in accordance with the moral values of users and society at large (van den Hoven et al., 2015,p1). I suggest a step-by-step VSD handbook with each method in detail explained with illustrative cases to attract more VSD participants and to clarify the process of the conceptual analysis. It would be impossible during this thesis to make all steps explicit but it would be an interesting idea to further develop a platform that allows for summarising and describing these methods by the VSD community.

Suggested action 1: I would suggest making *VSD methods* more explicit and providing a platform where all methods can come together.

- *Identify Potential Value Conflicts.*
  - *For the purposes of design, value conflicts should usually not be conceived of as "either/or" situations, but as constraints on the design space. Typical value conflicts include accountability vs. privacy, trust vs. security, environmental sustainability vs. economic development, privacy vs. security, and hierarchical control vs. democratisation.*

My critique: In this phase, potential Value Conflicts are gathered but no explicit reference is made towards an ethical framework. We require a conceptual framework that can gather motivations or value conflicts and providing ethical reasoning. Literature also include this critique, VSD seems to lack a fundamental discussion of ethical theory. Even though VSD draws

on ethical theory and takes universal values into account it is not exactly clear what theories and which values this includes (Albrechtslund, 2007,p67). Authors of the paper "Value Sensitive Design: Applications, Adaptations, and Critiques" conclude descriptions of VSD do not recommend the use of any particular ethical theory (Davis & Nathan, 2013). Manders-Huits advances this critique as well, arguing that the notion of values in VSD is "underdeveloped." She claims that VSD provides "no methodological account for distinguishing genuine moral values from mere preferences, wishes, and whims of those involved in the design process". She argues that VSD requires a complementary ethical theory not only to demarcate moral values but also to provide a basis on which to make principled judgments about which values are most important to support (Manders-Huits, 2011).

Another more general critique concerning the values I have is that VSD literature does not investigate or connect the role of the organisation its values that determine most of the production its outcome. A general assumption is that an engineer's principal concern is to make a technology which has the required functionality (Manders-Huits, 2011,p3) but what about the role and influence of the organisation that the engineer works for?

Most critiques on incorporating VSD do not open this discussion, however I argue it is important to connect organisation values with the designers their values. In order to improve the engineer his or her moral decisions I argue that the organisation should be transformed in the first place because engineers are often limited to their corporate resources and may be heavily influenced by corporate culture. Design for Value deals with a lot interesting design methods for all sort of domains, yet almost none discuss the role of the organisation or makes a real recommendation/ connection with VSD on the organisational level. This means that I advocate that the scope of incorporating Design of Values is a lot broader than the role of solely the engineer, the designer, the ethicists or the design team. Organisational contingencies influence the design space and the design space influences the team and individuals.

Suggestions: I suggest further developing the concept of Moral Intelligence as an ethical theory that VSD can use alongside its iterative approach. Moreover, I will connect organisational contingencies with the incorporation of VSD to go beyond the 'standard' role of the engineer and to make 'design space' more specific.

Suggested action 2: I suggest combining *ethical and organisational theory* with VSD to support value incorporation within the IT industry.

Technical Investigation

- *Technical Investigation Heuristic Value Conflicts.*
  - *Technical mechanisms will often adjudicate multiple if not conflicting values, often in the form of design trade-offs. It may be helpful to make explicit how a design trade-off maps on to a value conflict and differentially affects different groups of stakeholders.*

My critique: the wording 'technical mechanisms' is very ambiguous within the technical investigation. VSD does not conceptualise or define technology components clearly. While it states technology should be made explicit for design, I suggest to scrutinise technology in a more pragmatic way, defining important components of the technology to be monitored and evaluated during the process of development. In other words, we require a method to see how technology is built from the ground up in very detailed level (e.g. by scrutiny of the design methodology) and making the design and technology processes specific, only then it would be clear to indicate how a design trade off maps on to a value conflict and how to propose and incorporation of VSD within technology.

Suggested action 3: I suggest developing a framework to make *technology* more specific and to place *values* within this framework.

- *Technical Investigation Heuristic Unanticipated Consequences and Value Conflicts.*
  - *In order to be positioned to respond agilely to unanticipated consequences and value conflicts, when possible, design flexibility into the underlying technical architecture to support post-deployment modifications.*

My critique on current VSD methods: In the above technical investigation, it is stated that we require "design flexibility" to be built into the underlying architecture, including support for post-deployment modifications. But, what is the architecture that technology engineering deals with? How can this architecture be modified to be flexible and ready for post-deployment modifications? We may need to look into the practices of engineering to support post-deployment modifications e.g. in a recent interview Jeroen van der Hoven mentions: "One important area of research could be to design and evaluate agile methodology and methods for value-sensitive design" (Maedche, 2017, p300).

Suggestions: to incorporate technical investigations we may need to open up and visualise technical architecture and design flexibility into design practices. This means scrutinising development methods, the engineering processes, tooling and to see post-deployment modification.

Suggested action 4: I suggest providing a novel framework that integrates *agile methods* and that visualizes architecture, making it possible to work on ethical choices (see action 3) and going more deeply into characteristic of design.

4.4 Cultivating Moral Intelligence with technology and methods

From the above paragraph and suggestions, it became clear that we require a framework that can gather motivations/value conflicts that provide ethical reasoning and lead designs to suit post-deployment modifications.

Before any VSD intervention can take place, I argue that a new component has to be introduced during the conceptual, technical, and empirical investigations. I specifically argue that for solving the philosophical problems, we require an ethical capture component for learning, monitoring, and reporting values during design, which I earlier introduced as Moral Intelligence. Moral Intelligence offers a way to capture value conflicts, a way to monitor changes and to report, which makes it easier to integrate the requirements during development. If learning, monitoring, and reporting systems are implemented within design, this will allow developers to be transparent and flexible after software deployment. The moral component can be of use for VSD, but it can also be of use for a standalone theory that can contribute to VSD. Because VSD has its own methods, the suggested theories I propose can be used within the practice of VSD. However, I do not want to dictate how VSD should be practiced, as it is a continuously evolving practice.

4.5 Operationalisation of an ethical capturing component

Without any ethical reference, theory, or norms, we cannot build ethical intelligence. We require additional standalone theories to extend our implementation of adding Moral Intelligence that will lead us ultimately to a way of capturing values and extend it to a more pragmatic and practical approach.

Now that VSD methods are described, I have chosen finding a theoretical basis in the theory of technology ethics and emerging technologies. `

Appendix K provides an introduction to contemporary ethical theory, which may be used for ethical discussions, however these may not always be suitable for practical every day usage as further explained in that section.

Emerging technologies are new technologies that are currently developing or will be developed over the next five to ten years, and which will substantially alter the business and social environment. These include information technology, wireless data communication, man-machine communication, on-demand printing, bio-technologies, and advanced robotics. In the theory of emerging technologies, there are several ethical frameworks that can help to assess technologies such as IT. Technology ethics can be used to form a basis for looking at the ethical role of technology in a greater sense.

The Anticipatory Technology Ethics (ATE) framework may be well suited to be extended to VSD as well. In order to do so, this section focuses on two distinct theories, namely VSD (design) and ATE (assessment) and both methods will be scrutinised intensively at the core of their methodology and methods to find integration possibilities. After introducing ATE, the link with VSD will be developed and I will argue how both approaches can benefit from each other to form a strong foundation to be ethically integrated within conceptual, technical, and empirical investigations.

This part of the thesis is structured as follows. In the first section, three contemporary approaches of technology assessment will be distinguished. The aim in the first section of this chapter is to (1) explicate and critically discuss ethical approaches that may be for assessing technologies, the methods that are used and their presuppositions. The next section (2) deals with comparing and contrasting different approaches, which forms the basis for Anticipatory Ethics. The final part (3) will consider how to further develop the concepts or theories to incorporate Moral Intelligence in VSD.

4.6 Towards anticipation of technology assessment and methods of ethical analysis

An important argument mentioned by Roland of InterDC was that technology scaling is required. Without a clear vision of how technology is developed, Moral Intelligence would be hard to integrate. Luckily, there are several abstractions of technology that help us to identify some core components. The ethics of emerging technology is the study of ethical issues at the R&D and introduction stage of technological development through the possible anticipation of future devices, applications, and social consequences. Essentially, the Ethical assessment of Emerging Technologies tries to answer: What is good and bad about devices and processes? What is right or wrong in how they are used? An ethicist may not lose him/herself in idle speculation of future ethical issues. However, they must not forget to say anything about the future because they do not know which devices and uses may result from them.

Brey discusses three contemporary approaches to ethics of emerging technologies that use forecasting: (1) Ethical Technology Assessment (ETA), (2) Techno-Ethical Scenarios, and (3) the ETICA approach (P. A. E. Brey, 2012). He later on presents his own approach, namely ETA, a conceptual and methodological rich approach that incorporates a large variety of ethical principles, issues, objects, levels of analysis, and research aims. I argue that this ethical reasoning based on the assessment of emerging technologies may be used to provide an ethical framework within the Conceptual, Technical and Empirical investigations of VSD to better capture values. These (1)(2)(3) methods are introduced on the next page.

4.6.1 Ethical Technology Assessment

Palm and Hansson (2006) provide indicators of negative ethical implications at an early stage of technological development. In many cases, this can be used to guide design or policy. A good highlight of this assessment is that it focuses on the whole life cycle of technology development (from initial R&D to impacts on society) and a close interaction with developers of technology. To attain an adequate understanding of future developments, ETA relies on technology assessment and on close **interactions**. ETA does not aim to look far in the future but rather continuously (throughout all stages of development) and provides feedback to designers and policy makers. It does so by,

(1) Confronting projected features of the technology or projected social consequences with ethical concepts and principles.

(2) This ethical knowledge may then be used to adjust design processes so as to avoid ethical concerns or to steer decision-making on an emerging technology.

(3) Evaluation using an ethical checklist (also see Appendix G). Brey, (2012) mentions it's rather vague in methodology (not what knowledge is required, but how it is best) and the ethical checklist seems limited so he suggests and extended list: no moral values on the list, such as human dignity, informed consent, and autonomy (Palm & Hansson, 2006).

Overview of pros and cons

In short, the pros are (1) ETA focuses on the whole life cycle of technology development (from initial R&D to impacts on society). (2) Close interactions with developers of technology; e.g. technology questions and social consequences to organise interactions with technology developers. (3) Feedback to designers. Critiques are about its rather vague methodology, the ethical checklist seems limited, and, as mentioned, no moral values are on the list.

4.6.2 Techno Ethical Scenario Approach

A second approach is the Techno Ethical Scenario Approach, mostly based on sketching the moral landscape based on current moral beliefs, practices, and regulations. At its core, it aims to identify ethical issues and arguments in a new technology's promises and expectations in order to help policy makers to anticipate ethical controversies related to an emerging technology (Boenink, Swierstra, & Stemerding, 2010).

A main focus of this approach is to anticipate the mutual interaction between technology and morality and the changes in morality that may result from this interaction.

It does so by finding a chain of arguments and counter-arguments regarding positive and negative aspects of the technology with three steps (1) "sketching the moral landscape". In this phase, it aims to describe the *moral beliefs, practices, and regulations* that are directly or indirectly relevant to the technology. (2) Generating potential moral controversies, using NEST-Ethics[6], which function here to identify promises and expectations concerning a new technology but also raises critical arguments against these promises. This can include rights, harms, the good life, justice. The different steps may include literature reviews of technologies, workshops with policy makers and Technology Assessment (TA) experts. (3) The third step in the techno-analytical scenario approach is constructing closure by judging plausibility of resolutions.

In this step, the multitude views and their arguments are reduced by imagining which resolution of the debate is the most plausible. Brey praises the technical ethical scenarios approach for incorporating a larger time frame and identifying complex patterns of argumentation regarding ethical issues (Boenink, Swierstra, & Stemerding, 2010,p7) than ETA — which focus more on incremental steps — but criticises its descriptive and predictive approach instead of being a normative and prescriptive one. In techno-ethical scenarios, important moral controversies may remain hidden because of the complexity or opaqueness of technological artefacts or practices, e.g. debates that have not been taken into account (P. Brey, 2012).

Overview of pros and cons

The pros are that it offers a larger time frame than ETA (which focuses more on incremental steps). The cons, however, are that it does not provide a normative and prescriptive account for how the issues are resolved. Also, in techno-ethical scenarios, moral controversies may remain hidden.

4.6.3 ETICA

Last but not least, Brey (2012) discusses (a general approach for the ethical assessment of emerging information and communication technologies (ETICA). It gives a general approach for ICTs, but because it is so general, it can be also be used for the assessment for other technologies too. Its aims are a foresight analysis especially for the medium term future (forecasts multiple possible futures) and an overview of ethical issues.

The primary stages are (1) the identification stage particular applications, artefacts, and properties. Most ethical values and principles used in this approach are derived from a prior list of ethical issues, especially for ethical evaluation in a European context. These resulting ethical issues are summarised in a matrix,

[6] A full NEST ethics explanation and an introduction to contemporary ethics is covered in Appendix J

which specifies relative normative issues in relation to particular emerging technologies, artefacts, and applications that resulted from them expectantly. Sources include governmental and political sources, scientific text, non-academic text, or scientific methods. (2) The second stage of ethical analysis is the evaluation stage that ranks ethical issues from the identification in relation to each other. In the final third stage, the government stage, recommendations are developed for policy makers to deal with ethical issues described in the earlier stages (P. Brey, 2012).

The aggregated approach is considered better compared to an individual assessment. It is often referred to as studying the generic ***properties*** of technologies. Therefore, the range of artefacts and implications considered are often limited. For example, robotics is a large field but only a few types of robots and ***applications*** of robots are considered.

Overview of pros and cons

Pros are that ETICA aims at thoroughness by considering a wide range of technological properties, artefacts, applications, and ethical issues. Cons are aggregating versus individual assessment. Often it refers to generic properties of studied technologies and a limited range of artefacts and implications are considered.

Analysis

In the previous section, we discussed contemporary analysis of emerging technology that forms a basis for Anticipatory Technology Ethics. However, the ultimate goal in this thesis is to bring Moral Intelligence into VSD. The next part forms an analysis on how to achieve this and what the core problems of VSD are to build upon.

4.6.4 Anticipatory Technology Ethics

In response to the earlier discussed contemporary approaches, Brey (2012) suggests an ethical approach of his own which that he formulates as ATE that includes three levels of ethical analysis, namely technology, architectural, and an application level (P. Brey, 2012). At each of these levels, various objects of ethical analysis are defined: *things, properties, or processes* that raise ethical issues. The three levels are similar to those of ETICA, defining features of technology, artefacts and applications.

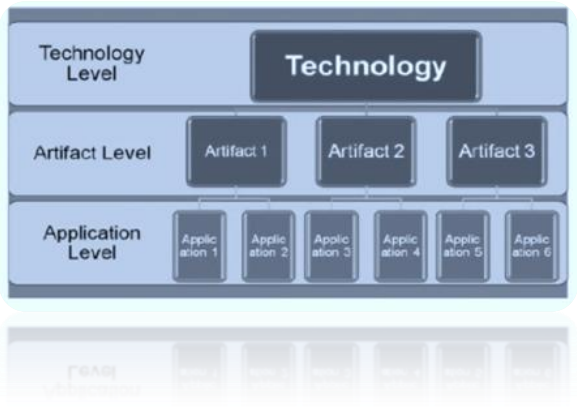


Figure 4.2: The levels of ethical analysis. Source: (P. A. E. Brey, 2012)

(1) The technology level is the level at which a technology is defined. A technology is a collection of techniques that are related to due to a common domain, purpose, formal, or functional features. In nano technology, it is possible to distinguish optical-nano technology and DNA-nano technology. In the level of ethical analysis, the focus is especially on the *general features* of the technology (P. Brey, 2012).

(2) The artefact defines the physical configuration that produces a desired result when operated. A *procedure* is a sequence of actions that produces a desired result when performed using the *right tools*. Technological artefacts and procedures are derived from that which technology produces. At the artefact level, ethical analysis focuses on types of artefacts and processes that result from a particular technology (P. Brey, 2012).

(3) At the application level, ethical analysis focuses on particular ways of using an artefact or procedure which is often the configuration or the context of its use. Ethical issues that may be involved at the application level are moral issues that could be related to the *intended use* of the artefact (P. Brey, 2012). Brey proposed a **forecasting method** specifically within these three levels of assessment.

First, at the (1) technology level, knowledge will be acquired from engineers because they are best positioned to understand and describe the features, techniques, and the subclasses of the technology. They are positioned to inform ethicists. This level may also require consultation of other experts from other fields of studies.

Within the (2) artefact level, analysis should utilise existing studies in forecasting and technology assessment (TA) about the technology, to the extent that these are available.

Within the (3) application level, TA experts should perform analysis with a view of artefacts and applications that are most likely to emerge in the future by consulting engineers, technology forecasters, and TA experts but also historians, sociologists and marketing experts. Because of its imaginative activity in the methods identification stage, it may also be useful

to consider policy documents, company studies, academic texts or even science fiction stories for ideas about possible future artefacts and applications, as long as these ideas are then subjected to scrutiny regarding their feasibility and plausibility. This particular interest may imply that ethicists will sometimes have to develop their own forecasts and scenarios that focus on such matters.

Methods of ethical analysis

Two methods of ethical analysis can be distinguished for ATE:
Identification: operationalising moral values and principles are cross-referenced with technology descriptions resulting from the forecasting stage. Negative impacts are also investigated through inputs from the ethical checklist, technology ethics literature, and a bottom-up analysis.

Evaluation: ethical issues are evaluated and elaborated, mainly to improve technology/guide development. Brey (2012) refers to the design feedback stage, which is also prominent in Palm and Hansson's ETA. Methods like **VSD** can be used to help implement the results of ethical evaluation into design processes, as suggested in the ATE approach. Brey (2012) does not go into further detail on this in his paper, although he hints that VSD and ATE could be benefit from each other. "Admittedly, some parts of the approach are still sketchy. It is my hope that a further development of ATE may advance the field of technology ethics and may afford more ethical development and governance of technology" (P. Brey, 2012).

In response to this part, I will explain how the previous explained theory will relate to VSD and its methodology and where ATE can help to engage with VSD and vice versa. I will do so by examining VSD and ATE methods.

4.6.5 Combining all methods to cultivate Moral Intelligence

In this section, I develop a more standardised methodology for VSD that incorporates the strong points of Anticipatory Ethics and the previously explained contemporary methods. First of all, Brey (2012) introduced three levels of analysis, as explained earlier and suggest the *classification of technology, analysis of objects, and procedures analysis.* I propose a schematic overview of the analysis and its abstraction of how technology (architecture) is built.

Anticipatory Technology Ethics (ATE) form an important framework in which all technology ethics and emerging technologies theories can be used and extended iteratively in the future.

- ATE explicates (1) technological properties, (2) artefacts, (3) applications.
- ATE describes a model of technology which can be taken apart and explicated towards analysing methods.
- ATE can be used as a framework to explicate and categorise ethical methods within levels of technology.

Diving again into classes, methods, and procedures

Table 4.1 Abstraction of technology architecture

Class	Object of analysis	Procedures analysis
Technology and its future application	Methods, e.g. Systems as objects	Variables, configurations

On the basis of a technology class, functional artefacts (objects), like systems and procedures, are developed. Below, I present ATE by means of its *abstracted model*. The technology can be defined as a class that requires assessment, such as its general purpose, for example nano-technology (defined in a class). Second is the artefact level and derived artefacts, also called objects. Third are the applications, produced systems, and variables/configurations. The last one indicates the context of usage, as explained by Brey (2012).

Anticipatory Ethics

Table 4.2 ATE Levels of Ethical Analysis

Class	Object Analysis	Procedures
Technology Level (defining the technology)	Artefact Level (system)	Application Level (context of use)

VSD and Anticipatory Integrated Approach

As explained earlier, VSD focuses on a tripartite methodology of *Technical, Empirical and Conceptual*. Below is an integrated approach using the best methods described within the three levels of ethical analysis. Within the object analysis, artefacts are analysed through conceptual, empirical, and technical methodologies, and the below blueprint is explained in the next section. The methods explained earlier have been inserted into the levels of technology creation.

Table 4.3 VSD and Anticipatory Technology Ethics Methods Integrated into a learning component

Class	*Object Analysis*	*Procedures Analysis*
Technology creation level	Empirical, Technical, Conceptual (ETC)	Empirical, Technical, Conceptual (ETC)
How?	**How?**	**How?**
Engineers consult ethicist and vice versa for forecasting. - ethical analysis focuses on features of the technology at large, particular subclasses of it, or techniques within it.	*-Extended ethical checklist -NEST ethics - scenario sketching, define larger scopes. -Expert survey -Utilise existing studies in forecasting and TA about the technology, to the extent that these are available*	*-Future assessment e.g. with workshops, evaluation, consulting TA experts historians, sociologists, and marketing experts, policy documents, company studies, academic texts. -Analyse application and system variables*
Ethical analysis focuses on	**Ethical analysis focuses on**	**Ethical analysis focuses on**
Features of the technology at large (the defining feature of the technology)	Types of artefacts and processes that result from the technology	Concrete use of the application

4.7 A value capturing model that promotes Moral Intelligence

The previous scheme may provide a blueprint on how VSD could gather Moral Intelligence across a broad domain. On the technology creation level, engineers consult ethicists, and vice versa. On the object analysis level, analysis can be more focused towards building components that make up the technology, such as a checklist approach, NEST ethics, or an expert survey. If someone wants more details of actual practices and the application context, workshops may be used with specific consultants on various domains because they can provide in-depth knowledge through workshops, evaluations, consulting TA experts and historians, sociologists, and marketing experts, policy documents, company studies, and academic texts. Making sure that the correct system variables are used, such as during product development, is also part of this analysis level.

Of course, there may be many more methods to gain and capture insights into values, for example the envisioning cards methods (Section 6.4.1).

How would VSD benefit from anticipatory ethics and vice versa?

I suggest that it's better to develop a methodology that embeds moral and ethical intelligence (instead of just conversations) into technological designs, company structures, and technology assessments. In the analysis, I have proposed that the best practices and methods of contemporary ethical analysis and ATE could fit within VSD to make a more standardised approach to capture values, which I present as a VSD capture framework for the technology at hand.

ATE provides a good model of doing so because the three levels of analysis are integrated together in Table 3 with each of the tripartite methodology of VSD. Each level of analysis, such as classes, objects and procedures will require the technical, empirical and conceptual investigation. Methods like the extended checklist or NEST ethics could bring intelligence into VSD within the object of analysis. Also, scenario sketching and moral beliefs would create a vision through which objects should be investigated in a future stage. Within the technical investigations, for example, engineers can be consulted by ethicists, focusing on how existing technological properties and underlying mechanisms support or hinder human values.

For the object analysis, there is room for methods like the extended ethical checklist (also see Appendix G), NEST ethics, and scenario sketching. Expert surveys may be used for the empirical, technical, and conceptual investigations. For the procedure analysis, workshops, evaluation, consulting TA experts, historians, sociologists and marketing experts, policy documents, company studies, and academic texts could be used. VSD does not state that this can't be done, but this approach includes ethical grounded theory in VSD such as NEST ethics and an ethical checklist. Capturing this intelligence in a sophisticated, yet pragmatic manner and translating these moral issues to working requirements will be further developed in the next chapter and, in the final chapter, tools are developed to implement these methods.

Using Moral Intelligence in practice

In the Chapter 2, I introduced two business cases and emphasised that technology scaling for companies is very important. El Niño-ICT emphasised that they have a commitment to build long-term relationships and investments. Generalising the company's intentions and especially IT companies, it is of no doubt those companies require technology scaling if they emphasise long-term investments. Building new technology and extending their features has become a daily operation for these IT companies. If the core focus is ultimately on technology scaling (e.g. supporting more business processes and customers) then exact know-how is required in the long run for how to specifically design technology so that is configured optimally. The ATE framework works well to define the technology in an abstract model for these companies on different levels to bring focus on their developments.

The Data Warehouse, InterDC can define corporate strategies on three different levels. One metric of ethical innovation can be the increase of quality due to product improvements. The Moral Intelligence framework is proposed three different levels. The Moral Intelligence framework can function as a pragmatic scope analysis on the levels of technology.

Classes of Technology analysis for InterDC

We focus here on current technologies. This can be new innovation in data techniques, storage, transmitting, or even quantum computers to compute and store data in innovative ways. For InterDC, learning to think in a greater sense about these techniques can

benefit in creating innovative technology in the long run. The ethical analysis can function here as a check on these developments, e.g. which directions will be perhaps acceptable for public? Will more computer power be dangerous for the public, e.g. by increasing hacking capabilities? This greater scope helps InterDC to research the current impact of technology.

Supporting the object analysis

Within object analysis, new objects will be discovered that are often created by tooling. At this level, tooling can be scrutinised, such as the components that are important to support hosting. For example, tooling (the object) can apply to the server's hardware. The quality of the server can be measured (experience of problems, safety, information) with surveys. Creative scenarios can also be developed: what if we replace this server with something else? An ethical checklist may help to raise questions about risks and benefits, specific rights for end-users, privacy, property (e.g. intellectual property), justice, and so on. NEST ethics may function here as a check on the general debate. What trends concern the general public regarding storing data at data warehouses? By incorporating trends of moral debate, companies may have a better picture of moral reasons that implicate positive investments.

