Know Techn Pol (2010) 23:491–505
DOI 10.1007/s12130-010-9130-8

SPECIAL ISSUE

Designing for Trust: A Case of Value-Sensitive Design

Pieter E. Vermaas · Yao-Hua Tan · Jeroen van den Hoven · Brigitte Burgemeestre · Joris Hulstijn

Received: 8 April 2010 / Accepted: 24 July 2010 / Published online: 22 September 2010

Abstract In this paper, we consider the meaning, roles, and uses of trust in the economic and public domain, focusing on the task of designing systems for trust in information technology. We analyze this task by means of a survey of what trust means in the economic and public domain, using the model proposed by Lewicki and Bunker, and using the emerging paradigm of value-sensitive design. We explore the difficulties developers face when designing information technology for trust and show how our analysis in conjunction with existing engineering design methods provides means to address these difficulties. Our main case concerns a concrete problem in the economic domain, namely the transfer of control from customs agencies to companies. Control of individual items is increasingly untenable and is replaced by control on the level of companies aimed at determining whether companies can be trusted to be in control of their business and to be in compliance with applicable regulations. This transfer sets the task for companies to establish this trust by means of information technology systems. We argue that this trust can be achieved by taking into account philosophical analyses of trust and by including both parties in the trust relationship as clients for whom the information technology systems are to be designed.

Keywords Design methods · Design for trust · Information technology · Trust · Value-sensitive design

P. E. Vermaas (✉) · J. van den Hoven
Philosophy Department, Delft University of Technology, Delft, The Netherlands
e-mail: p.e.vermaas@tudelft.nl

Y.-H. Tan
Information and Communication Technology Department, Delft University of Technology, Delft, The Netherlands

B. Burgemeestre · J. Hulstijn
Faculty of Economics and Business Administration, VU University Amsterdam, Amsterdam, The Netherlands

1 Introduction

A lack of trust can be expensive. Trust allows trusting parties to keep information and transaction costs low and to enter into mutually beneficial interaction. Trust has, therefore, started to be something of a holy grail in the world of information systems, web services, IT tools, and IT applications: if only users could distinguish warranted trust from unwarranted trust, in financial services, in accounting, in medical online information, and if only the IT environment could reliably capture and express trustworthiness.

In this paper, we consider trust in the economic and public domain, focusing on a particular case in information technology (IT) in which IT developers face the task of designing for trust. We analyze this task by means of a survey of what trust means in the economic and public domain, and by means of the emerging paradigm of value-sensitive design and existing engineering design methods. We argue that designing IT systems for trust requires that all parties concerned must be seen as clients for whom the systems are designed: if IT developers design an IT system for a company with the aim that the company can be trusted by a third party, then that third party should be involved as a second client for whom the system is designed. We analyze the overall and general value of trust in terms of series of more specific values for which the IT systems are to be designed, and we describe two design methods for doing so.

Our case originates in the context of a concrete problem in the economic domain, namely the transfer of control from authorities and government agencies to companies. There is currently a tendency to exercise control, monitoring, and inspection of the business of companies on a higher level of aggregation. Controlling on an item level, that is, monitoring or inspecting each and every container, each and every individual food product, or each and every financial transaction, is in many contexts typically no longer physically or economically possible. Governmental inspection agencies such as customs, tax, and food safety control agencies are, therefore, increasingly looking directly at the level of the companies themselves in order to determine which can be trusted to be in control of their business and to be in compliance with applicable regulations. In turn, companies are by this tendency now faced with the task of establishing this trust, and they typically turn to IT systems designs for fulfilling this task. Nevertheless, companies have difficulty designing for trust in this way, a difficulty we set out to analyze with philosophical and design methodological resources.

We start by introducing our case in Section 2, which is called system-based control as developed by the Dutch Tax and Customs Administration for controlling companies operating in the Netherlands. We analyze the task of designing IT systems that establish the required trust in Section 3, drawing from the model of trust by Lewicki and Bunker. Finally, we collect means to design for trust in Section 4 on value-sensitive design, and in Section 5 on design methodology.

2 System-Based Control

Governments have the responsibility to check whether companies are compliant with regulations in various areas such as health, safety, security, tax, and customs.

Typically, this control is exercised in a command-and-control fashion, where companies are required to provide large amounts of control data about their business to government agencies, and in addition to checking this data, the agencies have to perform all kind of inspections and checks on the businesses. Providing this data and facilitating the inspections have become such an administrative burden for companies that the European Commission has started an initiative to reduce the burden by 25%. Hence, in recent years, governments have been looking into new supervision models that build on the responsibility and participation of companies, also called *public–private partnerships*. Trust plays a prominent role in these new participative supervision models, and it acquires also a different meaning. In the traditional deterrence models, trust in the compliance of companies is built up by the agencies on the basis of their ongoing checks of the companies' business practices. In the new models, the companies themselves are required to establish the trust in their businesses, and companies are expected to do so by demonstrating that they themselves can be trusted to be *in control* of their businesses. Meeting these requirements poses a difficult challenge to companies. IT systems are currently the appropriate means for companies to provide information about their business practices to government agencies; with the traditional deterrence models, IT systems can be designed to make the required control data available. With the new participative models, however, companies are expected to generate trust of being in control, and it is currently unclear to companies how they can design IT systems that do this.

For analyzing the difficulties involved in designing IT systems for trust, we first introduce a specific case of a participative supervision model. In the next section, we focus on the shift in the meaning of trust that this model entails.

Our case is a governance strategy that is currently being explored by the Netherlands Tax and Customs Administration (Dutch Customs, for short) and that is called *system-based control*. In system-based control, the focus is less on the control of commercial transactions and movement of goods by a company, and more on the underlying procedures that the company has implemented to control its own business processes. System-based control can thus be considered as a form of enforcement that is based on mutual cooperation and trust between taxpayer and tax administration (Gribnau 2008; Kamerling and van der Putten 2007). Typically, these internal control procedures are made possible by enterprise information systems. The aim of system-based control is to enhance taxpayer compliance while at the same time achieving a more effective allocation of Dutch Customs resources. In system-based control, a company receives more responsibilities and is expected to interpret the legislation and to implement appropriate compliance measures and to provide evidence of its progress. The tax administration must enforce compliance by guiding the company and monitoring its progress. A penalty may be imposed only if progress is inadequate. In practice, Dutch Customs often works with a certification scheme. Companies that are able to demonstrate that they are in control of their business processes and that they are compliant may apply for a so-called *Authorized Economic Operator* (AEO) certificate or covenant that will give them less supervision and a lower administrative burden in return (European Commission 2007). Companies that do not have a certificate will also not receive these benefits and will be under stricter supervision, since they will be perceived by Dutch Customs as being unable to reach a sufficient level of compliance.

For system-based control, Dutch Customs must establish the reliability of the company's internal control framework. This forms the basis for their trust in the company's record keeping. Companies have some grasp of what reliability means in this context: the *Committee of Sponsoring Organizations of the Treadway Commission* (COSO) has provided a well-known standard for setting up an internal control framework, which typically applies to control frameworks implemented in enterprise information systems (COSO 1992). The standard recommends:

- A control environment where integrity and ethical values are supported by the top echelons