Supporting procedures analysis

On this level, concrete usage is well analysed. On the procedure levels, application variables can be made specific (e.g. which protocols, bug reporting) When scrutinising application usage (e.g. the application and system variables), often business processes (procedures) are assessed. With the support of workshops, experts, company studies, and consultants, new ways may be revealed to include ethical analysis within system variables. This could involve meet-ups with data ethicists, end-users, web designers, software engineers, and the infrastructure provider (InterDC) may stimulate engagement by creating a solution to satisfy all needs in line with important values. Because employees have different tasks within a company, sometimes scenarios must be focussed on and workshops may suit quick results. Resources can be made explicit, such as consulting policy makers, marketing experts, to allow us to morally think about the concrete use of an application.

Extending the Moral Intelligence Framework

I have discussed several ways of cultivating Moral Intelligence by learning and the theoretical background of technology assessment. Following up, in-depth research is required in the form of case studies to gather more supporting empirical data. Most CEOs are busy with normal daily operations, so the Moral Intelligence model provides not only a scope but also an overview and creative opportunities by unveiling new, interesting methods to gather empirical data.

Table 4.4 Moral Intelligence conceptualisation

Moral Intelligence conceptualisation			
Intelligence factor	Promoted function	*Abstract Moral Intelligence system*	*How to?*
-Logic and understanding, -emotional understanding -Self-awareness	Promotes providing ethical knowledge (e.g. to support maintenance)	*(1)Learning*	*Make use of learning components described in this chapter.*
-Planning	E.g. Promotes ethical intervention. Should measurable and practical,	*(2)Monitoring & Reporting*	*Discussed in upcoming chapters*
-Problem solving.	E.g. Promotes the scaling of technology	*(3)Implement-ation*	*Discussed in upcoming chapters*

Supporting creative methods in an iterative manner may also be suitable or Moral Intelligence. Another innovation lies in this framework that can be easily extended with additional methods, such as from other ontological values. If one seeks to gain values in a more creative way, the Foresight Diamond method can be used.

The Foresight Diamond is a method to gather qualitative and/or quantitative data based on a set of methods positioned in creative, expertise, interactive, and evidence thinking.

Creativity-based methods normally require a mixture of original and imaginative thinking, often provided by technology "gurus", via genius forecasting, backcasting, or essays.

Expertise-based methods rely on the skill and knowledge of individuals in a particular area or subject.

Interaction-based expertise often gains considerably from being brought together and challenged to articulate with other areas of expertise. This means, "bottom-up", participatory and inclusive activities, not just reliance on evidence and experts (which are liable to be used selectively!).

Evidence-based methods attempt to explain and/or forecast a particular phenomenon with the support of reliable documentation and means of analysis.

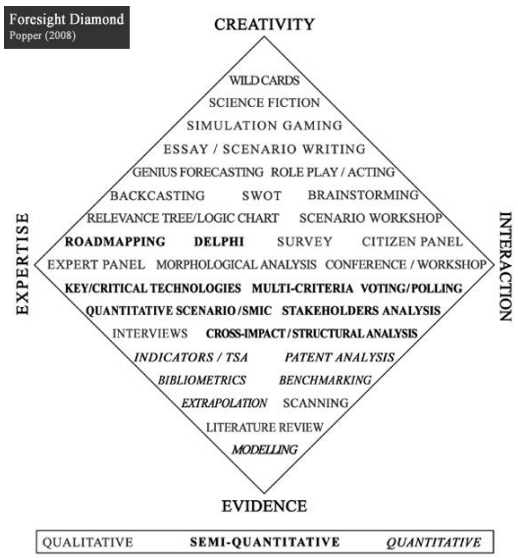


Figure 4.5: The Foresight Diamond, All these methods may be used to raise questions or capture important values. (Source: Popper, 2008)

4.8 Conclusion and Reflection

How does this chapter provide a solution for VSD? Chapter 4 provided a basis to further investigate technology assessment and how ethical theory could fit within Moral Intelligence. The Table 5 overview showcases how several methods could be useful for VSD. This framework may be extended by other methods as well and is not limited to the suggestions provided. Building blocks of technology on macro and micro levels were specified to emphasise different domains of technology. ATE brings in the notion of different levels of technology assessment and I proposed how to use these building blocks within technology design, first on the abstract level and, in the upcoming chapters, within organisational dimensions, tooling, and practice.

The aim of this chapter was to bring Moral Intelligence into the tripartite methodology of VSD. These methods form the basis to define a “VSD capture method” to gather information and useful empirical data that is strongly linked to ethical reasoning. A blue print was proposed as a framework that can guide the development of practical tooling.

Anticipatory ethics opened interesting possibilities to dive deeper into R&D and to ground ethical theory into VSD—the suggested levels of analysis can form a pragmatic approach by specifying different classes, methods, and procedures of technology, which ultimately allow for different roles in research.

The suggested blueprint does not account for the integration and eventual incorporation of VSD in the corporate setting and its strategy (how to?), as I have argued that (technology) architecture such as tooling, production methodology, and company structure may account for the ultimate success of its operation. The willingness to adopt these methods, its mitigation between hostile environments, or the rejection of ethical concerns to speed up the project also account for true success.

The next chapter deals with this corporate world and its production context to examine these contingencies and factors of Moral Intelligence to find a hotspot for practical ethical embedment and integration into design procedures.

CHAPTER 5

VSD and Design Methodology

Perfection is not attainable, but if we chase perfection we can catch excellence.

Vince Lombardi[7]

5.1 Introduction

In this chapter, the framework of Chapter 4 will be put into a more practical engineering context. It is very interesting that the VSD community raised a lot of questions regarding VSD practice, but lacked the precise scrutiny of VSD implementations within the mechanics and values of tech-companies.

If the design context is better understood then the methods of VSD can be linked better into design to solve 1.1-1.4, as introduced in Chapter 2. From there, Moral Intelligence can also be a fundamental part of the development/design process if incorporated correctly.

The main question in this chapter is: if the design context is so important then what is the design context? And, how can people engage within the design context to incorporate VSD and Moral Intelligence?

5.2 Embedding VSD in production methods

We start with the first two operational problems as formulated in Chapter 2.

(1.1) The difficulty to properly explicate and translate ethical considerations to workable requirements and specifications for other project participants actually building the system.

The above problem provides us with the insight that translating ethical considerations to workable requirements and specifications by a team is considered as problematic. I argue that it is necessary to look into what "workable" actually means and also the process of capturing specifications. What is really required for capturing the values and turning them into well-designed solutions that work?

For this reason, the problem to operationalise values (1.2) so they can integrate into technological design may overlap with problem 1.1. The difference is that 1.1 defines the ethical issues that may be required and the second illustrates the problems that may be at stake while implementing and translating the issues from ethical values to technical features.

First of all, "workable" means having the freedom to choose different approaches. Therefore, the next step for incorporating VSD in the IT industry is to find out what the current practices are within the development/production of IT software.
For this, I propose a design analysis dimension that unifies the empirical, technical, and conceptual investigations. The fourth dimension is very important, the design analysis, which should unify the empirical, technical and conceptual analyses in terms of specific (organisational) processes, production, and design to eventually steer the integration/implementation of VSD.

Until now, the existing dimensions (conceptual, technical, and empirical) fail to create pragmatic steps—the technology dimension adds a technological scope; the empirical adds empirical knowledge, and the conceptual investigation simply adds conceptual knowledge and resources.

One critical question remains: Why is there no design analysis dimension that captures organisational dimensions, such as design practices, competences, and production methodology? In order for VSD to successfully extend into other industry domains, we also require many practical considerations, which are predominantly design related. A design analysis will simply stimulate the incorporation of the other three dimensions into technology creation and clarify what design actually is or should be at the project level.

Table 5.1 the investigation methods need to come together in the design analysis

Class →	*Object Analysis →*	*Procedures Analysis →*	**Design Process Analysis**
Technology creation level(ETC)	Empirical, Technical, Conceptual (ETC)	Empirical, Technical, Conceptual (ETC)	(Practical) Design analysis
How?	**How?**	**How?**	**How?**
Engineers consult ethicist and vice versa for forecasting - ethical analysis focuses on features of the technology at large, particular subclasses of it, or techniques within it.	*-Extended ethical checklist -NEST ethics - scenario sketching, define larger scopes. -Expert survey -Utilise existing studies in forecasting and TA about the technology, to the extent that these are available*	*-Future assessment e.g. with workshops, evaluation, consulting TA experts historians, sociologists, and marketing experts, policy documents, company studies, academic texts. -Analyse application and system variables*	*-Integrate ETC in production methodologies and tooling -Link ETC to process and design methods -Analyse by using the 7s model (style, system, structure skill, etc.)*
Ethical analysis focuses on	**Ethical analysis focuses on**	**Ethical analysis focuses on**	**Ethical analysis focuses on**
Features of the technology at large (the defining feature of the technology)	Types of artefacts and processes that result from the technology	Concrete use of the application	The (to be)designed processes and used production methodologies

[7] http://www.vincelombardi.com/quotes.html

Moreover, by incorporating a design analysis in the VSD methodology, we ensure that design procedures are taken into account as well. The empirical, technological, and conceptual investigations need to come together in the design dimensions.

5.3.1 The link between VSD and production methodologies

During an interview on "Why do Ethics and Values Matter in Business and Information Systems Engineering?" Prof. Jeroen van den Hoven mentions that an important area of research could be to design and evaluate agile methodology and methods for value-sensitive design (Maedche, 2017,p300). The software development process will provide insights into the design and creation of new software. If we scrutinise the software development process, we come to the conclusion that within the industry, engineers will use development methodologies and tooling that define how a product is built.

Some methods will be better suitable to incorporate VSD and Moral Intelligence than others. When analysing the advantages and disadvantages of these development methods, a new method for VSD and cultivating Moral Intelligence may emerge. To cultivate these production methods the strength and weaknesses of each method has to be analysed which is the main goal of this chapter. Once the strengths and weaknesses are known a selection and better method may be developed for the incorporation of ethics.

5.4 Methods of development and industry standards

In the IT world, there are methods for development, e.g. frameworks that help designers and engineers build their technology. Software development is the set of activities that results in software products. Software development may include research, new development, modification, reuse, reengineering, maintenance, or any other activities that result in software products. Thus, in software engineering, a software development methodology splits software development work into distinct phases (or stages) containing activities with the intent of better planning and management (McCarthy, 2006).

If we understand these activities, we can breakdown development and link in VSD, where possible. The development of software (including the design aspect) takes place with several design and production approaches, which also include the developer tooling. For this thesis, particular development methods have been analysed that are being used within the companies CAPE Groep BV BV, Elnino-ICT BV and TRIMM BV. Although there may be other development methods possible, the analysis focuses on the feedback that was provided mainly from these companies to gain more insight into processes rather than only description.

A well-known term within the software development industry is Agile development that incorporates multiple development approaches and philosophies. Contrary to Agile development is the Waterfall approach.

5.4.1 The Waterfall approach

The Waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases. These phases are often considered as sequential. This model assumes that every requirement of the project can be identified before any design or coding occurs. One major difference between agile and waterfall models is the approach to quality and testing. In the waterfall model, there is always a separate testing phase after a build phase; however, in agile development, testing is usually done concurrently with, or at least in the same iteration as, programming.

Waterfall consists of five main practices: system requirements, concept, analysis, design, coding and testing (Qumer & Henderson-Sellers, 2008).

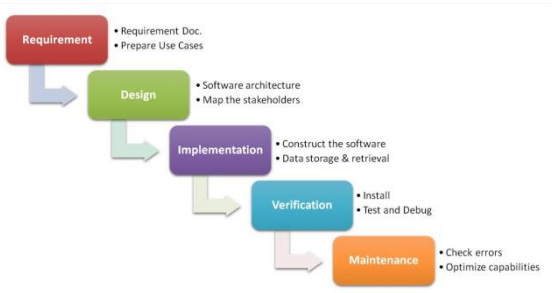


Figure 5.1: The Waterfall approach. Source: (Subbaiah, 2016)

Below are the pros and cons[8]

Strengths

- In general, Waterfall provides a clear and specific project plan.
- Waterfall stresses the importance of documentation.
- Because of the heavy focus on planning, timetables and budgets can be estimated more accurately.

[8] (Mikoluk, 2013)

Weaknesses

- Each stage must be reviewed before going to the next stage. This may be very time consuming.
- The Waterfall method is incredibly rigid and inflexible.
- All information is gathered upfront.
- Altering the project design at any stage in the project can be detrimental, e.g. once a stage has been completed, it is nearly impossible to make changes to it because of the upfront specifications.
- In addition, the problem with the Waterfall method is that feedback and testing are deferred until very late into the project.

Best parts for incorporation of VSD

When looking at the strengths of Waterfall, we see the importance of solid documentation and a clear and specific project plan. In this sense, it stimulates foreseeing requirements. In turn, this may be an opportunity for the incorporation of VSD to investigate the documentation or stages and find where VSD specifications can possibly improve quality.

Table 5.1: The Waterfall : VSD

Waterfall strengths	VSD possible match
-Documentation	+Stimulate the implementation of protocols
-Foreseeing requirements	+Create ethical standards
-Staging development	+Create a focus on quality

5.4.2 The philosophy of Agile development

Agile development was designed to provide light-weight development methods. During the late 1990's, several methodologies gained increasing public attention. They emphasised close collaboration between the programming team and business experts; face-to-face communication (as more efficient than written documentation); frequent delivery of new deployable business value; tight, self-organising teams; and ways to craft the code and the team such that the inevitable requirements churn was not in crisis. Qumer and Henderson offer the following definition for the agility of any entity:

"Agility is a persistent behaviour or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment" (Qumer & Henderson-Sellers, 2008). Moreover, they also provide a definition of a software development method:

"A software development method is said to be an agile software development method when a method is people focused, communications-oriented, flexible (ready to adapt to expected or unexpected change at any time), speedy (encourages rapid and iterative development of the product in small releases), lean (focuses on shortening timeframe and cost and on improved quality), responsive (reacts appropriately to expected and unexpected changes), and learning (focuses on improvement during and after product development)" (Qumer & Henderson-Sellers, 2008).

An Agile manifesto was developed to provide general guidance that summarises the above philosophy("Agile Alliance :: The Agile Manifesto," 2001).

The Agile Manifesto

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle.

The below schematic highlights the importance of this philosophy (Ellis, 2016, p225)

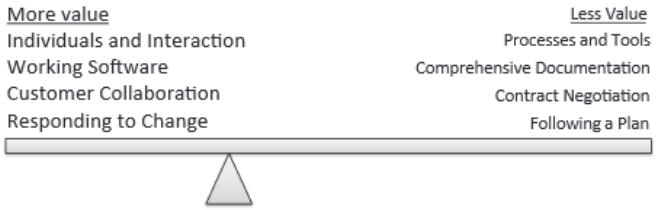


Figure 5.2 **The defining principles of Agile from the Agile Manifesto**. **Source: (Ellis, 2016, p225)**

Strengths:

- Agile promotes an incredibly flexible design model, promoting adaptive planning and evolutionary development.

- Software developers work on small modules at a time. Customer feedback occurs simultaneously with development. (Cockton, Lárusdóttir, Gregory, & Cajander, 2016,p8-9).

Weaknesses

- Agile projects tend to be hard to predict in terms of timelines and budgets.

Why Agile?

Although Agile development was more descriptive (such as the clear manifesto), several approaches derived from this approach.

Best parts for incorporation of VSD

When looking at the strengths of Agile, we see that it heavily promotes iterative development, flexibility, and an adaptive development model. Software engineers develop small modules at a time or milestones and perform software testing simultaneously. Agile development focuses heavily on creating agile teams.

Table 5.2 Agile : VSD

Agile development strengths	VSD possible match
-Iterative development	+Iterative
-Promotes flexibility	+Flexibility is also required
-Works with milestones	+Analysis during creation of milestones
-Respond to change and customers	+Promotes stake holders analysis

5.4.3 The Scrum approach (an iterative approach)

A derived approach of Agile development is the Scrum approach. Scrum is the set of processes that most strongly distinguish Agile from Phase–Gate methods. Where Phase–Gate methods define a project as a series of tasks completed over many months, Agile Scrum breaks a project into short sprints, which are iterations that sum to the full project. Each sprint is almost a mini-project, lasting just a few weeks and ending with a new product that could be released (Ellis, 2016).

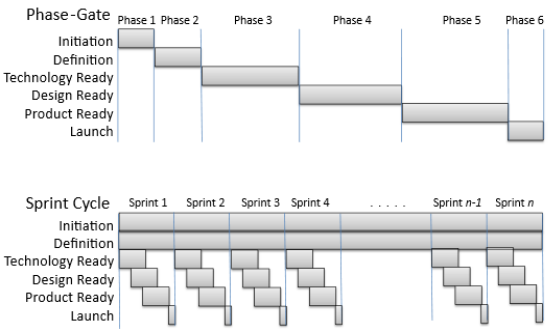


Figure 5.3 Traditional Phase-Gate vs. Sprint cycle development (Ellis, 2016).

Scrum is the most popular way of introducing agility due to its simplicity and flexibility. Scrum emphasises empirical feedback, team self-management, and striving to build properly tested product increments within short iterations. Self-organising teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others outside the team.

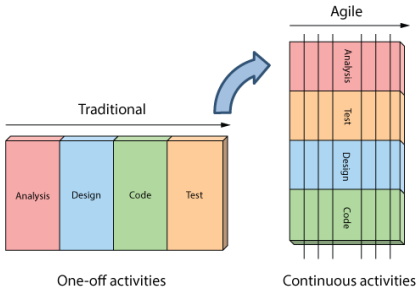


Figure 5.4 Traditional Phase-Gate vs. Sprint cycle development (Ellis, 2016).

The team model in Scrum is designed to optimise flexibility, creativity, and productivity. Scrum has multiple roles: product owner, team, and scrum master. These are described in detail by the Scrum Training Series. The responsibilities of the traditional project manager role are split among

these three Scrum roles. Scrum has five meetings: backlog grooming (aka backlog refinement), sprint planning, daily scrum (aka 15-minute stand-up), the sprint review meeting, and the sprint retrospective meeting.

Practical business case: Trim Scrum, Enschede

TriMM, a web design company situated in Enschede, heavily relies on Scrum within their core-business when developing their products. One product delivered using Scrum (but not only) is hosting[9]. CEO, Marc Woesthuis, states that the acceleration of project execution will be capitalised by Scrum. Plans for digital transformations are being executed. Time to market becomes faster, channels will be integrated. Organisations are not only looking at how processes could be cheaper, but which components of the organisation could be accelerated (Schulkes, 2015).

Strengths

- A Scrum team is not too big in size which makes it very flexible.
- Scrum emphasises collaboration and daily communication.
- Scrum promotes transparency; you no longer need to hide the truth, you can be open and honest with everyone.
- Decision making is shifted to the lowest level (line employees), to the people best able to understand all of the facts

Weaknesses

- Lacking documentation
- Training intensive
- Requires focus

Best parts for incorporation of VSD

Below are the best parts in which Scrum may be suitable for VSD. First of all, the small team size allows for a short communication in which VSD requirements may flow. Secondly, because of the promotion of collaboration a diverse team is also promoted. This is very beneficial for a VSD team or a team which may require VSD elements. Thirdly, transparency is promoted which allows for the openness towards stakeholder their input. Furthermore, decision making is delegated towards teams, instead of management., which means a Scrum team may more easily adopt VSD.

[9] http://docplayer.nl/4272984-Scrum-bij-hosting-philippus-baalman.html

Table 5.3 Scrum : VSD

Scrum strengths	VSD possible match
-small team size	+Short line communication, Iterative
-collaboration	+Creating a diverse team
-transparency	+Openness for input
-collective decision making	+No management involved directly

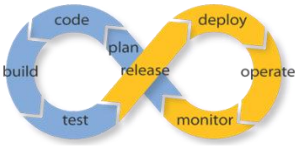
5.4.4 The Continuous Integration Approach (CI)

Continuous integration (also known as CI) is the practice of frequently integrating new or modified code within an existing source control repository. “Frequently” typically means several times a day, when all working code is integrated with a shared mainline. Each check-in is verified by an automated build.

Characteristics: The main aim of CI is to prevent integration problems, referred to as "integration hell" in early descriptions of XP. CI isn't universally accepted as an improvement over frequent integration, so it is important to distinguish between the two as there is disagreement about the virtues of each. Within continuous integration, software is integrated everywhere throughout the software architecture and the software team.

Practical business case: El Niño-ICT, Enschede

El Niño-ICT mentions that CI enforces the discipline of frequent automated testing. During the interview, El Niño-ICT stated that they work via a method called continuous integration. El Niño aims to develop their applications using one solid framework, rather than multiple. This means they integrate several frameworks into one (modified) so the company can develop more efficiently. El Niño-ICT specially focuses on maintaining a code repository and automating the building process.



Best parts for incorporation of VSD

VSD allows for a continuous integrated philosophy meaning that the embedding of systems is key throughout development. A streamlined process and heavy focus on adaptability and

automation may be suitable to create a monitoring system and continuous learning.

Table 5.4 CI : VSD

Continuous integration	VSD possible match
-integration	+Streamlined build process
-automated testing	+Continuous learning
-**embedding** systems	+Monitoring system

5.4.5 The Model Driven Development (MDD) Approach: Core Concepts

There are two core concepts associated with MDD: abstraction and automation. The software application model is defined on a higher abstraction level and then converted into a working application using automated transformation or interpretations. The right MDD approach leverages model execution at run time, where the model is automatically transformed into a working software application by interpreting and executing the model (removing the need to generate or write code). Because of the many changing requirements during a VSD project, an MDE approach would fit the tooling.

A model driven development platform is often referred to as a high-productivity platform as a service due to the unprecedented speed at which users build and deploy new applications. This speed is derived from the use of models and other pre-built components that business and technical teams use to visually construct applications.

To provide a practical example, one big player within the software industry is the Mendix platform that provides visual development:

- Rapid application development by the use of models and domain definitions.
- Business innovates faster. IT stays in control.
- One platform to transform ideas into applications. Designed for business & IT teams. Innovate faster while IT stays in control.
- Agile project management to manage scope and progress.
- Visual application development achieving 6x higher productivity.
- 1-Click deployment for fast, yet controlled release management.
- Central application management for efficient operations.

A company situated in Enschede uses MDD as their core development method. “Modern IT offers opportunities to innovate, specializes, and boost productivity. We work with innovative, model-driven development platforms so that we can realise novel software and integration solutions together with business administrators. In small, flexible teams, we perform close cooperation within people projects.” Our technology includes infrastructure, development, and integration platforms, web-based software, dashboards, and reporting. With the Mendix App Platform, we are developing cloud-based applications and applications for tablets and smartphones. (CAPE Groep | Grensverleggend vooruit, n.d.).

Figure 5.4 Software is done by developing visual models, forms instead of code with the Mendix platform.

Source: https://www.intrapreneur.nl/2014/04/23/applicaties-ontwikkelen-code/

Business case of Model Driven Development (MDD)

CAPEGROEP grensverleggend vooruit

mx mendix

Best parts for incorporation of VSD

Within the MDD approach, there are several opportunities for incorporating VSD. Firstly, because of the usage of models, engineers and ethicists or consultants can understand the technology and components being used. An ethicist does not have to know actual coding but can see the big picture and the processes of the application when taking a closer look at the models. Secondly, MDD development makes it very easy to reuse models and continuously improve them, which may suit the VSD community. Lastly, the central application management (software is deployed as a platform) can make it easier to have features implemented—for example, when a VSD requirement is discussed, accepted and built, it will be very easy to deploy it widely.

Table 5.5 MDD pros and cons

MDD development	VSD possible match
-Usage of development models	+Understandable for novices, e.g. consultants or ethicists
-Re-usage of software	+Components can be improved and reused
-Central application management	+Targets the core development

5.5 Conclusion and reflection

This chapter has shown that different development methodologies may be compatible with VSD, all having their strengths and weaknesses. Waterfall tends to be best for static projects, where it's not likely that many changes will be made throughout the development process. In contrast, Agile tends to be a better option for smaller projects where changes are likely during the design process. VSD may be compatible with a flexible way of developing and allowing interactions of continuous integration philosophies in order to work.

Training VSD at the level of Scrum may be an issue and requires further training. An ethicist may help steer and support a design team. By creating an adjustment of Scrum and implementing VSD practices and tools, VSD could be introduced during meetings, sprints, and to raise ethical issues.

The Waterfall method may provide solid awareness for building documentation to fall back on or stimulating intense development. As a follow-up on this chapter, I strongly suggest developing a method that combines the strengths of the discussed production methods with VSD and the Moral Intelligence framework from Chapter 4.

In the next chapter, a combined design approach is proposed for building a prototype that combines the moral capture framework of which we spoke earlier with the different production methodologies. The VSD methodology will also be embedded within the Moral Intelligence framework.

CHAPTER 6

A Moral Intelligence Practical Guide

You've got to be willing to read other people's code, and then write your own, then have other people review your code. You've got to want to be in this incredible feedback loop where you get the world-class people to tell you what you're doing wrong.

-Bill Gates
Interview from Programmers at Work (1986)

6.1 Introduction

This chapter combines all the components discussed till now and links them to corporate and practical contexts. We take the Moral Intelligence framework from Chapter 4 and the strengths and weaknesses of the software production methodologies from Chapter 5 to shape the blueprint of a VSD design methodology that will fit the IT industry.

Because VSD has its own methods (the conceptual, empirical and technological investigations), I will continue to extend the idea of Moral Intelligence within a more practical field. Since the foundation of Moral Intelligence has been built on parts of the tripartite methodology of VSD, I suggest that Moral Intelligence can also be a standalone operationalised theory, not necessarily dependent on VSD but still embodying the tripartite methodology.

This is done intentionally so that Moral Intelligence can grow as a separate theory but also fall back on VSD theory if necessary. Promoters of VSD may also see this as a benefit to adopt my theories of Moral Intelligence without feeling their core VSD values are at stake.

I will promote Moral Intelligence as a theory that could be hooked into VSD as an iterative progression. If VSD requires more ethical reasoning then use Moral Intelligence; if not, it's up to the VSD participants.

I now come to the important question: *How to integrate VSD outside the academic laboratory and confront values within a design context?*

To start the integration of VSD, I have operationalised VSD within a Moral Intelligence framework in which ethical issues could be gathered. Gathering data is an import process in IT businesses.

Chapter 4 discussed how a framework could be developed for gathering moral data using several investigation methods. The process of gathering specifications and analysing data that can be read by a computer is called "data capture". As this method suits the IT world, I will now call this process "Capturing Data", since data has to be collected and somehow stored and eventually used and converted to data a computer understands (e.g. via a word processor).

Step 1: Capture within the Moral Intelligence Framework

Different data can be captured on different levels of technology, namely the technology at large, the object of the technology, and procedures. Different methods described how data could be gathered. The capturing phase provides a model to raise certain ethical issues using several approaches to gather moral input that can be extended in the future. The data is captured by ethicists, but this could be also done by experts. All the entities involved are given in the Moral Intelligence chart.

Step 2: Design within the Moral Intelligence Framework

The next step is that we require methods that use the captured and analysed data and that can turn it into workable products. A great catalyst for quick deployment through organisations can be the choice of the right tooling within organisational structure. In order to adopt VSD, we may look at exactly what industry tooling is currently used and that may be modified to better support VSD. I have earlier discussed methods of development and their possible match with VSD. It is now time to integrate the two using a practical context, like discipline, tooling and design processes.

Philosophical questions were formulated at the beginning of the thesis. This is also a good point to reflect on those current problems to know what tools may be developed.

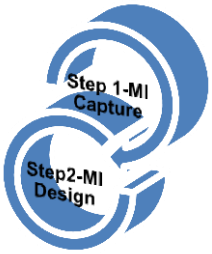


Figure 6.1 Steps in the Moral Intelligence Framework

6.2 How to solve the philosophical questions in practice (2.1-2.3)

I will now reflect on the questions formulated earlier it the research design. It is advised to read the research questions again. For easier reading, I have formulated them once more below.

2.1) How to embed an ethical approach in VSD and in production?

2.2 How to identify the role the values advocate?

2.3) How to select the justification of a value framework?

2.4) How to confront competing values?

2.1 Is now answered: we have a moral framework for capturing ethical issues. I will continue with answering 2.2.

Identifying the role of the values advocate

The role of the values advocate lies to me in the value that a team can have. Identifying the values advocate may not be easy but I propose that the role of the values advocate lies in each member of the development team and responsibility is distributed over each team member as well as contribution. According to Grunwald, in many areas of society, there is general agreement on the proposition that ethical reflection should be indispensable for the development of technology. Not only philosophers, but also many managers and engineers are deeply convinced that ethical reflection plays an important role for the development of technology in a socially acceptable and environmentally sustainable manner. However, as he critically states many situations in engineering as well as in business, can be classified under moral aspects as ”business as usual”. Such situations can be shown to give no demand for ethical reflection – with respect to a normative framework encompassing this particular context (Grunwald, 2001,p416). This is very unfortunate, but a great value lays in the ethical reflection within the business as usual context if possible. How can it be incorporated? Or perhaps even more important why should it be incorporated? I will explain this important “why” question first.

By reflecting on the role of the network-like character of engineering practices, Swierstra and Jelsma find that it gives the opportunity of shifting moral responsibility to others (also see Van de Poel et al. 2012), while at the same time recognition of this structural character would open new options for individual responsibility, by transcending the level of the individual engineer and by creating incentives at the institutional and societal level to enable engineers to assume social responsibility (Pesch, 2015). Lanzo provides from a pedagogic perspective very interesting benefits and examples of new options for individual responsibility (Lanzo, 2003, p3).

1. To increase the knowledge about the duties and moral responsibilities of the engineers in the performance of their professional labour.
2. Transmit essential ethical values for the performance of the profession.
3. To develop human and social skills for team work and innovation.
4. To complement the technical perspective with moral judgments to favour a responsible decision taking that attends not only to the legislation in force, but also to the exigency of universal moral principles.
5. To promote the knowledge and development of professional virtues to obtain excellent engineers that contribute to the progress and the social justice.

In general we can generate three reasons to stimulate the development of a subject of ethics and deontology for engineers which also answers the justification of a value framework (2.3):

- In view of the enormous power of science and technology and the risks that these carry it is indispensable to develop the conscience of moral responsibility of engineers.
- There exists an urgent need to complete technical knowledge with the development of values, attitudes and knowledge that facilitate the professional excellence.
- It is necessary to develop social skills and capabilities of team work based in the respect to the proper values of civic ethics

With these important arguments in mind, it is of great importance to connect engineers with the ethical world. Connecting can be done by creating responsibility. But what is responsibility exactly? Lanzo provides four concepts of responsibility which I will use as a basis to promote the values advocate within a team.

i. *Causal responsibility*, the responsibility for the consequences of our action. Where a person is responsible for the results of his/her action or of the omission of his/her action (negative causal consequence). Also in the field of the causal responsibility, there is a type of responsibility very frequent in engineering, which is the responsibility for prevention (Lanzo, 2003,p2).
ii. *Shared responsibility* (that refers to the responsibility that somebody has for the performance of another person (for instance the father with regard to an unable son). This type of responsibility also includes the responsibility of the boss for the performance of its subordinates; it is the so called control and management responsibility(Lanzo, 2003, p2).
iii. The responsibility derived from the task or the role that occupies the agent is one that refers to the *special responsibility* of role or professional qualification. Here it would be necessary to differentiate between the internal responsibility (towards the members of the organization or the profession) and the external responsibility towards society (Lanzo, 2003, p2).
iv. The responsibility based in the *capacity* is that responsibility that derives from our aptitude to do something. That is to say, if someone is qualified to understand, to plan, to act and to judge, and if he/ she possess the knowledge and capacity of accomplishment as well as the demanded qualification (Lanzo, 2003, p2).

Operationalising responsibility within a development team

Now that different forms of responsibility are made explicit I will connect these conceptualizations within team roles that explains how to confront competing values. Starting from development practice Scrum described several roles which we can take as an example to benefit the industry. The Scrum team consists of a (1) Product Owner, a (2) Development Team, and a (3) Scrum Master. As mentioned before, Scrum teams are self-organising and cross-functional.

The (1) *Product Owner* is responsible for maximising the value of the product and the work of the Development Team. How this is done may vary widely across organisations. In general, the Product Owner is responsible for clearly expressing Product Backlog items by ordering the items in the Product Backlog to best achieve goals and missions; optimising the value of the work the Development Team performs; ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; ensuring the Development Team understands items in the Product Backlog to the level needed.

The (2*) Development Team* is responsible for building the product. In Scrum, a team is not just the executive organ that receives its tasks from the project leader, it is rather a self-dependent decision, which defines the requirements or user stories it can accomplish in one sprint. The Development Team is often a self-learning and flexible team with specialists.

It is the role of the (3) *Scrum Master* to protect the team from external influences and to make sure that everyone abides by the Scrum rules. The Scrum Master shields the team from the product owner and other stakeholders, allowing it to fully focus on the software development ("Scrum Guide | Scrum Guides," n.d.).

The roles of the engineer, the ethicist, and the consultant working together.

Although Scrum describes the product owner, development team and Scrum Master I will simplify for this purpose the roles as the *engineering role* (also found in Scrum), *ethicist role* (advisor) and the *consultant* (product owner) to take a more generic approach which is more understandable. Many roles may overlap each other in the general basis. VSD literature described several roles of the ethicist as a designer (van Wynsberghe & Robbins, 2014). I summarise the following specified tasks of the ethicist according to this paper: (1) value discovery, and (2) translating values into design requirements. These tasks can be further divided into subtasks in order to accomplish the range of goals pertaining to each A-C (van Wynsberghe & Robbins, 2014,p12).

1. Value Discovery

A. making explicit the intended values as conceived by engineers and designers,
B. scrutinising these values by illustrating value conflicts and trade-offs,
C. Comparing intended values with the ethics literature.

2. Translating Values into Design Requirements

A. value conceptualisation,
B. describing the disconnect between intended and realised values by speculating unintended uses and contexts of the artefact,
C. Exploring the translation/specification of values into design requirements.

These suggested roles describe the function of ethicists as a designer but leaves really important questions open for the (shared) responsibility and contribution of other members of the development team as discussed earlier.

In the earlier introduced Model Driven Development method practiced at Cape Group BV, *consultants* also do actual engineering and may have a role in *special responsibility* (iii) within a project. This means the consultant has an engineering role (building a product) and also a very important consultant role. Furthermore, he can have a shared responsibility within the project (ii). In other words, all team members may contribute to VSD and ethical discussions within the team will lead most likely lead to new features.

This perspective builds on the view that an ethicist could also be a designer but a consultant or engineer can also contribute to the ethical outcome in either consequently in a good or bad way (i). For example: what about an engineer who happens to know a lot about ethics due to a minor in philosophy? Especially within the creative technology industry an engineer can come from different backgrounds, engineers from the first days when Apple computers was founded had different backgrounds in liberal arts (e.g. philosophy or social science), not only from an engineering background. Steve Jobs famously mused that for technology to be truly brilliant, it must be coupled with artistry. “It’s in Apple’s DNA that technology alone is not enough,” he said. “It’s technology married with liberal arts, married with the humanities, that yields the results that make our hearts sing” (Segran, 2014). I suggest not demarcating between the ethicist role and someone who is *capable* to understand the role of ethics (iv).

In order to build a multidisciplinary framework to work better together, I will start with the roles of ethicists as a designer. I suggest, in addition to the mentioned contribution by an ethicist that value discovery/translating values into design requirements may also be done by engineers or consultants and not solely by ethicists. The role of capturing values and their conceptualisation may be distributed over the entire group (e.g. other researchers, consultants, and ethicists) which is mediated through responsibility. Very essentially, Scrum does not demarcate between roles or skills within the development team and only makes judgment about what is valuable for the whole group. I would argue that for the most valuable output, the designer, ethicist, and engineer all have the unifying “consultant role” in common, which makes all of them responsible for the outcome.

For example, an ethicist may help translate values into design requirements (e.g. through specification, making values explicit), but the entire team is responsible for the total result. An engineer may have good suggestions too regarding moral issues based on his experience (that goes well beyond his engineering knowledge). Often, IT consultants who also have an engineering role speak for customers or for the public. Consultants possibly know their customers and stakeholders quite well, making them very important for the development team. Because of this liaison role, they can take an active role in value discovery as well. Ethicists and IT consultants act as liaisons when making contact with people outside their area of responsibility, both inside their organisation and outside.

Now to finally answer on question 2.2: How to identify the role of the values advocate? I firstly suggest looking at the team and integrating the above insights to form an overview of domains and possible roles. In other words, what needs to be added to the above scheme is the domain and the other roles, namely the engineer (programmer) and the consultant (in broad term), and how THEY work in practice. As Scrum teams are focused on agility, specialised teams, and stakeholders, Scrum gives a great window of opportunity that needs to be operationalised. A suggestion for this is given below.

Step I: Capture and start with an Agile team first

In the upcoming overview I have explicated the roles as suggested by van Wynsberge of the ethicist and I have extended this with the role of engineer/programmer and the consultant. Scrum gives a good workable method for an agile and flexible workflow, one that suits VSD because of its iterative approach to create a workable setting. These roles could be incorporated within a Scrum team.

Table 6. 1 Moral Intelligence Team Matrix - Distribution of roles and domains to support learning

(Shared) responsibility		
Multiple responsibility	Capable responsibility	Special responsibility
1 **Value Discovery Ethicist**	**Engineer /programmer**	**Role Consultant**
A. Making explicit the intended values as conceived by engineers, designers or consultant.	Provide input & arguments for the intended values together with the ethicist.	Provide input & arguments for the intended values.
B. Scrutinising these values by illustrating value conflicts and trade-offs.	Provide input e.g. using the Moral Intelligence framework together with the ethicist.	Provide input e.g. using the Moral Intelligence framework together with the ethicist.
C. Comparing intended values with use of Moral Intelligence framework, e.g. ethical literature, workshops or value for design methods , etc.	Make a priority list of the intended values and ethical participation in workshops, brainstorms sessions.	Make a priority list of the intended values and ethical participation in workshops, brainstorms sessions.

2 Translating Values into Design Requirements Ethicist	**Engineer/Programmer**	**Consultant**
A. Value conceptualisation (Customers, Public)	Gathering, providing feedback for the ethicists and consultant e.g. with the use of Moral Intelligence.	Value conceptualisation with customers and stakeholders (liaison role).
B. Describing the disconnect between intended and realised values by speculating unintended uses and contexts of the artefact.	Gathering (realised) value concepts given by ethicist or IT consultants and advising which engineering concepts may be suitable.	Mediate between the engineer and ethicists to find the best possible solution between intended and realised values e.g. within the role of a product owner (as an advisor to the ethicist)
C. Exploring the translation/specification of values into design requirements.	Exploring and realising the best possible building blocks for the implementation of values.	Exploring the translation/specification of values into implementation.

As argued in Chapter 4, for solving the problems (2.1)(2.2)(2.3), the concept of Moral Intelligence is key to balancing these values and a pragmatic approach has to be proposed—one that emphasises learning, self-reporting, and monitoring—as the most helpful way to address the ethical implications of technology.

The Capture step does not demarcate between the roles of *who* captured the values to maximize intelligence (knowledge). It is important to have 360 degree feedback, meaning direct input from the whole team during development—something Scrum actively promotes. The IT consultant, engineer, and ethicist could work together to speed up the project without

demarcating between one role being more important than another (overcoming the problem of a hostile environment). A consultant could also be a sociologist or any other expert brought into the team to provide significant contribution.

But what about confronting values and implementation as provided in question 2.4? In order to solve this question, we require the right tools that allows issues and features to become more transparent but this is also a bit of an operational problem. Therefore, let's solve the operational problems first and keep the confronting values dilemma in the back of our minds.

6.3 How to solve the operational problems (2.1-2.4)?

Step II Implementing VSD in the building process and practical tooling

Now that most philosophical problems have been discussed let's take a closer look at the operational problems.

(1.1) How to solve the difficulty to properly explicate and translate ethical considerations to workable requirements and specifications for other project participants actually building the system?
(1.2) How to solve the problem to operationalise the values so they could incorporate into technological designs?
(1.3) How to solve the creation of a hostile environment?
(1.4) How to solve the rejection of ethical concerns to speed up the project?

To answer 1.1 and 1.2, I will make a use of a fictional case to illustrate a scenario and a solution. 1.3 and 1.4 will be dealt with in the next chapter to break down the above questions.

Let's start with the first question (1.1). To answer this question, I first recommend a close examination of the development methods being used within a company. The previous chapter discussed many development methods, with their own unique approach in gathering requirements and building software on the process level. It is now the question how to cultivate them with Moral Intelligence? I will start with model driven development (MDD), as it is a more understandable method to build software for non-programmers.

Engineering methodology, requirements, monitoring & tooling

A company that uses MDD within its core business is the IT consulting company, CAPE Groep BV BV, situated in Enschede, the Netherlands. For illustration the development team consists of an engineer (Pieter), an ethicist (Laura) and an IT consultant (Paul). They are developing an application that calculates the carbon footprint of trucks based on fuel usage calculated by emission factors ("Activity Based Carboning," n.d.).

Step 1: Requirements through user stories and model driven development

First we look at what production methodology is currently being used. Besides a great team, the right resources are required to build software as well. The IT company uses MDD as its development method.

Design and Development Analysis

MDD uses models as a basis rather than code. Because of this, there is a gap between business analysts and developers in traditional development. MDD can fill this gap so that communication between business analysts and developers runs smoother (van de Braak, 2010).

The next page shows the development of a carbon footprint application developed by CAPE Groep BV using the MDD software development platform, Mendix[10]. The model represents the mechanics of the working application without presenting any code. A non-programmer may not be able to read code, but the processes and the decisions made in this application may be clearer than actual programming code because of the easily interpretable MDD approach.

If a process changes in the future or there needs to be a new process then a design flow can be easily added as well. This way of developing provides a greater insight into the scenarios, processes, and values that may concern a project.

[10] www.mendix.com

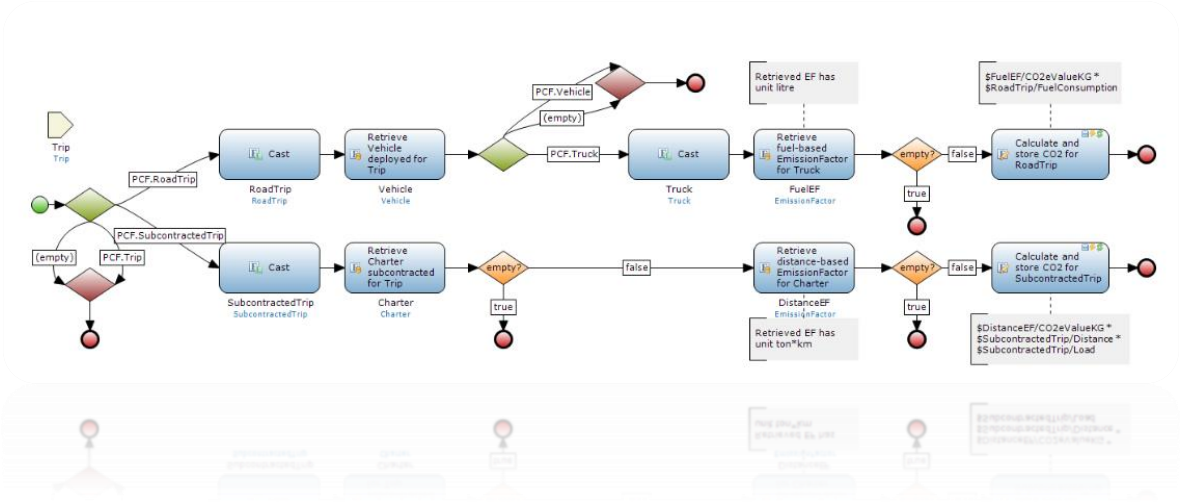


Figure 6.2 MDD allows for quick evaluation, design, and software building (Steenwijk, 2011)

How does MDD work?

Because of the underlying model driven architecture, projects can be built using scenarios. Below is a generic model of the MDD approach. Consultants develop their software using requirements, stories, scenarios, and then create a test model which can be explored and generated to build applications. The processes will be observed and any problems found in the code/processes will be evaluated with new requirements.

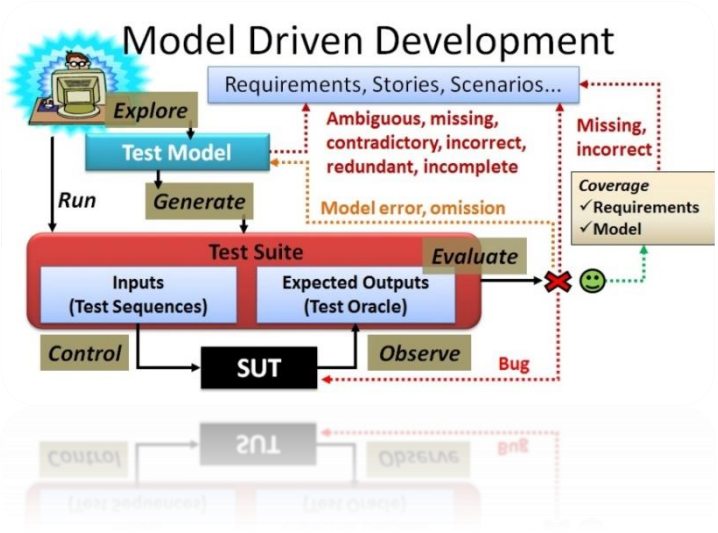


Figure 6.3 Requirements, Stories, Scenarios and a test model. Source: http://robertvbinder.com/wp-content/uploads/sites/4/2012/02/mdd.jpg

During the exploratory phase, as seen in the above schematic, our framework may be used to capture values which are turned into models by ethicists, engineers, and stakeholders by assessing the requirements, stories, and scenarios.

Together the team woks on building the domain model (a data model that describes the information in the application domain in an abstracted way). The domain model consists of entities and their relations represented by associations. Accordingly, the IT consultant builds the software after discussion and feedback, e.g. with an ethicist during an iterative process. Using this method, developers can focus on users and stakeholders to create requirement stories and also instantly test a model in conjunction with the whole team. They can define the inputs, decline or approve, and evaluate the expected outcomes. The processes can be evaluated by the ethicist or consultant as well due to the MDD approach. This method is fundamentally different than normal coding an application. The MDD approach promotes overview of the project and its processes even for non-programmers, making it easier to discuss its processes with ethicists.

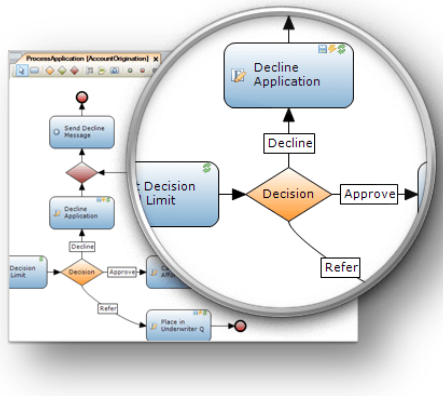


Figure 6.4 MDD allows for the evaluation of which decisions are made in which process
Source: http://www.agile.co.za/products/mendix

Below is a more concrete summary of the MDD approach and its relation to the proposed capture framework.

Within the MDD approach there may be plenty of room for scenario sketching and the analysis of a larger scope by using the Moral Intelligence framework and team matrix, provided earlier. An ethicist may dive into literature, perform a NEST analysis, and look for debates, perform a survey, technology assessment into an ethical capturing system (examples will be given) where ethical issues can be reported.

1. Evaluation can be carried out by consultants and ethics (non-programmers)

There are several benefits within this process:

1. In general, engineers or consultants can rapid develop and change processes and requirements that suit Moral Intelligence and Continuous Integration.
2. Ethicists may have a consulting and check role and participate in the software building process because they can understand what the application is doing.
3. The above model provides a way to steer more towards implementing values (visually), e.g. during sprints.
4. In the future, Moral Intelligence may use mechanics of MDD and build Moral Intelligence tools to better support values to be incorporated into software.

Specific team roles

5. The incorporation of values and the procedure specifically can be modelled first by Paul (IT consultant) and then evaluated by Laura (ethicist).
6. The engineer (Pieter) and ethicist (Laura) can start with a good conceptualisation of values and build processes together (pair-designing). The ethicist helps to design the process and the engineer helps to implement the features and search for the most compatible components.

6.4 Building community tools with object orientated languages

What about other development methods discussed earlier? Although the above MDD approach is very novel, not all IT companies may work with MDD. Most companies write their software using Object Orientated Languages (OOB), an architecture in which everything (processes, files, operations, etc.) is represented as an object like in programming languages like Python, C++, Objective-C, Smalltalk, Delphi, Java, Swift, C#, Perl, Ruby and PHP.

All these programming languages have an important concept, namely they work with classes, objects, and properties (parameters), which makes modular or reuse of code easier. I suggest that community tools for Moral Intelligence may be built and shared among developers and researchers using these object orientated languages.

6.4.1 Sharing code through community, a modular approach

The object orientated architecture does not involve visual model development, like the MDD approach, but building modular compatibility in object orientated architecture is very common. The benefit of a module is allowing new (external) applications to be accessed within a main application. The module will add new functionality to the project without upgrading the main software application.

As with MDD, we must first make use of the Moral Intelligence capture framework and the extended roles that have been provided. Secondly, to operationalise values, we need a technical system that can be accessed during development—at best, a supportive system, one that supports learning, monitoring and reporting. I will provide a basis to create such as system.

6.4.2 How to capture values technically using Moral Intelligence systems?

I will now answer: How to solve the problem to operationalise the values so they could be put into technological design (1.2)?

When developers use Moral Intelligence as a design method, they may use a (open-source) module with definable classes, objects, and

properties that they can share or adjust for customisation per project. Similar to the MDD approach, modules can be shared and be made available for other platforms as well. I suggest that a goal is to build architecture and tools to be incorporated in different IT systems, e.g. creating libraries and modules that can be made available for different ethical systems. This will help the VSD community and Moral Intelligence philosophy to grow the support for the ethical incorporation of values. A great practical tool that was already provided by the VSD community is envisioning cards. Based on roughly two decades of research into accounting for human values in the design of technology, envisioning cards were developed by the Value Sensitive Design Research Lab at the Information School at the University of Washington. The envisioning cards are built upon a set of four envisioning criteria that are intended to raise awareness of long-term and systemic issues in design. Each envisioning card represents a specific theme within one of these envisioning criteria(“About | Envisioning Cards,” n.d.)

The envisioning card can also be translated into community applications that can be hooked into an Integrated Development Environment (IDE) that developers can use during development and use when they speak with customers or implement software during a Scrum sprint.

Figure 6.5 Envisioning cards. **Source: http://www.envisioningcards.com/envision_pdfs/Values-Value-Tensions.pdf**

Values may be saved as a priority within the application and be accessible throughout development once they have been captured. Also, the envisioning cards may be brought up during a Scrum Sprint; a consultant using the MDD approach can directly evaluate their model and make the necessary adjustments. The cards will be digitally used in the form of apps or within a development platform.

Figure 6.6 VSD Capturing values through apps that can be used during development. Apps allows for quick storage and further assessment of the input data supporting the Moral Intelligence learning, monitoring and reporting philosophy.
Source: image is specifically designed for this thesis.

The above shows that the idea that the envisioning cards can be also designed as a digital app to capture data. Modules and tools may be developed that can be used as input for repositories. We recommend building multiple modules that can be directly accessible from industry tooling, such as MDD or even the building tools of developers. The above screenshot is just an example that, of course, requires an extension of ideas and an improved reporting feature. I argue that this is also the first step for confronting competing values. By making these values more apparent within teams, a basis for Moral Intelligence is created. However, more has to be done to create interaction and value confrontation within teams.

In the next section, I will discuss Moral Intelligence tools that may speed up the process of monitoring development and value implementation.

Moral Intelligence and Community Tooling

The earlier app idea is just an example of how VSD/Moral Intelligence community tools may be extended. As a response to the further development of VSD, Borning & Muller suggested that opening up the process to broader participation and sharing of power would be useful. According to Borning & Muller, forms of community-led VSD may emerge (Borning & Muller, 2012). VSD researchers could provide collaborative online tools to facilitate sharing, discussing, and refining research materials (e.g., lists, envisioning cards) among the growing community of VSD practitioners.

With the Moral Intelligence blueprint, the team matrix and the adjusted methodologies in mind, it would be possible to extend towards a supportive community of tools and practices to allow better confronting of values (as an answer to 2.4 from the previous chapter). One way of extending to a community is by making the knowledge more accessible (besides creating a mobile app). A good way of doing this is by creating a VSD/Moral Intelligence repository where people could access useful information. For example, a useful resource would be the usage of a Wiki and an online help forum with the aim to provide practical implementations of VSD/Moral Intelligence and related discussions. A development team can access information they need and discuss their development with other participants.

Examples:

- The creation of a wiki, forum, and online repository to share practical knowledge.
- Share and incorporate intelligence through apps, cases, software components or values and experience gained from previous development.

Building and sharing experience is a very valuable asset for future projects which should not be underestimated. For example reusable (ethical) components may be a catalyst for future development.

Building the basics for a learning, monitoring, and reporting system

With created apps for VSD and Moral Intelligence, it will be possible to share code and applications on a higher level than just abstracted theory. The aim is that workable tools (apps, forum or repository) will be hooked into existing software development architecture to guide development. A main benefit would be to build software that allows us to think about ethical considerations and that can be used by anyone, opening up ethical checks, reasoning and interventions during development as Moral Intelligence suggests with its learning, monitoring, reporting system. The community e.g. VSD practitioners, developers, product owners, consultants can learn more about ethical topics and learn more about value conceptualization.

The community thought

- Sharing source code that has been used during a project with consultants.
- Sharing experience and project details that may have involved thinking about values and ethical issues.
- Reflect on procedures that may involve reflection on values

Below is a suggestion of what such tooling and a community roadmap could look like.

Community roadmap: Step 1 Create a repository for learning and monitoring

First, the data has to be store in some online database (repository) where specific information about VSD/Moral Intelligence can be obtained easily from anywhere in the world. Think about general information, policy, (quality) standards, literature, but also making the empirical research data available.

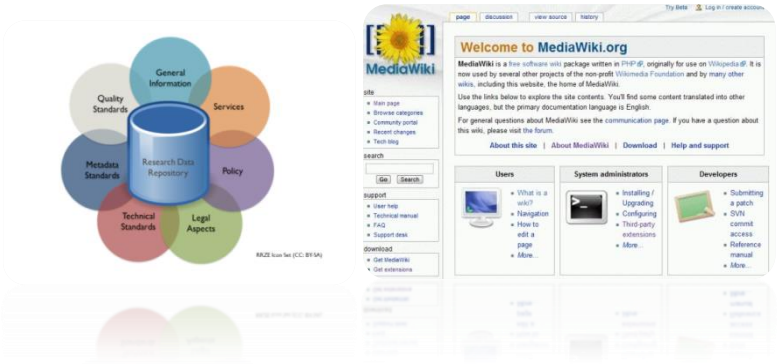


Figure 6.7 Wikipedia like VSD library for easy access.
Source: https://www.mediawiki.org/wiki/MediaWiki

Community Step 2: example create a wiki to create VSD standards

Once a repository is created the data may be presented more clearly online. A way of presenting processes is by having a detailed wiki (an online encyclopaedia) about developments where people can contribute. MediaWiki is a free server-based software which is licensed under the GNU General Public License (GPL). A Wikipedia may be essential for looking up information and getting an overview of all the methods used in the literature. I particularly suggest creating a good method overview that concerns all empirical, technical, and conceptual investigations.

Community Step 3: Creating an integrated web forum for dialogue and feedback during development

A specially design web forum may be used to provide support to researchers and developers around the world for easy access. A forum can also be integrated within any development environment to support accessible knowledge. Below are some topics a VSD forum may contain.

Community VSD Forum for learning

1. Getting started

This provides the basic Q&A for getting started with VSD/Moral Intelligence

2 VSD/Moral Intelligence methodology

In this section, people may ask questions about the VSD/Moral Intelligence methodology, e.g. concerning the tripartite methodology.

3 Resources

In this section, people may find interesting cases and stories of successful VSD/Moral Intelligence implementations. They can also get in touch with the right people to promote training sessions and workshops.

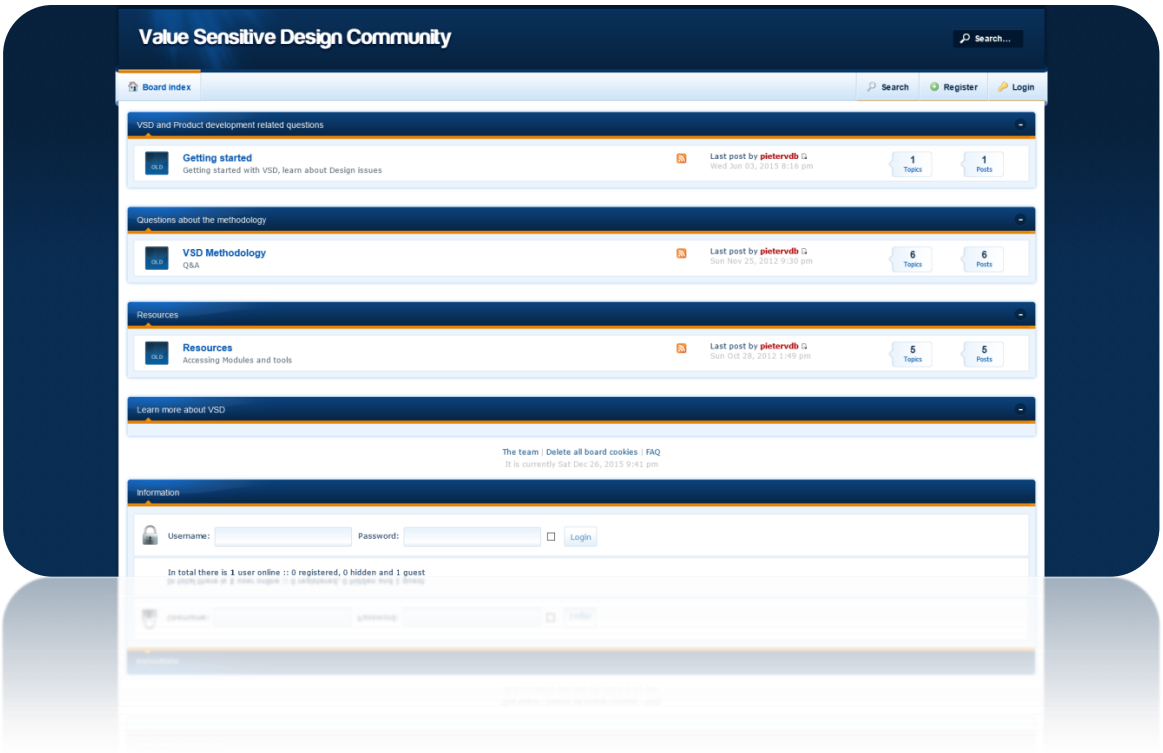


Figure 6.8 Example of interactive forum (example created specifically as an example for this thesis)
Source: image is designed specifically for this thesis**.**

Step 4: Moral Intelligence platform integrations

VSD could be accelerated with the support of major development platforms. For example, distributing envisioning cards in a modular form to these platforms can help developers learn about their decisions and allow them to be more creative familiar with VSD. A suggestion would be to develop modules that can be integrated within the digital working environment of engineers or consultants. Such modules can be directly loaded from within the interface of the development environment. The goal is to create architecture that could be integrated within (normal) software development by enforcing building reusable components and to support the distribution of Moral Intelligence through tooling.
These development environments may include commonly used software such as:

- Visual Studio[11] (for a wide range of programming languages and platforms)
- Netbeans[12] (e.g. for Java development)
- Xojo [13](Windows, Mac, Linux development software development)
- Code igniter (website development)

[11] https://www.visualstudio.com/
[12] https://netbeans.org/
[13] http://xojo.com/

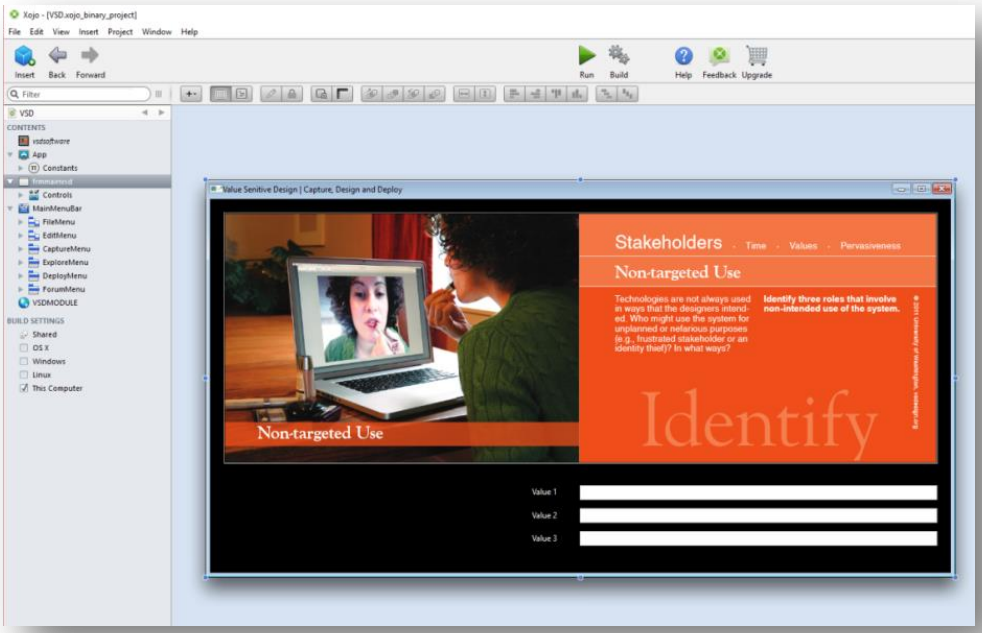


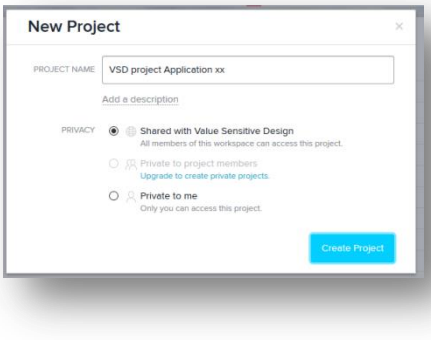
Figure 6.9 Running a VSD module from an IDE to be further developed, similar may be developed to assist during sprints or in general.
Source: image is designed specifically for this thesis**.**

In the above example, a module is directly loaded from within the software development environment. Being open source means that people can modify the module themselves, e.g. values that are given as an input can be directly used within the project or within a database for re-use. A Scrum team can use the modules as standalone plug-ins or modify the modules for later usage. There are many possibilities of developing the right architecture and tooling that can help people engage with VSD/Moral Intelligence.

1. Support the development and introduction of more value discovery apps.
2. A Moral Intelligence app to capture and access literature and intelligence.
3. Promote a feedback system for ethisist, programmers, consultants to perform ethical analysis and implementation.

Retrieving captured data from a learning, monitoring & reporting system

The proposed Capturing Framework may be translated into apps as I suggested with the app of the envisioning cards. Another example is the usage of a flexible task/feedback system such as Asana.com which opens up more



conversation and participation.

Figure 6.10 Creating a new project

With Asana.com. you can create a task list and share it with other stakeholders, colleagues, and developers. Feedback can be given by ethicists and engineers on these tasks. For example, a new project is created: “VSD project application xx”.

Next, a task list is created which can be seen by the developers but it will also be reviewed by the consultants and ethicists. For example, during the building of a login form for an application, the task will be first be added into Asana. Next, this task will be reviewed by the ethicist and the *programmer and ethicists* will discuss the login features. Asana allows adding ethicists, stakeholders so they can directly support or reject values.

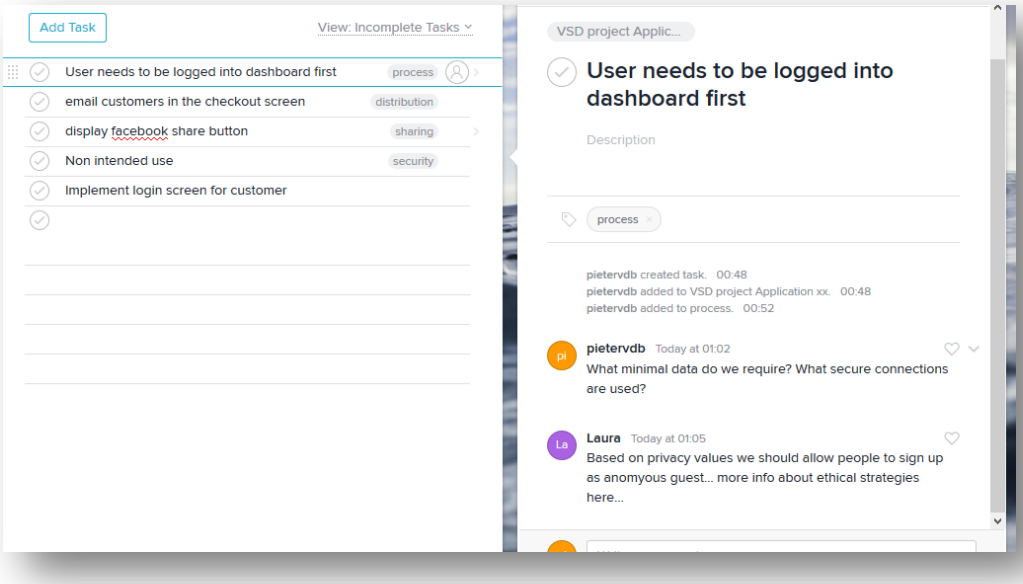


Figure 6.11 Discussing login of a dashboard by ethicist and developer. Source: image designed specifically for this thesis.

Discussion of sharing features

In the below example, ‘’Facebook sharing’ is discussed

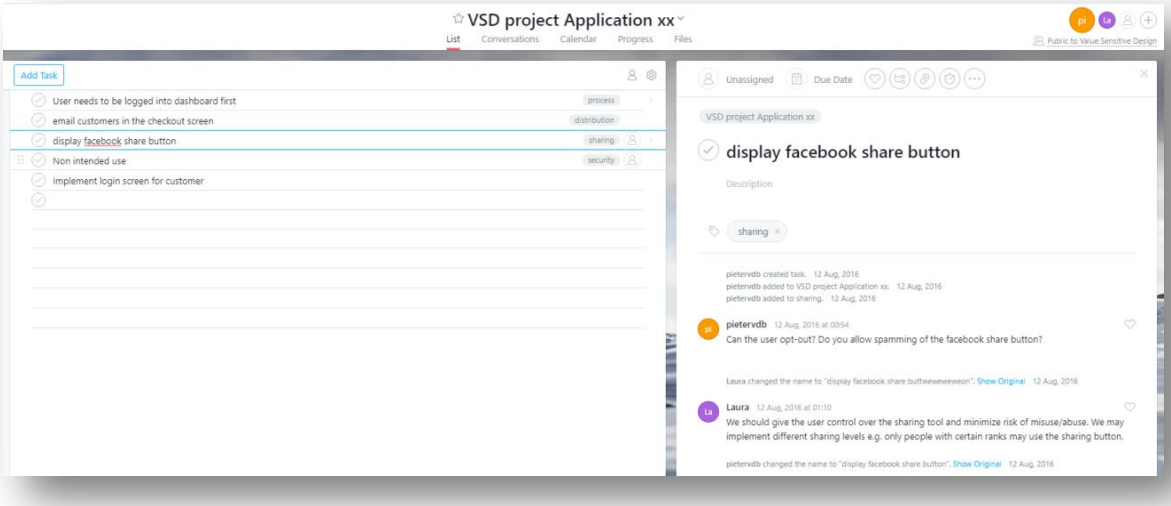


Figure 6.12 In the above example, a minimal feature for a Facebook share is suggested by the ethicist, Laura.
Users may not instantly use the Facebook share feature, unless they have gained a certain credit and trustworthy level.

Source: image designed specifically for this thesis.

6.5 Step 3: Deploy within the Moral Intelligence Framework

Now that requirements and examples of possible industry tooling have been discussed, there are still two questions left regarding the rejection of ethical concerns and the creation of a hostile environment.

(1.3) The rejection of ethical concerns to speed up the project.

(1.4) The creation of a hostile environment, These questions are mainly organisational questions so I will first combine all these steps into an iterative

I argue to solve these problems we have to look at the mechanics and organisational structure of companies. One of the cornerstones of Moral Intelligence is making a company aware of its actions, not just during a short moment, but always. Therefore, we need a deployment strategy to deploy VSD and Moral Intelligence through the organisation. Below are the processes that have been introduced until now Capturing Moral Intelligence have been discussed in Chapter 4, Cultivating Moral Intelligence during development has been discussed in Chapter 5 and 6. Now it is time to develop a strategy to incorporate (Deploy) Moral Intelligence within organisation.

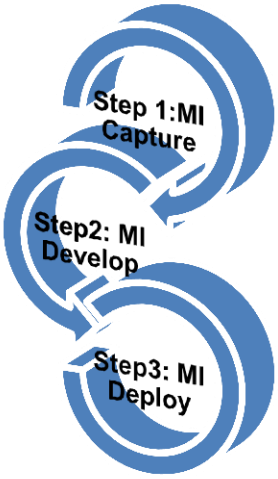


Figure 6.13 Pragmatic three step Moral Intelligence development model

In Figure 6.13 an iterative Moral Intelligence cycle is shown. This suggestion uses the philosophy of Scrum and the continuous integration philosophy (development conceptualised as an iterative cycle) as were strong foundations for a flexible team. Change of requirements are dealt with during development and the below model can be interpreted as a full development cycle in which values are captured, components are developed by the production methods, and deployment occurs throughout the organisation.

Figure 6.14 Moral Intelligence Methodology: Iterative building blocks for VSD/MI

Step 3 Deploying within organisations (VSD and management)

Reflecting on organisations, how the people in the organisation approach their work and make decisions is driven by a confluence of many factors. Organisational values and the resulting culture they create serve as an "invisible hand" guiding these outcomes (Reddy, 2016). The aspects, philosophy, value, vision, mission, goals, leadership, social systems, format systems, informal systems, intangible systems and group dynamics all contribute to the organisation outcome.

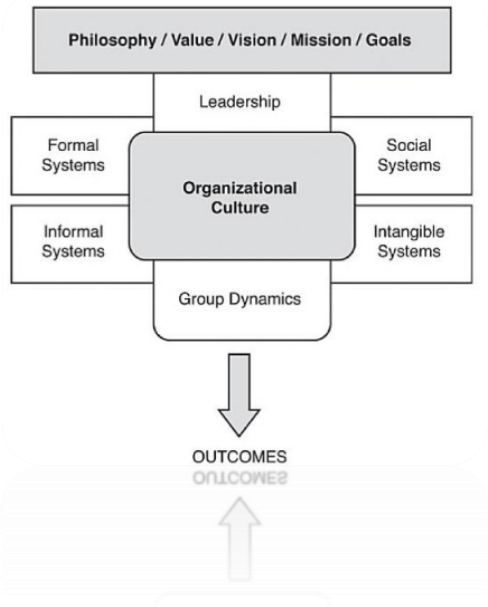


Figure 6.15 How the people in the organisation approach their work and make decisions is driven by a confluence of many factors. Organisational values and the resulting culture they create serve as an "invisible hand" guiding these outcomes Source: (Reddy, 2016)

To start implementing VSD, or even Moral Intelligence, it is important to understand how companies are structured. To begin developing our deployment strategy, the 7s Model analyses by

McKinsey shows how to conceptualize corporate culture. In a practical context, the 7S model is often used to improve the performance of a company, examine the likely effects of future changes within a company, align departments and processes during a merger or acquisition, and to determine how best to implement a proposed strategy ("McKinsey 7Ss Model | Developing Together," 2015).

The basic premise of the model is that there are seven internal aspects of an organisation that need to be aligned if it is to be successful. As stated by Roland by InterDC earlier in the thesis, several considerations need to be taken into account for successful implementation. Will the organisation be flexible enough? Will there be enough resources? These are important questions that may be asked. The McKinsey organisational model acknowledged that corporate culture is built up by *structure, systems, style, staff, skills, strategy, and shared values*. Now that we know these values, we can assume that the rejection of ethical concerns to speed up the project may be related to the supportive systems available within the corporate setting. The hostile environment can be seen as a very important factor that causes the rejection of ethical concerns to speed up a project.

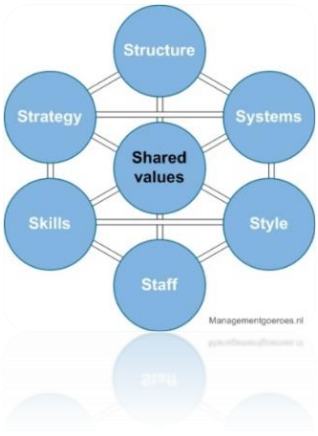


Figure 6.16 The 7S Model (Source: McKinsey, 2015)

There are two categories of elements that defines organisational culture

Hard elements are easier to define or identify and management can directly influence them: these are strategy statements; organisation charts and reporting lines; and formal processes and IT systems. There may be important connections between two dimensions. For example, between staff and structure, where people may need creativity but there may be too much controlling structure and procedures to create a corporate setting where innovative ideas can be generated.

- *Strategy:* Every company has its own strategy and when implementing VSD/Moral Intelligence questions will be raised whether this will fit the corporate strategy? A new project must first be introduced to the company or a team. Alignment must take place between the corporate vision and mission and the strategy (how to get there?) must be developed. This, in itself, may be seen as a difficult process.
- *Structure:* Every company has its own infrastructure. A good question to ask is, is the right infrastructure offered for implementing VSD/Moral Intelligence?
- *Systems:* The system dimension describes all the systems that the company is using. Will there be supportive information systems that support VSD/Moral Intelligence?

We can also identify Soft Elements

"Soft" elements, on the other hand, can be more difficult to describe, and are less tangible and more influenced by culture. However, these soft elements are as important as the hard elements if the organisation is going to be successful.

- Shared Values: the shared values of the staff such as vision and mission.
- Skills: Will the staff be skilled enough to implement VSD/Moral Intelligence? Or does the team need additional training or workshops? Is there enough flexibility in the team?
- Staff: Is the staff diverse enough to add value and contribute to VSD/Moral Intelligence?
- Style: Is the management style and production style flexible enough to adopt new methods?

In the next figure, I will show the link between different connections and possible problems that may occur: e.g. problems within the link between strategy and structure, systems and staff and staff and structure.

Example of VSD problems

Below important links are described between the contingencies and what can be motivators for unsuccessful implementation of VSD/Moral Intelligence and the rejection of ethical concerns to speed up the project.

Example problem 1: There may be not enough resources available to support the strategy and structure for a good work environment. Engineers and project managers may experience, therefore bottle necks and rejecting ethical concerns to speed up the project.

Example problem 2: Staff may not be able to have the right reporting tools or systems to incorporate Moral Intelligence or VSD. Systems may include the absence of a feedback system.

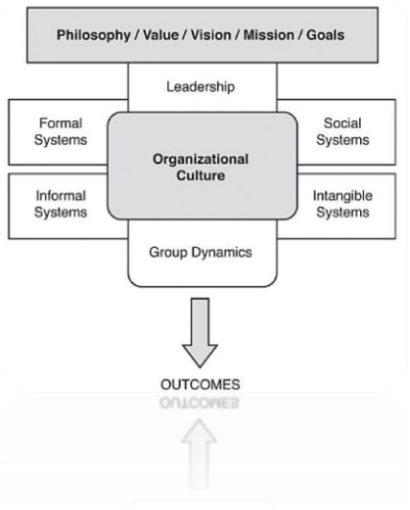
Example problem 3: The organisation culture does not support creative ideas and out of the box thinking. Corporations could be lacking creativity due to inflexible staff and structures. Teams and managers could create corporate settings where creativity could be stimulated by 'Social Fridays' with creative sessions to allow discussions of ethical concerns. Creating a corporate social agenda is a good way for ethical issues to be addressed throughout the organisation. According to Porter, categorizing and ranking social issues is just a means to an end, which is to create an explicit and affirmative corporate social agenda. A corporate social agenda looks beyond community expectations to opportunities to achieve social and economic benefits simultaneously. It moves from mitigating harm to finding ways to reinforce corporate strategy by advancing social conditions (Porter & Kramer, 2006, p7).

Nr	Connection between two Ss	Why a bottle neck?	Performance of the company	
			Who suffers from the bottleneck?	**Who is / are responsible for solving the problem?**
1	*Strategy and structure*	VSD may be knowledge intensive and resource demanding	Engineers, project managers	Management, project managers
2	*System and staff*	No (VSD/Moral Intelligence) information systems	Engineers, project managers	HRM, Management
3	*Staff and structure*	Lacking creativity	Project managers, teams, customers	Teams and managers can create a creative corporate culture

Figure 6.14 Qualitative interview insights related to 7s

6.6 Suggestion for a Moral Intelligence deployment framework

Now that problems within 7S have been exemplified, a well solution can be proposed for (1.3) The rejection of ethical concerns to speed up the project and (1.4) The creation of a hostile environment. Problems within the linkage between the 7S must be solved to stop the rejection of ethical concerns to speed up the project. To solve 1.4, a deployment framework may be developed. As explained earlier each organisational component as shown in the figure, may contribute to the organisational outcome. In other words to force a different outcome I suggest cultivating each dimension for the incorporation of VSD/Moral Intelligence. Important dimensions to cultivate are the factors described: organisational culture, philosophy, value, vision, mission, goals, leadership and group dynamics. 7s describes most of the social systems, intangible systems, formal systems and informal systems. The below figure describes the relation to Moral Intelligence and the suggested intervention to create a deployment framework for Moral Intelligence.



Use the below interventions to incorporate Moral Intelligence within an IT organisation.

Dimension	Relation to VSD/Moral Intelligence	Suggested Intervention
Shared value	These are the core values of the company that are evidenced in the corporate culture and the general work ethic.	A clear vision and mission that helps to steer VSD/Moral Intelligence integration *See also Chapter 7*
Systems	Develop systems that can provide the basis for capturing values, e.g. use envisioning cards, applications to gather information, like surveys or a repository and promote the overall learning, monitoring and reporting aspect of Moral Intelligence.	Build systems for the capturing framework (learning, monitoring, reporting), e.g. survey systems, feedback systems, interview systems, use of envisioning cards, 360 degree feedback or other VSD/Moral Intelligence components. *See also Chapter 4*
Structure	Provide the necessary structure to support active participation in VSD/Moral Intelligence. Structure can mean how departments within the organisation, roles, coordination, communication lines cooperate.	Create flexible teams, a flat organisation structure; allow VSD/Moral Intelligence reporting within the organisation and quickly incorporate changes e.g. with the usage of the Moral Intelligence disciplinary team matrix. *See also Chapter 6*
Style	Provide sessions to raise ethical questions within the organisation.	Incorporation within the strengths of Scrum and continuous integration may provide quality improvements within Sprints. If another development method is chosen plan extra meetings e.g. 'Social Fridays'. *See also Chapters 5/6*
Staff	Have staff that can work together, take leadership, and engage with ethical issues to support ethical reasoning staff development.	A disciplinary team matrix that stimulates collaboration and ethical investigation. *See also Chapter 6*
Skills	Ensure that the development team gets involved with VSD/Moral and develops skills.	VSD/Moral Intelligence training, workshops, supporting literature, online forum, repository *See also Chapters 5/6*
Strategy	Incorporation of VSD/Moral within methodology to develop better quality products, for example.	A quick deployment framework and incorporation within Scrum, MDD, and Continuous integration *See also Chapter 7*

Figure 6.15 Moral Intelligence deployment framework for IT organisations

Creating leadership and a Vision for Moral Intelligence for Organisations

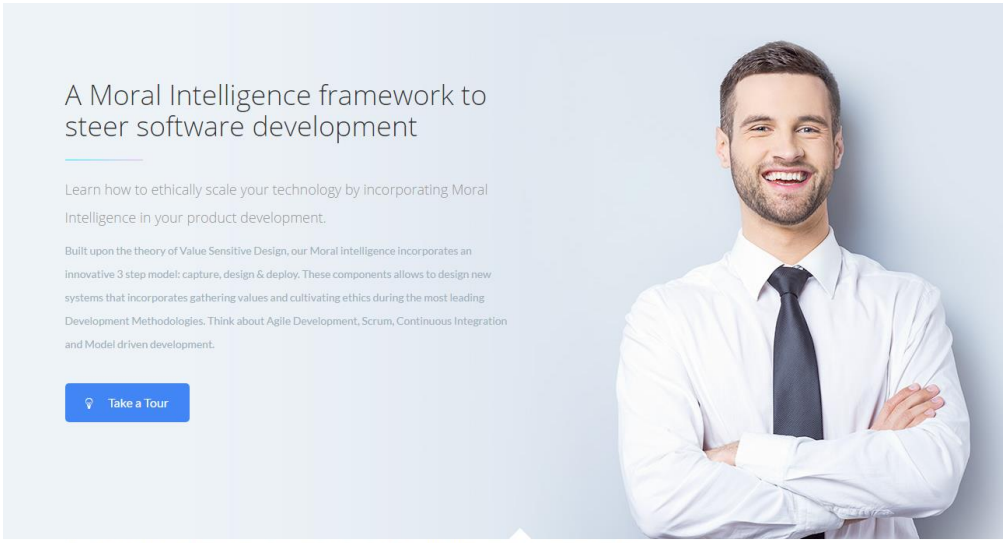
As a final suggestion for the incorporation of Moral Intelligence I suggest to also cultivate the factors philosophy, value, vision, mission, goals and leadership. These factors will be different for each company depending on their situation. However, Moral Intelligence as a concept also requires these factors if people want to actively participate in this. I suggest the creation of a Moral Intelligence website or an organisation/community to promote its concept. As per example, I have created a website which deals with organisational questions and that will describe and explain all the components that have been developed during this thesis.

The website, Ethics4industry.com, is a product of this thesis, which promotes the Moral Intelligence philosophy for IT businesses. The site may function as a blueprint example of how Moral Intelligence can be promoted and grow as a separate theory/philosophy, featuring reflection on technological impact and innovation using industry standard methodologies and practical tooling. Moreover, the ethical capturing framework that has been developed may be used as a source for inspiration to build new applications, such as a learning, monitoring, and reporting system (similar to Asana.com) to promote ethical intelligence.

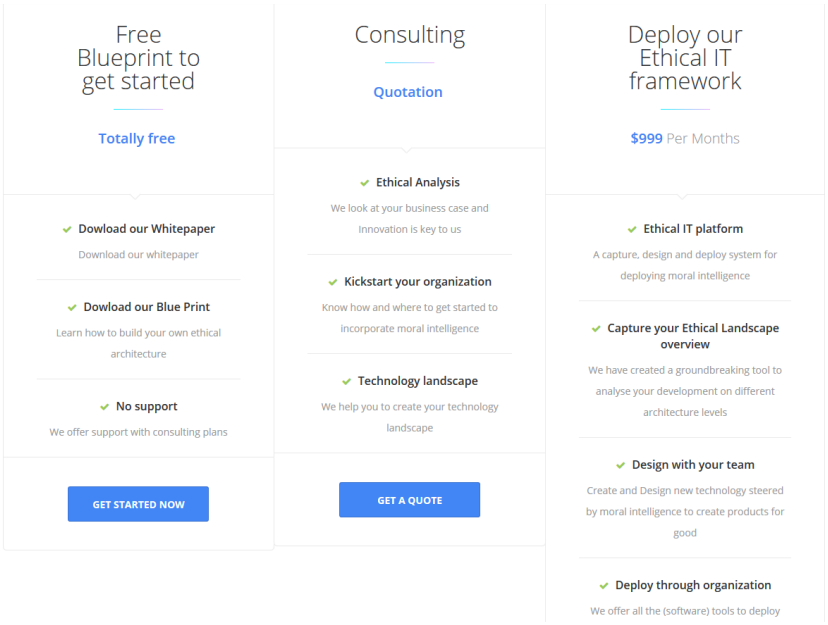
Figure 6.16 (Main banner of ethics4industry.com) Source: ethics4industry.com

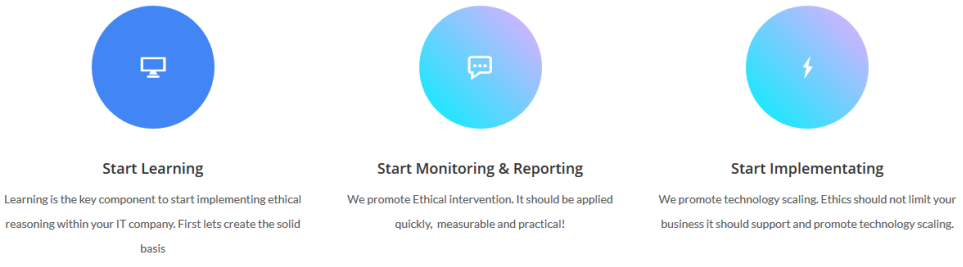
Moral Intelligence made attractive for IT business

The above image is the banner for the ethics4industry website. The idea is to create a novel and attractive good looking website which will attracts IT businesses to participate in Moral Intelligence. The right presentation is of great importance because companies that will participate will most likely be innovators. Therefore, a website for ethical innovation has been designed to communicate philosophy, value, vision, mission, goals and leadership. Below are more screenshots of the website:



Want to learn how to incorporate ethics within your IT company?

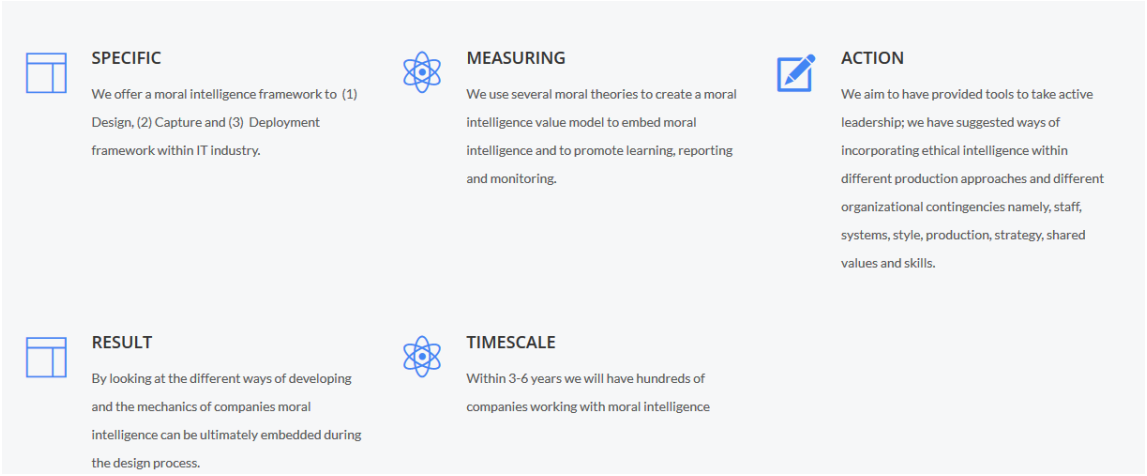




The website will feature downloadable content, business cases and ways to grow and explore IT landscapes. In the future ethical tools may be promoted from developers to support ethical intervention. Also a clear mission and vision has also been created to grow the Moral Intelligence philosophy. This mission and vision statement is pure indicative and may change over time when more people will join the website and the website will grow more towards a community for e.g. businesses, engineers, ethicists, consultants.

Our mission & vision

We [A] intend to provide the (IT) community with the methodology, production and practical tools from beginning to end during the design process with supporting literature, a searchable website, easy-to-follow instructions and tailored tools [S]. Further expansion of the 'integral vision' consolidates a framework for understanding, acknowledging and weaving together different perspectives and worldviews [M]. When applied to design [R] this kind of framework can help us to conceptualize how different values, systems and different onto-epistemological assumptions change our experience and therefore intentionality behind design. This change in why we design technology and processes, in turn, affects what and how we design to create a better future. [T]

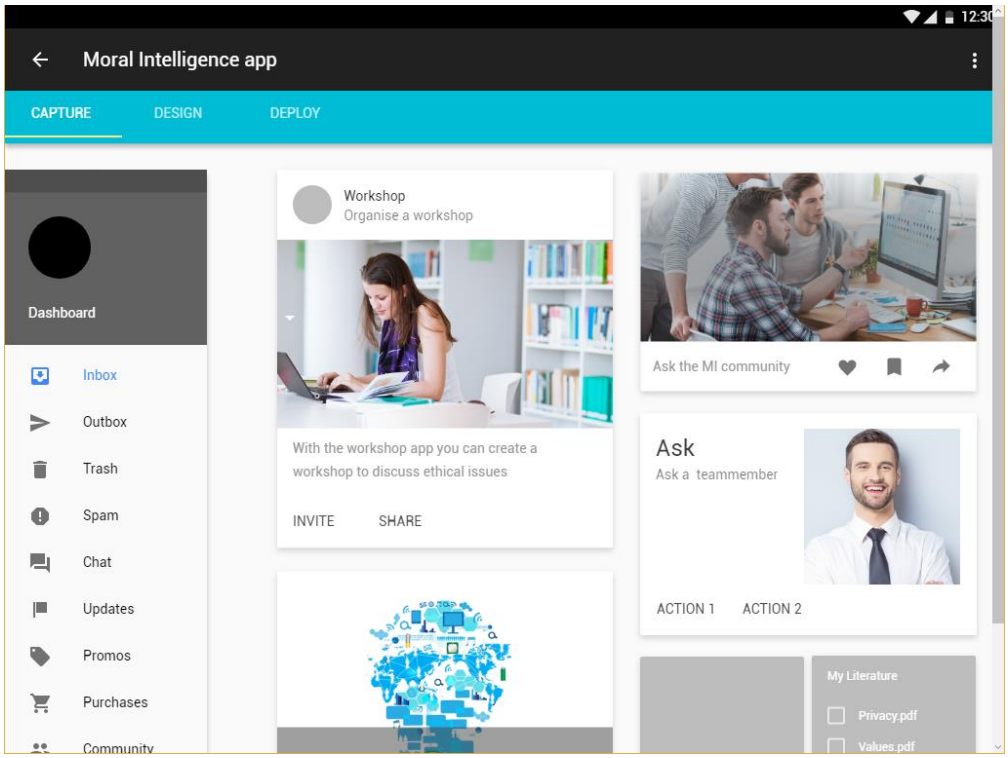


Distributing the Moral Intelligence vision can be done by creating an app (see next page) that incorporates the investigation methods as presented in Chapter 4. More methods may be added by the VSD community e.g. the 10 methods. A specific feature list may therefore be developed first.

(1) stakeholder analysis; (2) designer/stakeholder explicitly supported values; (3) co-evolution of technology and social structure; (4) value scenarios; (5) value sketches; (6) value-oriented semi-structured interview; (7) granular assessments of magnitude, scale, and proximity;(8) value-oriented coding manual; (9) value-oriented mock-ups, prototypes, and field deployments; (10) ethnography focused on values and technology; (11) model for informed consent online; (12) value dams and flows; (13) value sensitive action reflection model; and (14) envisioning cards.

Example: Moral Intelligence App for IT production

An example has already been given how to incorporate the envisioning cards in an app. The creators of the investigation methods can work together to build a smart learning, monitoring and reporting app which can be used during capturing, designing and deploying technology. The methods that have been suggested in the thesis can be put into an app which integrates within development and workflow of the team



Justification of the Moral Intelligence Framework in practical context

I suggest that values are evaluated with the use of iterative feedback due to the promotion of the learning, monitoring and reporting systems throughout the Moral Intelligence methodology. Therefore, values are not seen as given 'constant' value but as a changing concept of interpretations. Value conflict can be captured by agile learning methods. Once values are gathered it can be monitored, reported and evaluated with the designers (e.g. with use of the team matrix). A monitoring system ensures that features, issues, stakeholders can watch closely along the side line and adjust production.

The suggestion of an app is a practical example for translating Value for Design methods. This app may also incorporate the discussed 10 VSD methods in the future (Davis & Nathan, 2013). I suggest to take 'a lead by example stance' by creating more handbooks and practical tooling with the VSD community with e.g. the usage of new digital media such as apps, software and novel tooling to put VSD/Moral Intelligence at the forefront of technological development and innovation.

Industry roadmap and growing the community

Also a roadmap has been provided to set goals and to promote Moral Intelligence progress. A goal after this thesis would be to build a Moral Intelligence system (e.g. a similar application like asana.com) but tailored for gathering ethical data.

Source: http://www.ethics4industry.com/home/roadmap/

As a result of this Master's thesis and website, Saxion University has shown interest in a presentation and sharing information about Moral Intelligence and exploring how the Moral Intelligence framework would be useful for students to learn more about ethics.

Last, but not least, a quick-start guide has been proposed as a step-by-step case. The case illustrates how to use and incorporate Moral Intelligence from its ethical cultivation to development and finally deployment within an IT organisation (see appendix L).

CHAPTER 7

Conclusion

In this thesis, a separate theory of Moral Intelligence was proposed that could link ethics with VSD or be used as a separate theory. During the thesis, the foundation for these components has been carefully developed by investigating cases of VSD core-problems, e.g. what goes wrong with VSD in a practical context? The most essential problems have been taken into account to further build a new theory on.

The central research question was based on the following statement, describing VSD's core problem: "Perhaps the most apparent challenge of engaging in VSD outside the academic laboratory is the inevitability of confronting competing values within varied design contexts" (Zimmer & Manders-Huits, 2009,p6).

This led to the following research question:
How can Value Sensitive Design be extended towards the IT industry and be integrated in the design and production process of IT companies?

In order to answer this question, important philosophical and operational problems were first investigated. Research of contemporary literature led to the following sub-research questions, which are described in detail in the research design that charted the questions and ideas for important next steps to incorporate VSD (Borning & Muller, 2012; Davis & Nathan, 2013; Friedman, 1999; Grunwald, 2001; Lanzo, 2003; Maedche, 2017; Manders-Huits, 2011; Pesch, 2015; Spiekermann, 2015; van den Hoven et al., 2015; Zimmer & Manders-Huits, 2009).

Two important questions derived from this research:

1 How to operationalise VSD within a Corporate IT Setting?

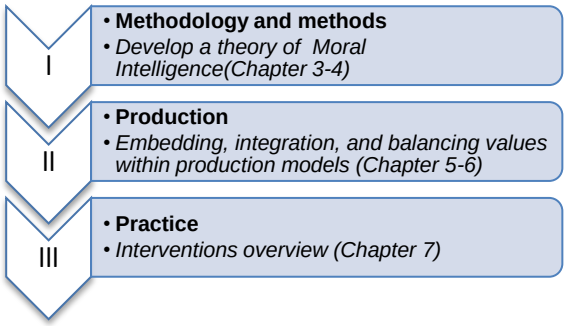
- Chapters 5 and 6

2 How to embed ethics in VSD and in production so that values can be operationalised?

- Chapters 3 and 4

These questions were formulated because of the sub problems that exist within VSD practice.
To build a stronger case, three companies from The Netherlands were interviewed to chart their current ethical investigations and the issues related to their practical challenges and the challenges addressed in literature.

At the beginning of the thesis, a research design was systematically proposed, investigating first the *methodology and methods*, then the *production context* and then the *practical context* to answer the research question and the sub questions.



I Methodology and Methods

Philosophical problems have been addressed first in the Chapter 3 and a methodology and methods were developed in Chapter 4.

How to embed ethics in VSD and in production so that values can be operationalised?

In order to answer the two questions, sub questions were formulated to investigate the philosophical and operational problems. Below are the answers to these sub questions.

2.1) How to embed an ethical approach in VSD and in production?

For ethical investigations, a Moral Intelligence framework has been suggested that contributes to VSD. The capture framework provides a scope for ethical investigation on different levels of technology and useful methods to gather data to gain a quick overview of ethical issues during development. Ultimately during the thesis, three pragmatics steps were developed to operationalise Moral Intelligence within organisational contexts.

“Step 1: Capturing ethical data”

A Moral Intelligence Capture component allows any person to start capturing ethical issues by choosing from a selection of capture methods to gather ethical data on different levels of technology inspired by Anticipatory Technology Ethics (P. A. E., Brey, 2012). Its major function is to allow learning and gathering of ethical data (intelligence) using several methods (Boenink et al., 2010; Brey, 2012; Palm & Hansson, 2006; Swierstra & Rip, 2007). More methods may be added iteratively, e.g. adding the foresight diamond or making use of traditional VSD methods.

“Step 2: Ethical reasoning throughout design and tooling”

A Moral Intelligence Design framework deals with the varied design disciplines within the IT industry and links this to gathering and the discussion of values and tooling. Its major function is to allow ethical monitoring during development and promoting iterative improvements. Additionally, a Moral Intelligence team matrix has been proposed to discuss and incorporate values throughout design. Concepts of shared responsibility have been the basis for this framework (Grunwald, 2001; Lanzo, 2003; Reddy, 2016).

“Step 3: Deployment throughout organisation”

A Moral Intelligence Deployment framework allows for the integration of Moral Intelligence within important corporate contingencies that may expedite the process of adopting Moral Intelligence within corporate cultures. The deployment framework has seven distinct systems that are related to the incorporation of Moral Intelligence within an industry context. Its major function is to report ethical improvements back to the organisation on all organisational levels (“McKinsey 7Ss Model | Developing Together,” 2015).

2.2) How to identify the role of the values advocate?

In the thesis, I argue that product decisions and values are ultimately shaped by the team. Therefore, not only an ethicist is taken into account, but also engineers, consultants, and how they can interact and provide input for an ethicist. Supportive conceptualisations of shared responsibility motivate this perspective (Grunwald, 2001; Lanzo, 2003; Reddy, 2016).

2.3) How to select the justification of a value framework?

The Moral Intelligence framework can be used as the justification for a value framework. Firstly, it uses ethical theories and different levels of technology to provide the scope of ethical investigations and the operationalisation of values. Of course, other value frameworks are possible, such as the Foresight Diamond. The Moral Intelligence framework is therefore intended as an iterative investigation model for ethical concerns at different levels of technology—from technology at large to the processes involved in building IT technologies.

2.4 How to confront competing values?

Competing values must be made transparent during development. Moral Intelligence focuses on *learning, monitoring, and reporting* values throughout the design process. Continuous Integration (CI) showed that learning and gathering data could be constantly promoted during development. Embedding values throughout design should not only be monitored but also embedded throughout design, e.g. during MDD or agile development. This means that during MI Capture, MI Design, and MI deploy, the processes of learning, monitoring, and reporting also take place, via the disciplinary team matrix, reporting tools for which examples have been given, or the Moral Intelligence and community tools (Chapter 6).

II Production Context

After the investigation of methodology and methods of Moral Intelligence, it was time to operationalise the framework within the industry context. For the industry context design, methodology and organisational contingencies have been carefully investigated (e.g. roles, teams, systems, staff, strategy, skills, shared values, structure).

The below questions describe problems within the industrial context and the incorporation of ethics. The answers are carefully given to these questions during the thesis by also taking practical cases into account of IT companies (e.g. InterDC, CAPE Groep BV BV, Elnino-ICT).

1.1) How can we solve the difficulty to properly explicate and translate ethical considerations to workable requirements and specifications for the other project participants actually building the system?

This question firstly involved scrutiny of development methods, such as MDD or Agile development methods at companies situated in Enschede. Jeroen van der Hoven stated “One important area of research could be to design and evaluate” agile methodology and methods for value-sensitive design (Maedche, 2017). While Sarah Spiekermann mostly spoke of the commonly known waterfall model, there are also many other to be considered (Spiekermann, 2015,p160). From their best parts, a three-step development methodology was proposed: Capture, Design and Deploy. Ethical considerations are taken into account during the capture phase, as explained earlier. Different design disciplines may use the capture framework to define an ethical scope and make use of the overarching distributions of roles to implement requirements within their own development method, as shown with the MDD example in Chapter 6.3.

III practice

1.2) How to solve the problem of operationalising values so they can be incorporated into technological design?

I have suggested several community tools and a website to promote (ethical) values so they can be added into technology design in Chapter 6. Asana.com is an example of how to create quick dialogue within a team through feedback provided by team members.

This supports the perspective of Sarah Spiekermann: “For people to thrust machines and machine operations, data and information must be collected in a legitimate way” (Spiekermann, 2015). The collaborative distribution of roles (team matrix) involved in a team could provide support for operationalising values so they can be added into design.

1.3) How to solve the creation of a hostile environment?

For this problem, I suggested that Moral Intelligence should focus on creating a less hostile environment by the operationalisation of more roles and better ways to work together. The team matrix is an example of creating collaborative values in domains of knowledge. A concrete, more holistic deployment framework has been proposed to look at the most essential parts of a hostile environment, which is “corporate culture” defined as 7S contingencies (“McKinsey 7Ss Model | Developing Together,” 2015). The deployment framework of 7S ensure interventions are in place to promote a less hostile environment.

1.4) How to solve the rejection of ethical concerns to speed up the project?

The full concrete deployment framework in Chapter 7, which also takes 7S into account, deals with corporate culture, such as a hostile environment and ethical concerns, to expedite the project. Because Moral Intelligence is embodied throughout the company, ethical concerns become key to a project, making it too important to reject when a project has begun. Here, using Moral Intelligence products could reduce ethical conflict.

Ethical building blocks for innovation as a thesis result

During this thesis, Moral Intelligence has been the building block for innovation, not only by looking at the different levels of technology but also by incorporating ethical theories and designing a learning, monitoring, and reporting system. The deployment framework allows companies to intervene and implement the Moral Intelligence philosophy. The concepts and practical tools that have been showcased could catalyse the development of ‘intelligent’ moral systems. Moral Intelligence focuses on community and contributions, not only the contribution by researchers, but also from engineers, consultants, and CEOs.

With the right extension of Moral Intelligence as a platform for technical and ethical innovation, Moral Intelligence could grow into something big, adopted possibly by technical firms and bigger companies such as Facebook, Microsoft, Google. In the next section, future research is suggested to grow the Moral Intelligence concept further.

Future Research

During the thesis, ethics4industry.com was proposed to promote the concept of Moral Intelligence and to inspire researchers to participate and to further expand the creation of Moral Intelligence methods. Several important blueprints were presented—the Moral Intelligence framework, the disciplinary team matrix, and knowledge of industrial tools, such as project management and reporting software, Asana.com.

Fundamentally, to take active leadership in implementing Moral Intelligence, the pragmatic three-step model has been proposed on the ethics4industry.com website describing the iterative Capture (1), Design (2) & Deploy (3) model. This model describes all the fundamental processes that Moral Intelligence can face and these methods may be extended by other contributors.

Further development of Moral Intelligence involves building a community and attracting researchers, engineers, consultants, and business people (CEOs, business consultants) to incorporate new innovative solutions that will benefit the IT industry. It is hardly a discussion that our society needs these systems, and that we can build new convenient ethical systems together for the IT industry. To speed up innovation and contributions by others, a road map has been published[14].

Building a concrete Moral Intelligence system for IT corporations

With the help of the "Moral Intelligence framework" that was developed during this thesis, a software application may be developed in the future to capture data on different levels of technology. Just like a business model canvas[15], the "Moral Intelligence framework" allows for a quick mapping and investigation of the ethical situation within a company. When implementing this new software, a set of ethical requirements could be specified and the three organisations (InterDC, CAPE Groep BV BV, Elnino ICT BV) who have been involved during the thesis could participate in development to create a Moral Intelligence tool. MIT subsidies can also provide funding for building this software application ("Mkb-innovatiestimulering Regio en Topsectoren (MIT) | RVO.nl," n.d.).

Data analysis and quantitative research

The Moral Intelligence System is designed to quickly assess an organisation for its ethical and technology landscape. From design to end-product, future research and development could be performed with the use of the Moral Intelligence framework within a research lab or corporation. Performance and outcomes of the ethical system could also be measured. A quantitative research model may be developed in future studies to measure the performance of the Moral Intelligence system and provide confirming statistics that the Moral Intelligence framework is a great solution or intervention for society, IT, and engineering companies within the technology creation domain.

Distributing the Moral Intelligence Platform

Several meetings and lectures could be planned with universities. The results of the thesis have been noticed by Saxion and a presentation of this thesis has been scheduled for October 2017.

Offering a downloadable quick-start guide

Ethics4industry.com may present a downloadable "quick-start guide" to promote the Moral Intelligence framework vision. This also would attract new researchers or IT companies that may want to be involved in extending the Moral Intelligence framework. Sharing knowledge and providing practical tools will boost the distribution of Moral Intelligence material for the IT industry.

Community tools and Participation

Ethics4industry.com may attract investors, entrepreneurs, tech companies, etc. that would like to team up to build new innovative systems or further extend the Moral Intelligence Community app (as seen in Chapter 6.4). Moreover, building a Moral Intelligence community for ethics4industry.com is a next step. The strong mission, vision, and roadmap may create leadership so organisations can implement Moral Intelligence within their workplaces and workflow. A conceptual mission and vision can be found in Appendix I.

14 http://www.ethics4industry.com/home/roadmap/

15 https://strategyzer.com/canvas/business-model-canvas

Afterword

Writing this thesis has been a wild ride!

There are so many things that could have been added here or in the future. My dream was to connect people of various backgrounds with technology, its consequences for society, and to inspire people to look differently at various organisations such as in terms of their motivations and corporate vision. We all take part in this ‘wild’ journey of technology creation. I would therefore like to conclude this work with the following:

We’re here on this planet as individuals for a limited time, so let’s make sure that the future of humanity will progress in harmony by ‘connecting the dots’ that involve engineering, teaching ethics, and shaping society. I hope this thesis will inspire people to connect with each other to subsequently advance even further with technology creation and its innovation process.

Appendix

Appendix A Questions asked during the interviews

Description: To find out what contingencies may be useful for IT companies to practice VSD, simple questions were asked regarding their views on ethics and VSD. The questions are stated below.

Interview
1. What are the potential areas of including ethics/VSD in your IT company?
2. What are the positive and negative effects of doing ethics/VSD on different IT development economics in your company?
3. What are the main decision-making aspects when deciding between the incorporation of ethics/VSD or choosing none to be adopted in production?
4. What are the general implementing problems/challenges (let it be /managers, staff, developers) faced when implementing ethics/VSD or an ethical justified methodology?

The below interviews were used during Chapters 5 and 6. It gave interesting insight into important values of companies and their approach when dealing with ethics.

Interview Summary, Roland Kamphuis, CEO, InterDC, Enschede (Data Warehouse –small-sized company)

1. What are the potential areas of including ethics/VSD in your IT company?

InterDC acknowledges the importance of keeping the data of their customers safe and the CEO stresses the importance of data protection for both the digital and physical world (e.g. the physical servers). “It is important that the personal data does not become publicly known and cannot be found freely,” Roland Kamphuis mentions. Moreover, ethics in the sense of knowing what to do in specific cases, would help the Datacentre/Software development company to steer development, such as providing guidelines to safely use scripts and to explicate how customers could behave more responsibly. Also, InterDC mentions the delegation of this responsibility and creating awareness is important. Important questions that are raised are, for example:

What does software do for your surroundings? How do you control the flow of your information? The CEO mentions that he would have great benefit in creating an operational guide for escalations and how you create safety or applications that are secure and unhackable; those questions are raised within his company.

2. What are the positive and negative effects of doing ethics/VSD on different IT development economics in your company?

For InterDC, good ethical software development costs a lot of money, time, and investment, which are seen as negative aspects. On the positive side, marketing and corporate social responsibility can boost the quality of the company and also contribute to creating a better environment (e.g. having less waste or a reduced CO^2 footprint)

3. What are the main decision-making aspects when deciding between incorporation of ethics/VSD or choosing none to be adopted in production?

Roland mentions that they have to be sure that the cost will be covered and that a return of investment is the most important aspect of investing.

4. What are the general implementing problems/challenges (let it be managers, staff, developers) faced when implementing ethics/VSD or an ethical justified methodology?

Roland mentions that plans and specifications can contradict the corporate process and original production. For example, when you implement a measure/intervention for safety, Roland has to be aware if the new process could also affect other processes. An example is protecting the data centre with loud alarms; although the sound raises awareness that something has gone wrong, the frequency of the sound and its vibrations causes materials and the server hard disks to vibrate, which could damage stored data. In this case, it is a business process vs. a safety trade-off. Another example is the usability, which is important for customers and prioritised before ethics. Customers intend to find software features more important than the ethical implications; Roland mentions that

those are often found when something goes wrong (and not analysed in the prospective sense). Scalability of technology and maintenance are also seen as problematic.

Interview Summary, Michael, CEO, Elnin-ICT, Enschede (mid-sized software company)

1. What are the potential areas of including ethics/VSD in your IT company?

Michael mentions that security is most essential for his company. They have to encrypt their parties' hosting data, and they should save passwords accordingly. Also, keeping private information private e.g. employees may not leak private data, is crucial. Their most important stakeholders are customers and partners and they have to make sure that their knowledge is also not distributed to competitors or any other parties. Michael mentions when you have a third party hosting company, it is essential to be fair towards customers. If this hosting party is doing unusual activities but you still get a high commission then this is also not very ethical.

2. What are the positive and negative effects of doing ethics/VSD on different IT development economics in your company?

I think, in general, if you are unethical, it is easier to make more money but in the long-term, this is not efficient and you will lose customers. In this way, it is better to be ethical.

When using custom work and open source, the price differs. If a customer wants a blog site, we can use Wordpress or we can tell you need a custom blog for €10.000. We are very transparent about this and we do not sell a product for more hours. We think a long-term investment is more important over a short one in this sense.

3. What are the main decision-making aspects when deciding between incorporation of ethics/VSD or choosing none to be adopted in production?

For us, building relationships in the long term is most important besides economic factors.

4. What are the general implementing problems/challenges (let it be managers, staff, developers) faced when implementing ethics/VSD or an ethical justified methodology

It is important to train employees to be safe with data and specify categories of knowledge. (What is important for our customers, what can be dangerous etc).
For the implementation of a VSD methodology, we will check if it is easy to apply, quick, measurable, practical; the customer has to see that it is useful, and be able to quickly to adapt. Michael mentions that an ecommerce system (e.g. online store) is different than a sales software program. Within design, different information is required per software solution.

Besides a checklist, a good test would be useful; usage with stakeholders, or a monitoring system this can benefit the IT world. Michael mentions the possibility of VSD to be integrated within Continuous Integration. Each time code is committed, you can analyse your code and install plug-ins, and the server will check the code for any unsafe functions. It will give errors and indicate which files need to be checked. Michael mentions that it would be possible to develop multiple VSD systems that would fit within the CI method.

Appendix B Company Contact information

Description: all business cases were taken from companies situated in Enschede, The Netherlands. Their contact information can be found in the below overview.

	CAPE Groep BV used for MDD highlights, Medix case Chapters 5,6,7 **Contact details** Transportcentrum 14 7547 RW Enschede Tel: +31882273477 E-mail: info@CAPE Groep BV.nl
	El Niño-ICT, used for Qualitative Interview and Continuous Integration case Chapters 3,5 **Contact details** Adres Roomweg 100 7523BS Enschede Tel: + 31 53 820 02 85 E-mail: adresinfo@elnino-ict.com Skype ELNINO-ICT
	InterDC, Used for Qualitative Interview Chapters 3,5 **Contact details** InterDC Josinkstraat 30 7547AB Enschede Nederland E-mail: info@interdc.nl Tel:+31 (0)53 700 9757 Fax: +31 (0)84 747 3294
	TriMM, used for Scrum case Chapter 5 **Contact details** Moutlaan 25 7523 MC Enschede Mollerusstraat 1 3743 BW Baarn Tel: 053 – 48 00 48

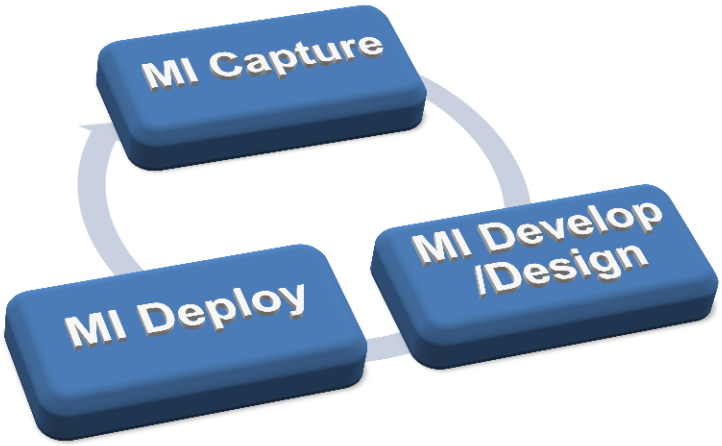
Appendix C Step 1: Capture Framework Blueprint

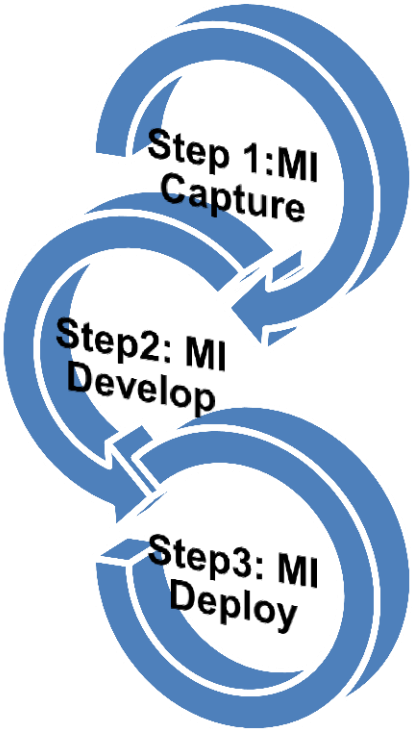
Instruction: this table was used as a result of the research in Chapter 4, the frameworks describes to capture ethical data on different levels of technology.

Class →	***Object Analysis →***	***Procedures Analysis →***	**Design Process Analysis**
Technology creation level(ETC)	Empirical, Technical, Conceptual (ETC)	Empirical, Technical, Conceptual (ETC)	(Practical) Design analysis
How?	**How?**	**How?**	**How?**
Engineers consult ethicist and vice versa for forecasting. *- ethical analysis focuses on features of the technology at large, particular subclasses of it, or techniques within it.*	*-Extended ethical checklist* *-NEST ethics* *- scenario sketching, define larger scopes.* *-Expert survey* *-Utilise existing studies in forecasting and TA about the technology, to the extent that these are available*	*-Future assessment e.g. with workshops, evaluation, consulting TA experts historians, sociologists, and marketing experts, policy documents, company studies, academic texts*. *-Analyse application and system variables*	*-Integrate ETC in production methodologies and tooling* *-Link ETC to process and design methods* *-Analyse by using the 7s model (style, system, structure skill, etc.)*
Ethical analysis focuses on	**Ethical analysis focuses on**	**Ethical analysis focuses on**	**Ethical analysis focuses on**
Features of the technology at large (the defining feature of the technology)	Types of artefacts and processes that result from the technology	Concrete use of the application	The (to be)designed processes and used production methodologies

Appendix D Step 2: Design Framework Blueprint

Instruction: the below graphs show the iterative development cycle of moral intelligence developed during the Chapters 5 and 6.





Appendix E VSD/Moral Intelligence Team Matrix Roles

Instruction: the below table is the result of Chapter 5, a framework to collaborate and to gather ethical values by a development team.

(Shared) responsibility		
Multiple responsibility	Capable responsibility	Special responsibility
1 **Value Discovery Ethicist**	**Engineer /programmer**	**Role Consultant**
A. Making explicit the intended values as conceived by engineers, designers or consultant.	Provide input & arguments for the intended values together with the ethicist.	Provide input & arguments for the intended values.
B. Scrutinising these values by illustrating value conflicts and trade-offs.	Provide input e.g. using the Moral Intelligence framework together with the ethicist.	Provide input e.g. using the Moral Intelligence framework together with the ethicist.
C. Comparing intended values with use of Moral Intelligence framework, e.g. ethical literature, workshops or value for design methods , etc.	Make a priority list of the intended values and ethical participation in workshops, brainstorms sessions.	Make a priority list of the intended values and ethical participation in workshops, brainstorms sessions.

2 Translating Values into Design Requirements Ethicist	**Engineer/Programmer**	**Consultant**
A. Value conceptualisation (Customers, Public)	Gathering, providing feedback for the ethicists and consultant e.g. with the use of Moral Intelligence.	Value conceptualisation with customers and stakeholders (liaison role).
B. Describing the disconnect between intended and realised values by speculating unintended uses and contexts of the artefact.	Gathering (realised) value concepts given by ethicist or IT consultants and advising which engineering concepts may be suitable.	Mediate between the engineer and ethicists to find the best possible solution between intended and realised values e.g. within the role of a product owner (as an advisor to the ethicist)
C. Exploring the translation/specification of values into design requirements.	Exploring and realising the best possible building blocks for the implementation of values.	Exploring the translation/specification of values into implementation.

Appendix F Step 3: Deployment Framework Blueprint

Instruction: the deployment framework was developed in Chapter 7. Use the below interventions to incorporate Moral Intelligence within an IT organisation at all organizational levels based on 7S.

Dimension	Relation to VSD/Moral Intelligence	Suggested Intervention
Shared value	These are the core values of the company that are evidenced in the corporate culture and the general work ethic.	A clear vision and mission that helps to steer VSD/Moral Intelligence integration *See also Chapter 7*
Systems	Develop systems that can provide the basis for capturing values, e.g. use envisioning cards, applications to gather information, like surveys or a repository and promote the overall learning, monitoring and reporting aspect of Moral Intelligence.	Build systems for the capturing framework (learning, monitoring, reporting), e.g. survey systems, feedback systems, interview systems, use of envisioning cards, 360 degree feedback or other VSD/Moral Intelligence components. *See also Chapter 4*
Structure	Provide the necessary structure to support active participation in VSD/Moral Intelligence. Structure can mean how levels within the organisation, roles, coordination, communication lines cooperate.	Create flexible teams, a flat organisation structure; allow VSD/Moral Intelligence reporting within the organisation and quickly incorporate changes e.g. with the usage of the Moral Intelligence disciplinary team matrix. *See also Chapter 6*
Style	Provide sessions to raise ethical questions within the organisation.	Incorporation within the strengths of Scrum and continuous integration may provide quality improvements within Sprints. If another development method is chosen plan extra meetings. *See also Chapters 5/6*
Staff	Have staff that can work together, take leadership, and engage with ethical issues to support ethical reasoning staff development.	A disciplinary team matrix that stimulates collaboration and ethical investigation. *See also Chapter 6*
Skills	Ensure that the development team gets involved with VSD/Moral and develops skills.	VSD/Moral Intelligence training, workshops, supporting literature, online forum, repository *See also Chapters 5/6*
Strategy	Incorporation of VSD/Moral within methodology to develop better quality products, for example.	A quick deployment framework and incorporation within Scrum, MDD, and Continuous integration *See also Chapter 7*

Appendix G Ethical Checklist

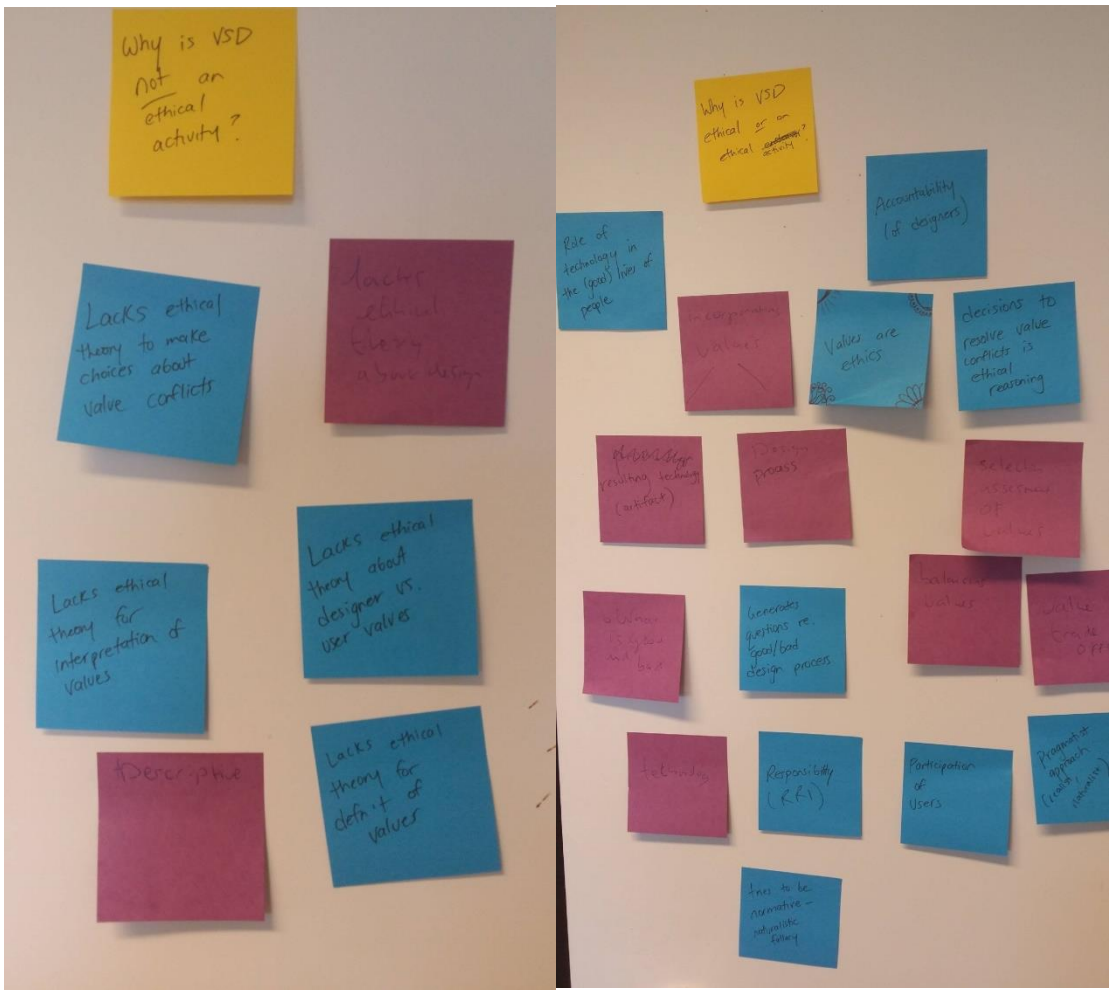
Instruction: this is an ethical checklist that can be used to start investigating values. This Ethical checklist can be used with the moral intelligence capturing framework.

Ethical Checklist

- Harms and risks
    - Health and bodily harm
    - Pain and suffering
    - Psychological harm
    - Harm to human capabilities
    - Environmental harm
    - Harms to society
- Rights
    - Freedom
        - Freedom of movement
        - Freedom of speech and expression
        - Freedom of assembly
    - Autonomy
        - Ability to think one's own thoughts and form one's own opinions
        - Ability to make one's own choices
        - Responsibility and accountability
        - Informed consent
    - Human dignity
    - Privacy
        - Information privacy
        - Bodily privacy
        - Relational privacy
    - Property
        - Right to property
        - Intellectual property rights
    - Other basic human rights as specified in human rights declarations (e.g., to life, to have a fair trial, to vote, to receive an education, to pursue happiness, to seek asylum, to engage in peaceful protest, to practice one's religion, to work for anyone, to have a family, etc.)
    - Animal rights and animal welfare
- Justice (distributive)
    - Just distribution of primary goods, capabilities, risks, and hazards
    - Non-discrimination and equal treatment relative to age, gender, sexual orientation, social class, race, ethnicity, religion, disability, etc.
    - North-south justice
    - Inter-generational justice
    - Social inclusion
- Well-being and the common good
    - Supportive of happiness, health, knowledge, wisdom, virtue, friendship, trust, achievement, desire-fulfilment, and transcendent meaning
    - Supportive of vital social institutions and structures
    - Supportive of democracy and democratic institutions
    - Supportive of culture and cultural diversity

Appendix H Brainstorm Session

Description: this is an overview of a brainstorm session with dr. A. van Wynsberghe to chart existing challenges, discussions that surround Value Sensitive Design.



Appendix I Mission and for Vision Moral Intelligence

Description: The mission and vision that has been developed has been stated below to further expand the Moral Intelligence concept on ethics4industry.com

Ethics4Industry mission and vision (SMART analysis)
We **[A]** *intend to provide the (IT) community with the methodology, production, and practical tools from beginning to end during the design process with supporting literature, a searchable website, easy-to-follow instructions, and tailored tools* **[S]**. *Further expansion of the 'integral vision of Moral Intelligence' consolidates a framework for understanding, acknowledging, and weaving together different perspectives and worldviews* **[M]**. *When applied to design* **[R]**, *this kind of framework can help us conceptualise how different values, systems, and onto-epistemological assumptions change our experience intentionality behind design. This change is why we design technology and processes and it affects what and how we design to create a better future* **[T]**.

Below are the questions that have been formulated to SMART (**Specific, Measurable, Achievable**, **Relevant,** and **Time-bound)** for this research. The questions form the mission and vision.

- Which specific tools should be further developed? [S]
- What (theory) should be justified/ or measured [M]
- Who takes active leadership and responsibility? [A]
- How can results be realistically put into relevant practice? E.g. what skills are required? [R]
- What is the timescale of the intervention [T]

Below are the interventions that have been formulated to answer SMART (**Specific, Measurable, Achievable**, **Relevant,** and **Time-bound)** for this research.

[Specific]
In this thesis, we proposed a (1) MI Design, (2) MI Capture and (3) MI Deployment framework for VSD within the IT industry.

[Measurable]
The MI Capture framework may be extended to incorporate measurement and quantitative research. A Concept has been provided for this in Appendix J.

[Actionable]
We aim to provide tools to take active leadership; we have suggested ways of incorporating Moral Intelligence within different production approaches and different organisational contingencies, namely, staff, systems, style, production, strategy, shared values, and skills. Moreover, we have provided a mission and vision of a basic approach for extending Moral Intelligence towards the IT industry for anyone who would participate in the VSD approach, e.g. programmers, consultants, or ethicists. Last, but not least, we have made suggestions to further build a Moral Intelligence community.

[Relevant]
By looking at the different ways of developing and the mechanics of companies, VSD/Moral Intelligence can be ultimately embedded during the design process.

[Time-bound]
The timescale of the intervention is unclear, however, the thesis was focused on developing a long-term strategy.

Appendix J Quantitative research

Description: this is a quantitative research concept for Quantitative investigation of the performance of Moral Intelligence

Step 1
I conceptualise the ethical performance as the success of adoption of Moral Intelligence within the corporate IT setting. The adoption is the use and reuse of Moral Intelligence rules and interventions within an organisation.

Step 2
I conceptualise the measurements, dependent, and independent variables. In-depth literature research will be required. **The adoption/reuse of Moral Intelligence is the dependent variable**

I state that adoption is important within strategy, leadership, product development, and protocols. Because of the iterative approach, I conceptualise Moral Intelligence as a reusable component.

Adoption of ethics and values is the dependent variable

- Reuse of Moral Intelligence in Strategy
- Reuse of Moral Intelligence in Leadership
- Reuse of Moral Intelligence in Protocols
- Reuse of Moral Intelligence in Product development.

Methodology (Independent Variable)

The independent variable is the chosen methodology itself. This can be, for example, Scrum, CI, MDD with usage of the capture, design, and deploy system.

Quantitative Measures in a System

I state the following quantitative measures. Literature should confirm if these or any other quantitative measures are also important.

Product Development

- Moral Intelligence during design:
  Size or saturation (e.g. ethical rules within design or code, # functions, protocols
- Moral Intelligence Reuse rate:
  Reuse rate = Reuse (reuse size)/ System Size
- Training:
  Amount of ethical training given

Development effort (#hours)

- Measure development hours., incl. analysing, designing, implementing the system.

Strategy

Adoption approach

- Systematic
- Ad-hoc

Leadership

Company-related measures

- Company review scores (e.g. scores of corporate culture)
- Productivity incl. verbatim code (Before Moral Intelligence intervention)
  Productivity (System S) = Size (S) /Development effort (S)
- Productivity excl. verbatim code (After Moral Intelligence intervention)

Quality in terms of #bad reviews

- # Reporting manifestations in products
- # Errors

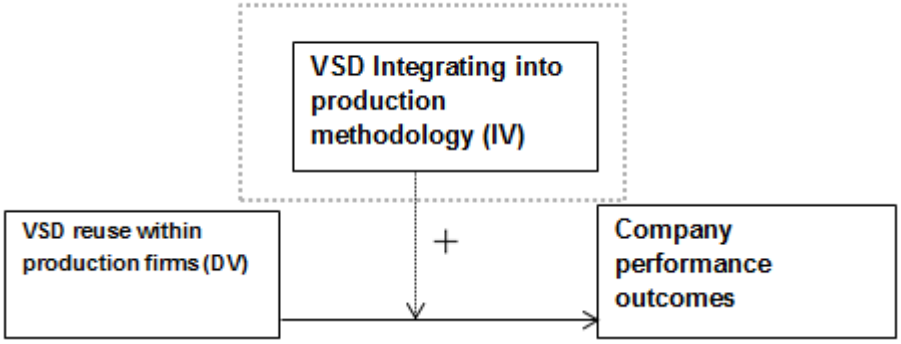
Rework effort (# development hours)
The amount of rework that needs to be done after the intervention.

Company score
#financial score

Control variables

- Company size
- Marketing budget ($/€)
- System size (# of components)
- Experience (# years)

The above measures show how ethical interventions can change ethical performance.



VSD/Moral Intelligence.

The production methodology is the independent variable and describes the methods within Moral Intelligence that we want iterate. We then want to know the outcome depending on the situation of the company, which is the dependent variable. This quantitative model helps us to understand the situations of Moral Intelligence on the component level. Specifically, it gives us a better insight into the company performance after the Moral Intelligence implementation. By measuring these scores under different situations, we may know how Moral Intelligence can be operationalised in terms of performance. After this is clear, ethical interventions may prove that Moral Intelligence allows for less rework, fewer errors in products, or will lead to any other interesting results, e.g. how the business performs and what ethical interventions could be interesting by measurement of the performance outcomes. This research takes a scientific approach and has to be translated to a practical Moral Intelligence monitoring system anyone can use.

Appendix K Introduction to Contemporary Ethical Theory

This extra chapter addresses a variety of ethical theories and their applicability for addressing issues in design. The goal of this chapter is to provide the general knowledge of different ways of looking at ethical issues and its conceptualizations. Several distinct ethical theories have embraced moral notions as central categories and emphasizing different conceptual and moral analyses. We explain them first here -step by step- and reflect on them in the concluding part of this chapter to find out which theories may be suitable for a practical context within IT industry.

1.1 Our starting point from Contemporary Ethics

In short, ethics deals with questions of the good and bad, right and wrong. These questions are asked in relation to the good life. One may ask “What is the right thing to do?” but also “What is the best way to live?” The first is a question for ethical reasoning specifically, while the second is for practical reasoning about one's life in more general: it questions what to do with one's life and how to make it a happy one?

Answering these questions involves reflecting on what sort of person to be and what sort of character to develop. And it is here that practical reasoning leads to thought about the virtues, excellences of character that consist in both caring about the right sorts of things and having the wisdom and practical skills to judge and act successfully. In general there are four branches of ethics generally recognized by philosophers: meta-ethics, descriptive ethics, normative ethics and applied ethics.

1.2 Introduction to meta-ethics

Meta-ethics is the attempt to understand the metaphysical, epistemological, semantic, and psychological, presuppositions and commitments of moral thought, talk, and practice (“Metaethics -Stanford Encyclopedia of Philosophy,” 2007).In context with dealing with good , questions can be raised such as:

- What is good? What does good mean?
- How can we tell what is good from what is bad?
- What is the meaning of these moral statements? Can they be true or false?

These questions aim to step back from assumptions, views or assumptions that are shared by those who engage in the debate. In other words, at the meta-ethics level it is conceptualized what the properties of good can be. As such, it counts within the design domain a broad range of questions and puzzles, including: is morality in relation to technology design more a matter of taste than truth? Are moral standards culturally relative? Are there moral facts? If there are moral facts, what is their origin? How is it that they set an appropriate standard for our behaviour?

1.2.1 Examples of meta-ethics and IT Industry examples

As explained here meta-ethics deals with what good can be. In relation to design on the meta-level questions can be raised regarding good design. E.g. What is good design? What does good design mean? How can we tell what is good design and what is bad design? What is the meaning of good design and can it be true or false? At the meta-level it is conceptualized what the properties of good design can be.

1.3. Introduction to Normative ethics

Normative ethics examines standards for the rightness and wrongness of actions. It examines the rules and principles that guide our behaviour. Normative ethics develops general moral theories (e.g. Consequentialism, Act-Utilitarianism, Deontology ethics, Virtue ethics). These theories are explained below.

1.3.1 Introduction to Consequentialism

Consequentialism is a theory of normative ethics. It holds that an act is only moral or ethical if it results in a good conclusion. This is in contrast to deontology, which teaches morality is based on duty; virtue ethics, which holds that morality is based on a good character; and ethical relativism, which asserts morality is based on whatever you want it to be based on. Consequentialism is the view that the right act is entirely determined by its consequences - the act that promotes the most good is the right act, and vice versa.

Plain Consequentialism: Of all the things a person might do at any given moment, the morally right action is the one with the best overall consequences (Haines, 2015). (If there is no one best action because several actions are tied for best consequences, then of course any of those several actions would be right.)

Consequentialism is based on two main principles:

- Whether an act is right or wrong depends only on the results of that act
- The more good consequences an act produces, the better or more right that act

1.3.2 Introduction to Act-Utilitarianism

Utilitarianism's starting point is that we all attempt to seek happiness and avoid pain, and therefore our moral focus ought to centre on maximizing happiness (or, human flourishing generally) and minimizing pain for the greatest number of people. In addition according to utilitarianism the right actions, laws or policies are those that promote the greatest amount of pleasure, or the least amount of pain, for all concerned. Classical utilitarianism's two most influential contributors are Jeremy Bentham and John Stuart Mill. Bentham, who takes happiness as the measure for utility, says, "it is the greatest happiness of the greatest number that is the measure of right and wrong".

In Act-Utilitarianism ethics the perspective focuses not on who's doing it but rather on the consequences that the act brings about. In this perspective the degree of moral rightness of an act is determined by the utility it produces and the final sum, the usefulness of happiness. Often general arguments in technological development are that new technology will bring us all kinds of good, because technologies have done so in the past, even if it may lead to some negative effects for some minor individuals. This means that there is an optimistic view that technological progress is basically beneficial. Consequentialist ethics measures the goodness of an action in terms of its effects, relative to some value ('happiness' in classic utilitarianism).

Consequently, it is really an outcomes-based ethic. It does not involve movement towards an end, but a maximization of a value. Because the perspective focuses not on the actor is doing it but rather on the consequence there is room for impartiality. Impartiality is a principle of justice holding that decisions should be based on objective criteria, rather than on the basis of bias, prejudice, or preferring the benefit to one person over another for improper reasons.

- Focuses on the nature or consequences of the act (or omission) itself.
- Means maximize happiness regardless of the character of the behaviour

1.3.2.1 Examples of Act-Utilitarianism and Industry examples

An important discussion within the technology creation domain is whether technology is beneficial. Act-Utilitarianism is often seen as a cost-benefit evaluation of ethical issues. For example, if the final sum accounts for more happy people then the technology is considered as good or the best way to go. An important question to ask is e.g. if a technology has an overall benefit then this is considered as good even with the costs of some individuals. When developers build new features into Facebook and the majority of the people that use Facebook vote that this new feature is beneficial, then from a consequentialist perspective this is considered as good at the cost of some individuals (because the final sum of happiness is most important).

The Greatest Happiness principle in general is good, but it has many flaws as any ethical system does. Due to our inability to perfectly predict the future according to our actions (assuming he future is capable of being altered with our actions), the results we desire are capable of, and often do, fall short of what was intended.

1.3.3 Introduction to Deontology ethics?

In Deontology ethics the perspective focuses the intention on the act, not on who is doing it but what act is done and under what moral description. Deontological arguments (i.e. right- and duty based) are expected to be at the forefront when a new technology strikes our deeply felt convictions and existential interests. They can also function as a check on consequentialism, because deontological principles appear to have a ''right of way" before consequences.

Deontological theories hold that some acts are always wrong, even if the act leads to an admirable outcome. Essentially, deontological theories differ from utilitarian theories in several key ways. Utilitarianism aims at a goal of greatest happiness (or the best consequence) and justifies any act that achieves that goal whereas deontological theories hold that some acts are always wrong even if the act leads to an admirable outcome.

Actions in deontology are always judged independently of their outcome. An act can be morally bad but may unintentionally lead to a favourable outcome (Shakil, 2015).

Kant is responsible for the most prominent and well-known form of deontological ethics. He was a proponent of deontological moral theory and formulated the most influential form of a deontological moral theory in 1788. Essentially, in deontology ethics, something is "good in itself" when it is described as intrinsically good, and "good without qualification", when the addition of that thing never makes a situation ethically worse.
Kant emphasizes the moral worth of an action that is essentially determined by the human will which is the only thing in the world that can be considered good without qualification. Good will is exercised by acting according to moral duty/law and that moral law consists of a set of maxims. These Maxims are categorical in nature and we are bound by duty to act in accordance with categorical imperatives.

Kant formulated the categorical imperative.

- Act only according to that maxim by which you can also will that it would become a universal law.
- Act in such a way that you always treat humanity, whether in your own person or in the person of any other, never simply as a means, but always at the same time as an end.
- Every rational being must so act as if he were through his maxim always a legislating member in a universal kingdom of ends (Shakil, 2015).

1.3.3.1 Deontology ethics and Industry examples

In sum, deontological morality leaves space for individuality and specials concerns. Deontology ethics in relation to designers or programmers can guide how programmers can be steered by moral values and intentions. It's for example the designers its duty to make applications that are not that invasive or persuasive and build trust for the end-user. Also a software application should not be too aggressive e.g. blocking a person's its operating system and it can be argued that it is the duty of the designer to balance these values. At a higher level within the organization duty may also be at play e.g. often designers do not make decisions on their own but are most of the time guided by a framework that the company provides. For example design guides (that allows to make intrusive apps), design ethics, or the tooling that forces programmers to build their components with (some components maybe more intrusive than others e.g. an application with a camera function. Later in the thesis specific examples will be given about the choice of components and design architecture that forces companies into different ways of developing e.g. changing design rules.

1.4 What is Virtue Ethics?

An important issue concerning design ethics is the nature of the moral character of the designer. In Virtue ethic the perspective focuses on the actor namely who performs an act moral or not. Virtue ethics has found a basis in the golden mean. This theory states that "in general good comes before right" (teleological), that a person should have courage, not be a coward but also not be easily fooled. Engineers can strive to be excellent engineers but this does not mean they can develop anything and they're able to do anything that they put their mind to.

Table I
The virtues as discussed in Aristotle's *Nicomachean Ethics* (*NE*)

Virtue	Sphere of exercise	Discussion in *NE*
Courage	Fear and confidence	II.6-9
Temperance	Bodily pleasure and pain	III.10-12
Generosity	Giving and retaining money	IV.1
Magnificence	Giving and retaining money on a large scale	IV.2
Greatness of soul	Honour on a large scale	IV.3
(Nameless)	Honour on a small scale	IV.4
Even temper	Anger	IV.5
Friendliness	Social relations	IV.6
Truthfulness	Honesty about oneself	IV.7
Wit	Conversation	IV.8
Justice	Distribution	V
Friendship	Personal relations	VIII-IX

(Little, 2014)

In Aristotle's Nicomachean ethics several examples of important virtues are addressed. For example, a person should have a balance between fear and confidence that makes up his courage. Or have an even temper.

1.4.1 Examples of Virtue ethics and Industry examples

Within industry there are several roles that a programmer or designer may have. In the below overview, examples are given how virtues may be at play within the industrial context. Don't let fear stop you from doing things, especially things you know you must do to be successful. You don't need to go that far, but if there are things you're afraid to do, or that you know are your weak points, do what you must to get better at them and become more comfortable doing them. The below example shows how i.e. *courage* and *accountability* may be important. Within business people should have the courage to make decisions under uncertain conditions and confidence to act. Or for example accountability is important, it may be important to take ownership and the commitment to make decisions and commitment to the execution.

Virtue	Good Organizational Outcomes (Virtue is present)	Bad Organizational Outcomes (Virtue is absent)
Judgment	Quality decisions, calculated risk-taking, commitment, support, trust.	Lack a balanced assessment of the issues leading to misinformed decisions, confusion, resistance to change.
Humanity	Social responsibility, good employee relations, understanding, support.	Misses critical social implications of decisions and actions. Alienation of followers, lack of respect for leader.
Justice	Use diversity, good employee relations, fairness, organizational citizenship behaviors.	Inequities not identified and managed thereby eroding trust. Favoritism, nepotism.
Courage	Decisions made under conditions of uncertainty, confidence to act, opposition to potentially bad decisions, innovation.	Going along with poor decisions. Satisficing rather than maximizing. Moral muteness.
Collaboration	Teamwork, use diversity, cross-enterprise value-added, innovation, learning, affiliation, confidence.	Individualism alienates potential allies. Poor understanding of decisions, friction, conflict.
Accountability	Ownership and commitment to decisions and their execution	Failure to deliver results and typically creates excuses for why not. Shirking of responsibility.
Humility	Continuous learning, quality decisions. Respect, trust.	Ego driven behavior, selective listening, difficulty admitting error or failure. Arrogance, overconfidence, complacency, hubris.
Integrity	Builds trust, reduces uncertainty, develops partnerships and alliances, promotes collaboration and cooperation.	Creates mistrust, requires firm guarantees, slows down action, undermines partnerships and alliances, reduces cooperation and collaboration.
Temperance	Quality decisions, reduced risk.	Short-termism, inability to see the possible constraints, instant gratification.
Transcendence	Focus on superordinate goals, big-picture thinking, strive for excellence.	Narrow aims, little inspiration, tunnel vision.

("Developing Leadership Character | Ivey Business Journal," 2012)

These virtues can be found different per company. Companies have different integrity for example how they handle and store private information, building trust with partners (aggressive or passive) and taking social responsibility.

1.5 Going more towards a framework of applied ethics and taking action

The earlier explained ethical theories provide a normative judgment and describe how to go about it. However, other theories describe that in most of the time a combination of these normative theories is used within society (or even without noticing).

While the classical ethical theories describe normative judgment they do not cover *practice.* Applied ethics is a branch of ethics devoted to the treatment of moral problems, practices, and policies in personal life, professions, technology, and government. In other words, applied ethics emphasizes the philosophical examination from a moral standpoint in the private or public life that requires moral judgment. Bioethics is for example the study of ethical issues related to controversial and emerging situations brought by advances in biology and medicine. Activities involve developing medical policy, practice and research regarding moral discernment.

Similar to bioethics, applied ethical issues have been subdivided into convenient groups such as medical ethics, business ethics, environmental ethics, and sexual ethics (Cochrane, 2015). Much of applied ethics is often concerned with just three theories (utilitarianism, Deontological ethics, and virtue ethics) as a basis of moral judgment. Ethical pragmatists acknowledge that it can be appropriate to practice a variety of other normative approaches (e.g. consequentialism, deontological ethics, and virtue ethics), yet acknowledge the need for mechanisms which allow society to advance beyond such approaches, a freedom for discourse which does not take any such theory as assumed.

1.5.1 Pragmatic Ethics

For the pragmatists essentially, practice is primary; theories serve practice and they are instrumental. ("Pragmatic Ethics," n.d.) In addition, pragmatic ethics treats morality like science: advancing socially over the course of many lifetimes, such that any moral criterion is subject to revision. Pragmatic refers to the belief that we should use what works and alter or discard what doesn't.

Pragmatic ethics, originated in the works of William James and John Dewey and in those of some later writers making the following claims: in general norms (knowledge of how properly to do things) are discovered by people as they solve practical problems. This is true of moral as well as non-moral norms. Our knowledge of ethical norms and goods is derived from our experience in resolving problems encountered in living with one another and pursuing common practices and goals; it is our knowledge of how to live with one another. The knowledge is *a posteriori* and based on experience. Moreover, pragmatic ethics emphasizes that the world and our lives change continually and that ethical norms undergo change as well, although the change is generally gradual and incremental. Changes are effected as people resolve problems involving conflicts among practices and their constituent norms and share with others what they have learned(Wallace, 2013).

1.5.2 An introduction to NEST-Ethics

Another applied approach that uses classical ethical theory is NEST-Ethics that focuses on the dynamics of ethical issues and debates within society regarding technological development. Ethics of **N**ew and **E**merging **S**cience and **T**echnology (NEST-ethics) as proposed by Swierstra & Rip(2007) discusses the characteristic **tropes and patterns in moral argumentation**.

NEST-ethics starts with the opening up an existing order by a scientific or technological novelty that undermines the self-evidence of existing moral routines, in combination with the additional challenge of our ignorance about the nature and effects of this novelty. NEST identifies and characterizes patterns of moral argumentation as they occur (with examples from nanotechnology) parts of the argumentation are at the meta-level, namely about our background understanding of issues and also how to approach them instead of good action and good life. This is because of the new and emerging character of NEST, where it is often too early to conclude about concrete technical issues, however the prospect of doing so induces discussion how to go about it and this will raise meta-ethical questions.

Considerations at the meta-level in NEST debates deal with how actors are involved in the new technology relate to the control of its development, the relation between technology and society, the relation between technology and morality. NEST-Ethics also recognizes that one can influence the development of new technology and therefore it is important to discuss the desirability and the feasibility. NEST-Ethics argues that human agency, so dear to classical ethics, has to be replaced by distributed and collective agency, and a time dimension has to be introduced. NEST repositioned technology determinism as a contingent result of actor's behaviours and interactions leading to unintended outcomes at the collective level. Understanding of such processes then enables agency, in the sense of making a bit of difference rather than forcing one' s way.

1.5.2.1 Consequential arguments

Consequentialist contestation follows a distinctive pattern, which is fuelled by two general perspectives on technology which are linked to the meta-ethical discussion of agency. There is an optimistic view that technology progress is basically beneficial and a pessimistic view that technology is risky and dangerous. The optimistic view emphasizes that in the long run new technology and its progress will benefit us. The optimistic view gets fuelled by the argumentations that you should not want to stop the technological advance but you cannot either. Resistance is often seen as bad and futile. The pattern of moral argumentation starts with promises which have the form: if we invest in this new and emerging science and technology, this will increase our knowledge as well as our scope in manipulating the natural world, which will eventually result in increasing

general happiness when application of such knowledge and manipulation leads to positive effects x, y and z. (Swierstra & Rip, 2007,p12)

- The first axis concerns the basis of the promises made, that is, their plausibility.
- The second axis along which promises can be contested, is not the plausibility of the benefits, but the ratio of benefits and costs.
- The third axis of consequentialist contestation consists of questioning whether the benefits promised are really benefits.

1.5.2.2 Deontological arguments

Technology may promise to maximize overall happiness but at the expense of moral convictions, duties and rights. Deontological arguments are not only brought up to counter optimistic promises but also in support of the technology: the duty to further human progress, the duty to diminish suffering, the duty to acquire knowledge but also the right to choose whether to use or not a technology.

1.5.2.3 Good life arguments

Promises concerning some technologies could go beyond utility, rights and duties. Good life arguments sometimes with appeal to mythological motives or cultural shaped identities .Promoting new technology typically draw upon a Promethean identity, mixed with some frontiers rhetoric “Boldly go where no man went before”.

1.5.2.4 Justice arguments

The argumentation of distributive justice are not detail featured in the NEST debates because of the speculative nature of the impacts on the distribution of goods. Arguments feature that the technology divides the gap between those who have access to technology and not, which may take many forms the gap between the rich and poor countries, the gap between the rich and poor countries of our population. The new technology will create more goods/value, and therefore everyone can have a larger piece in absolute terms of the expanded cake.

Technological steering: there are people who do not believe that significant impact of human agency is possible and they appeal to the free market mechanisms and the dynamics of international competition and the internal logic of technologies. On the other hand there are people who believe in social shaping of technologies and therefore they are supporting a democratic control which increases responsibility and accountability and require transparency (Niculescu-Dinca, 2009,p12).

Anticipatory Technology Ethics

The previous applied ethical theories provided an approach to look into the dynamics and patterns of moral debate in society. Another ethical approach is to look more at the ***components of technology*** itself and the processes that a new technology may bring about. Brey suggests an ethical approach which is formulated as Anticipatory Technology Ethics (ATE). Brey proposes three levels of ethical analysis namely Technology level, Architect level and the Application level (P. Brey, 2012). At each of these levels various objects of ethical analysis are defined: *things, properties or processes* that raise ethical issues.

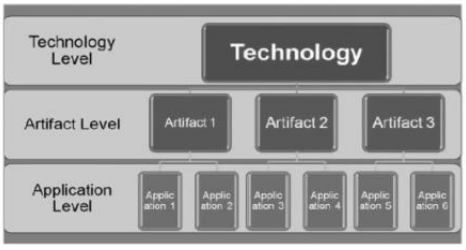


Fig. 1 Three levels of ethical analysis

These levels are explained below:

(1) **The Technology level** is the level at which a technology is defined. A technology is a collection of techniques that are related to due to the common domain, purpose, formal or functional features. In Nano technology it is for example possible to distinguish optical-nano technology and DNA nano technology. In the level of ethical analysis the focus is especially on the general features of the technology(Brey, 2012, p7).

(2) **The artefact** defines the physical configuration that when operated produces a desired result. A procedure is a sequence of actions when performed in a manner using the right tools, produces a desired result. From technology technological artefacts and procedures are derived which technology produces. At the artefact level ethical analysis focuses on types of artefacts and processes that resulted from a particular technology (Brey, 2012, p8).

(3) At **the application level** ethical analysis focuses on particular ways of using an artefact or procedure which is in many cases the configuration of its usage also speaking term of the context of use. Ethical issues that may play at the application level are moral issues that could be related to the intended use of the artefact (Brey, 2012, p8).

Forecasting

Brey proposed a forecasting method specifically within these three levels of assessment. First at the (1) Technology level, knowledge will be acquired from engineers because they best positioned to understand and describe the features of the technology, the techniques and the subclasses of the technology. They are positioned to inform ethicists. This level may also require consultation of other experts from other field of studies. Within the (2) Artefact level analysis should utilize existing studies in forecasting and TA about the technology, to the extent that these are available.

Within the (3) Application level, TA experts should perform analysis with a view of artefacts and applications that are most likely to emerge in the future. E.g. by consulting engineers, technology forecasters and TA experts but also historians, sociologists and marketing experts. Because of its imaginative activity in the methods identification stage it may also be useful to consider policy documents, company studies, academic texts or even science fiction stories for ideas about possible future artefacts and applications, as long as these ideas are then subjected to scrutiny regarding their feasibility and plausibility. This particular interest may imply that ethicists will sometimes have to develop their own forecasts and scenarios that focus on such matters.

Examples of Anticipatory Ethics and industry

Anticipatory ethics scrutinizes what objects, properties and processes in technology may bring about and how to asses those on the artefact or application level e.g. using forecasting scenarios. These levels provide a general scope to assess technology using multiple evaluation methods. Within industry, Anticipatory Ethics provides a practical and pragmatic way to find ethical issues related to technology that occurs on the macro and the micro level.

Often technology is build according a framework, by abstracting the core technology in the layers of the technology- level, artefact level and the application level specific methods of analysis may be reused, targeted and positioned. For instance workshops may be suitable on the application level that provides the analysis on that specific application level. A big question is how to capture the different ethical issues within these levels of analysis, throughout the thesis methods were discussed.

Conclusion

The discussed theories in this chapter provide a basis for how to ask ethical questions (e.g. meta-ethics), how to make decisions (e.g. pragmatic ethics or normative ethics), where to look on ethical issues (e.g. at a debate using NEST-Ethics or at technical components using Anticipatory Technology Ethics) and how to justify these decisions (e.g. normative ethics). These moral frameworks can be very helpful in technology design. As stated before technology aims at making people's lives better, e.g. increase efficiency in terms of saving time, or it aims to encourage people to stay connected e.g. social networking sites. With aims like this technology itself has an ethical role, given that ethics is dedicated to asking questions about what makes a good life and how we evaluate it. The contemporary ethical theories may not however be always that practical to implement within the practice but functions well as an introduction to ethics. Therefore, more practical theories have been added in the moral intelligence framework that is developed during Chapter 4. This extra chapter can function as an introduction for engineers, consultants and programmers to understand different ethical perspectives but we need a more pragmatic other approach to connect engineering (practice) and ethics.

Appendix L Quick Start Guide
Moral Intelligence for IT industry

Problem: CAPE Groep BV is an IT consultant company that develops software for the logistics industry by using Mendix as their main development suite. Their logistics software is sold as an ERP & CRM package for Logistic companies (e.g. Post NL), which allows them to keep track of their postage and parcels. Logistic companies have to deal with many (micro) transactions that must be stored and processed using the ERP & CRM software.

Goal: CAPE Groep BV seeks a transparent, secure method that incorporates foreseeing moral issues that may or not arise during the development of their software. Furthermore, CAPE Groep BV seeks a way of innovating and creating security measures and increasing the overall quality of their products for their customers and stakeholders

Figure 6.1 CAPE Groep BV - Workshop what are conceptualisations of security?
Source: CAPE Groep BV Facebook page

Practical problems

- Often old versions or new software updates can harm how the application handles the data. E.g. by neglected security patches or new feature releases that haven't been fully tested.
- CAPE Groep BV's aim is to get a better understanding of the moral issues they may face in a broad sense. Moreover, they want to gain better insights into the values of their customers and stakeholders.

Approach: According the Moral Intelligence Framework for IT industry, we can use the proposed Capture, Design & Deploy method to gather Moral Intelligence to design products. First, start with MI capture, gather ethical intelligence on the different levels of the technology. Secondly, integrate Moral Intelligence within the development method in this case Model Driven development as explained earlier.

Step 1: Start with Capturing Moral Intelligence

1. Scrutinize technology creation technology at large
2. The types of artefacts and processes that result from the technology (technology object level)
3. The concrete use of the application (on the micro level)
4. The (to be) designed processes and used production methodologies (linkage to design)

Tool: Use the provided Capture Blueprint, presented below, discuss with tools such as Asana.com. and work together within the model driven development methodology.

Class →	*Object Analysis →*	*Procedures Analysis →*	**Design Process Analysis**
Technology creation level(ETC)	Empirical, Technical, Conceptual (ETC)	Empirical, Technical, Conceptual (ETC)	(Practical) Design analysis
How?	**How?**	**How?**	**How?**
Engineers consult ethicist and vice versa for forecasting. - ethical analysis focuses on features of the technology at large, particular subclasses of it, or techniques within it.	*-Extended ethical checklist -NEST ethics - scenario sketching, define larger scopes. -Expert survey -Utilise existing studies in forecasting and TA about the technology, to the extent that these are available*	*-Future assessment e.g. with workshops, evaluation, consulting TA experts historians, sociologists, and marketing experts, policy documents, company studies, academic texts. -Analyse application and system variables*	*-Integrate ETC in production methodologies and tooling -Link ETC to process and design methods -Analyse by using the 7s model (style, system, structure skill, etc.)*
Ethical analysis focuses on	**Ethical analysis focuses on**	**Ethical analysis focuses on**	**Ethical analysis focuses on**
Features of the technology at large (the defining feature of the technology)	Types of artefacts and processes that result from the technology	Concrete use of the application	The (to be) designed processes and used production methodologies

Cultivating Moral Intelligence and data

Each level can be analysed using different data collection methods, which are, in turn, evaluated on a technological, conceptual, and empirical level. In the case of the technology creation level, CAPE Groep BV B.V. can intensively study the development of their suppliers, architecture providers, but also latest trends in the IT industry. What are standards to make applications safer? What features are used by the industry? These questions are asked in the technology creation level.

Towards the object creation level, CAPE Groep BV B.V. may ask what could be consequences, intentions, or the virtues of stakeholders when they choose certain actions and how are they embedded in the components used during development (e.g. which software libraries, which components?) NEST ethics may provide useful insights, for example, what are current debates in the community (Mendix community) about relevant issues within the same domain? Furthermore, a scenario sketching method may be used to layout practical yet pragmatic steps. The scenarios can be specified by knowing procedures quite well. Ethicists or consultants may organise workshops to gain better insights into detailed processes and embrace 1 capture, 2-design, 3-deploy iteration.

Tool: Use forums, scenario sketching, envisioning cards, brainstorm sessions, or even additional tools e.g. the Foresight Diamond (provided in Appendix D).

Step 2: Incorporate Moral Intelligence within Design Teams

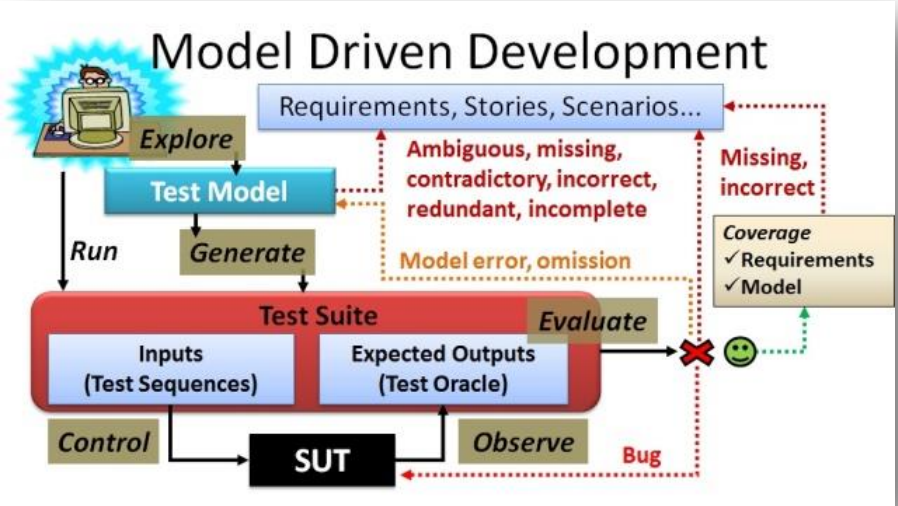
Use the developed team matrix to work together in combination with the Capture framework above.

(Shared) responsibility		
Multiple responsibility	Capable responsibility	Special responsibility
1 **Value Discovery Ethicist**	**Engineer /programmer**	**Role Consultant**
A. Making explicit the intended values as conceived by engineers, designers or consultant.	Provide input & arguments for the intended values together with ethicist.	Provide input & arguments for the intended values.
B. Scrutinising these values by illustrating value conflicts and trade-offs.	Provide input e.g. using the Moral Intelligence framework together with ethicist	Provide input e.g. using the Moral Intelligence framework together with ethicist
C. Comparing intended values with use of Moral Intelligence framework, e.g. ethical literature, workshops or value for design methods, etc.	Make a priority list of the intended values and ethical participation in workshops, brainstorms sessions	Make a priority list of the intended values and ethical participation in workshops, brainstorms sessions.

2 Translating Values into Design Requirements Ethicist	Engineer/Programmer	Consultant
A. Value conceptualisation (Customers, Public)	Gathering, providing feedback for the ethicists and consultant e.g. with the use of a Moral Intelligence	Value conceptualisation with customers and stakeholders (liaison role).
B. Describing the disconnect between intended and realised values by speculating unintended uses and contexts of the artefact.	Gathering (realised) value concepts given by ethicist or IT consultants and advising which engineering concepts may be suitable.	Mediate between the engineer and ethicists to find the best possible solution between intended and realised values e.g. within the role of a product owner (as an advisor to the ethicist)
C. Exploring the translation/specification of values into design requirements.	Exploring and realising the best possible building blocks for the implementation of values.	Exploring the translation/specification of values into implementation.

Incorporate Moral Intelligence within Design

Once important data is captured, the data needs to be incorporated into design. During the VSD/Moral Intelligence design phase, we will look at how CAPE Groep BV is currently developing its software. The development methodology[16] is based on MDD, which allows for consultants to scrutinise processes without needing to read and understand code. Consultants study requirements, stories, and scenarios in-depth obtained from the previous capturing phase and model the software accordingly. They may invite ethicists who may function as designers and vice versa as well.



The ethicist can integrate components together with the consultant. MDD development allows for working together and working on stories, models and directly integrating ideas.

[16] Use the development overview from Chapter 6

Step 3: Deployment of Moral Intelligence throughout the organisation

Increasing organisational flexibility and transparency

In the final step, 7s dimensions can be assessed to optimally align the organisational contingencies to implement new features. CAPE Groep BV may fall back on a business analysis to see how the company can improve its 7s dimensions in accordance the incorporation of Moral Intelligence and the optimisation of development.

Example: It is first important to understand what are the shared values of CAPE Groep BV? How can thinking about moral issues be incorporated in the products? What systems can CAPE Groep BV use to improve the quality and stimulate development and VSD practice? The deployment framework can provide the right systems and basics to start with the incorporation of creating solutions that may improve privacy issues.

Use the below interventions to incorporate Moral Intelligence within an IT organisation.

Dimension	Relation to VSD/Moral Intelligence	Suggested Intervention
Shared value	These are the core values of the company that are evidenced in the corporate culture and the general work ethic.	A clear vision and mission that helps to steer VSD/Moral Intelligence integration, e.g. the mission and vision. *See also Chapter 7*
Systems	Develop systems that can provide the basis for capturing values, e.g. use envisioning cards, applications to gather information, like surveys or a repository and promote the overall learning, monitoring and reporting aspect of Moral Intelligence.	Build systems for the capturing framework (learning, monitoring, reporting), e.g. survey systems, feedback systems, interview systems, use of envisioning cards, 360 degree feedback or other VSD/Moral Intelligence components. *See also Chapter 4*
Structure	Provide the necessary structure to support active participation in VSD/Moral Intelligence. Structure can mean how departments within the organisation, roles, coordination, communication lines cooperate.	Create flexible teams, a flat organisation structure; allow VSD/Moral Intelligence reporting within the organisation and quickly incorporate changes e.g. with the usage of the Moral Intelligence disciplinary team matrix. *See also Chapter 6*
Style	Provide sessions to raise ethical questions within the organisation.	Incorporation within the strengths of Scrum and continuous integration may provide quality improvements within Sprints. If another development method is chosen plan extra meetings e.g. 'Social Fridays'. *See also Chapters 5/6*
Staff	Have staff that can work together, take leadership, and engage with ethical issues to support ethical reasoning staff development.	A disciplinary team matrix that stimulates collaboration and ethical investigation. *See also Chapter 6*
Skills	Ensure that the development team gets involved with VSD/Moral and develops skills.	VSD/Moral Intelligence training, workshops, supporting literature, online forum, repository *See also Chapters 5/6*
Strategy	Incorporation of VSD/Moral within methodology to develop better quality products, for example.	A quick deployment framework and incorporation within Scrum, MDD, and Continuous integration *See also Chapter 7*

Figure 6.15 Moral Intelligence deployment framework for IT organisations

Literature Overview

About | Envisioning Cards. (n.d.). Retrieved June 1, 2017, from http://www.envisioningcards.com/?page_id=2

Activity Based Carboning. (n.d.). Retrieved May 30, 2017, from https://www.CAPE Groep BV.nl/activity-based-carboning/

Agile Alliance :: The Agile Manifesto. (2001). Retrieved December 6, 2015, from http://www.agilealliance.org/the-alliance/the-agile-manifesto/

Alan Borning - CROW 2016. (2016, March 15). Retrieved July 26, 2017, from http://conf.researchr.org/profile/CROW-2016/alanborning

Albrechtslund, A. (2007). Ethics and technology design. *Ethics and Information Technology*, *9*(1), 63–72. https://doi.org/10.1007/s10676-006-9129-8

Batya Friedman | Information School | University of Washington. (2016). Retrieved March 26, 2016, from https://ischool.uw.edu/people/faculty/batya

Boenink, M., Swierstra, T., & Stemerding, D. (2010). Anticipating the Interaction between Technology and Morality: A Scenario Study of Experimenting with Humans in Bionanotechnology. *Studies in Ethics, Law, and Technology*, *4*(2). https://doi.org/10.2202/1941-6008.1098

Borning, A., & Muller, M. (2012). Next steps for value sensitive design (p. 1125). ACM Press. https://doi.org/10.1145/2207676.2208560

Brey, P. (2009). *Values in Technology and Disclosive Computer Ethics* (Vol. The Cambridge Handbook of information and Computer Ethics). Cambridge University Press.

Brey, P. (2012). Anticipatory Ethics for Emerging Technologies. Springer Science + Business Media B.V. 2012.

Brey, P. A. E. (2012). Anticipatory Ethics for Emerging Technologies. *NanoEthics*, *6*(1), 1–13. https://doi.org/10.1007/s11569-012-0141-7

Campbell-Kelly, M. (2003). Software Industry – Dictionary definition of Software Industry | Encyclopedia.com: FREE online dictionary. Retrieved April 17, 2016, from http://www.encyclopedia.com/doc/1G2-3401803934.html

CAPE Groep | Grensverleggend vooruit. (n.d.). Retrieved January 7, 2016, from https://www.CAPE Groep BV.nl/

Cochrane, A. (2015). Ethics | Internet Encyclopedia of Philosophy. Retrieved July 13, 2015, from http://www.iep.utm.edu/ethics/#H3

Cockton, G., Lárusdóttir, M., Gregory, P., & Cajander, Å. (Eds.). (2016). *Integrating User-Centred Design in Agile Development*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-32165-3

Davis, J., & Nathan, L. P. (2013). Value Sensitive Design: Applications, Adaptations, and Critiques. In J. van den Hoven, P. E. Vermaas, & I. van de Poel (Eds.), *Handbook of Ethics, Values, and Technological Design* (pp. 1–26). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-6994-6_3-1

Developing Leadership Character | Ivey Business Journal. (2012, February). Retrieved June 22, 2015, from http://iveybusinessjournal.com/publication/developing-leadership-character/

Edward Snowden: Leaks that exposed US spy programme. (2014, January 17). Retrieved January 22, 2015, from http://www.bbc.com/news/world-us-canada-23123964

Ellis, G. (2016). Agile Project Management. In *Project Management in Product Development* (pp. 223–260). Elsevier. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/B9780128023228000085

Ford, H. (n.d.). 21 Quotes From Henry Ford On Business, Leadership And Life. Retrieved October 9, 2016, from http://www.forbes.com/sites/erikaandersen/2013/05/31/21-quotes-from-henry-ford-on-business-leadership-and-life/#ec4aee03700a

Friedman, B. (1999, August 23). Value-Sensitive Design: A Research Agenda for Information Technology. University of Washington.

Friedman, B., H. Kahn Jr, P., & Borning, A. (2001). Value Sensitive Design: Theory and Methods. UW CSE Technical Report.

Friedman, B., Kahn, P. H., & Borning, A. (2008). Value Sensitive Design and Information Systems. In K. E. Himma & H. T. Tavani (Eds.), *The Handbook of Information and Computer Ethics* (pp. 69–101). Hoboken, NJ, USA: John Wiley & Sons, Inc. https://doi.org/10.1002/9780470281819.ch4

Grunwald, A. (2001). The application of ethics to engineering and the engineer's moral responsibility: Perspectives for a research agenda. *Science and Engineering Ethics*, *7*(3), 415–428. https://doi.org/10.1007/s11948-001-0063-1

Haines, W. (2015). Consequentialism | Internet Encyclopedia of Philosophy. Retrieved July 12, 2015, from http://www.iep.utm.edu/conseque/

Kelly, H. (2013). Facebook Home, privacy and you - CNN.com. Retrieved January 22, 2015, from http://edition.cnn.com/2013/04/05/tech/social-media/facebook-home-privacy/

Kharpal, A. (2016). Apple vs FBI: All you need to know. Retrieved April 17, 2016, from http://www.cnbc.com/2016/03/29/apple-vs-fbi-all-you-need-to-know.html

Lanzo, J. F. (2003, July 25). Ethical Responsibility in Engineering: A Fundamentation and Proposition of a Pedagogic Methodology. International Conference on Engineering Education.

Legg, S., & Hutter, M. (2007). *A collection of Definitions of Intelligence*. Retrieved from https://arxiv.org/pdf/0706.3639.pdf

Little, D. (2014, December 7). character « Understanding Society. Retrieved June 15, 2015, from http://undsoc.org/tag/character/

Maedche, A. (2017). Interview with Prof. Jeroen van den Hoven on "Why do Ethics and Values Matter in Business and Information Systems Engineering?" *Business & Information Systems Engineering*, *59*(4), 297–300. https://doi.org/10.1007/s12599-017-0476-2

Manders-Huits, N. (2011). What Values in Design? The Challenge of Incorporating Moral Values into Design. *Science and Engineering Ethics*, *17*(2), 271–287. https://doi.org/10.1007/s11948-010-9198-2

McCarthy, J. (2006). *Dynamics of software development*. Microsoft Press.

McKinsey 7Ss Model | Developing Together. (2015). Retrieved January 22, 2015, from http://www.nwacademy.nhs.uk/developingtogether/mckinsey-7ss-model

Metaethics -Stanford Encyclopedia of Philosophy. (2007, January). Retrieved June 22, 2015, from http://plato.stanford.edu/entries/metaethics/

Mikoluk, K. (2013, September 9). Agile Vs. Waterfall: Evaluating The Pros And Cons. Retrieved January 3, 2016, from https://blog.udemy.com/agile-vs-waterfall/

Miller, J. K., Friedman, B., & Jancke, G. (2007). Value tensions in design: the value sensitive design, development, and appropriation of a corporation's groupware system (p. 281). ACM Press. https://doi.org/10.1145/1316624.1316668

Mkb-innovatiestimulering Regio en Topsectoren (MIT) | RVO.nl. (n.d.). Retrieved June 6, 2017, from http://www.rvo.nl/subsidies-regelingen/mkb-innovatiestimulering-regio-en-topsectoren-mit

Niculescu-Dinca, V. (2009). NEST-ethics in convergence : testing NEST-ethics in the debate on converging technologies for improving human performance. Retrieved from http://essay.utwente.nl/59321/

Nissenbaum, H. F. (2010). *Privacy in context: technology, policy, and the integrity of social life*. Stanford, Calif: Stanford Law Books.

Pablo Picasso Quotes. (n.d.). Retrieved October 9, 2016, from http://www.pablopicasso.org/quotes.jsp

Palm, E., & Hansson, S. O. (2006). The case for ethical technology assessment (eTA). *Technological Forecasting and Social Change*, *73*(5), 543–558. https://doi.org/10.1016/j.techfore.2005.06.002

Pesch, U. (2015). Engineers and Active Responsibility. *Science and Engineering Ethics*, *21*(4), 925–939. https://doi.org/10.1007/s11948-014-9571-7

Peter Kahn. (n.d.). Retrieved March 26, 2016, from http://faculty.washington.edu/pkahn/

Porter, M. E., & Kramer, M. R. (2006, December). Strategy and Society: The Link between Competitive Advantage and Corporate Social Responsibility. Harvard Business Review 84.

Pragmatic Ethics. (n.d.). Retrieved June 18, 2015, from http://www.philosophy.uncc.edu/mleldrid/SzCMT/prag.html

Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, *50*(4), 280–295. https://doi.org/10.1016/j.infsof.2007.02.002

Reddy, A. (2016). *The Scrumban [r]evolution: getting the most out of Agile, Scrum, and lean Kanban*. New York: Addison-Wesley.

Rouse, M. (n.d.). What is Information Technology (IT)? - Definition from WhatIs.com. Retrieved January 11, 2016, from http://searchdatacenter.techtarget.com/definition/IT

Schulkes, Y. (2015, December 14). De glazen bol van TRIMM: internettrends voor het komend jaar | TriMM Digital craftsmanship. Retrieved January 4, 2016, from http://www.trimm.nl/blogs/de-glazen-bol-van-trimm-internettrends-voor-het-komend-jaar

Scrum Guide | Scrum Guides. (n.d.). Retrieved May 29, 2017, from http://www.scrumguides.org/scrum-guide.html

Segran, E. (2014, August 28). Why Top Tech CEOs Want Employees With Liberal Arts Degrees. Retrieved July 19, 2017, from https://www.fastcompany.com/3034947/the-future-of-work/why-top-tech-ceos-want-employees-with-liberal-arts-degrees

Shakil, A. (2015). Deontology - Kantian Duty-Based Ethics - Seven Pillars Institute Seven Pillars Institute. Retrieved July 11, 2015, from http://sevenpillarsinstitute.org/morality-101/kantian-duty-based-deontological-ethics

Spiekermann, S. (2015). *Ethical IT Innovation: A Value-Based System Design Approach*. Auerbach Publications. https://doi.org/10.1201/b19060

Steenwijk, M. (2011, April 11). A Reference Architecture for Fuel based Carbon Management Information Systems in the Logistics Industry. Retrieved from http://essay.utwente.nl/61151/1/MSc_M_Steenwijk.pdf

—Steve Jobs | Values.com. (n.d.). Retrieved January 7, 2016, from http://www.values.com/inspirational-quotes/6652-if-you-are-working-on-somet-

Subbaiah, V. (2016, February 18). Why Agile Is The Best Alternate Methodology To Waterfall? – Blog | Asahi Technologies. Retrieved July 27, 2017, from https://www.asahitechnologies.com/blog/why-agile-is-the-best-alternate-methodology-to-waterfall/

Swierstra, T., & Rip, A. (2007). Nano-ethics as NEST-ethics: Patterns of Moral Argumentation About New and Emerging Science and Technology. *NanoEthics*, *1*(1), 3–20. https://doi.org/10.1007/s11569-007-0005-8

TU Delft: Prof.dr. M.J. (Jeroen) van den Hoven. (n.d.). Retrieved January 26, 2017, from http://www.tbm.tudelft.nl/nl/over-faculteit/hoogleraren/profiel-van-een-prof/jeroen-van-den-hoven/

Value Sensitive Design | Public Sphere Project. (n.d.). Retrieved October 3, 2015, from http://www.publicsphereproject.org/node/286

van de Braak, P. (2010, July 27). Finding a customer specific model driven development methodology. Retrieved from http://essay.utwente.nl/59918/1/MA_thesis_P_van_de_Braak.pdf

van den Hoven, J., Vermaas, P. E., & van de Poel, I. (Eds.). (2015). *Handbook of Ethics, Values, and Technological Design*. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-6970-0

Van Wynsberghe, A. (2012). *Designing Robots with Care*. WPI Wohrmann Print Service, Zutphen, The Netherlands.

van Wynsberghe, A., & Robbins, S. (2014). Ethicist as Designer: A Pragmatic Approach to Ethics in the Lab. *Science and Engineering Ethics*, *20*(4), 947–961. https://doi.org/10.1007/s11948-013-9498-4

Verbeek, P.-P. (2008). Morality in Design: Design Ethics and the Morality of Technological Artifacts. In P. Kroes, P. E. Vermaas, A. Light, & S. A. Moore, *Philosophy and Design* (pp. 91–103). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-1-4020-6591-0_7

VSD: projects. (2011). Retrieved May 26, 2015, from http://www.vsdesign.org/projects.shtml

Wallace, J. D. (2013). Pragmatic Ethics. In H. Lafollette (Ed.), *International Encyclopedia of Ethics*. Oxford, UK: Blackwell Publishing Ltd. Retrieved from http://doi.wiley.com/10.1002/9781444367072.wbiee256

Zimmer, M., & Manders-Huits, N. (2009). Values and pragmatic action: The challenges of introducting ethical intelligence in technical Design Communities. International Review of Information Ethics.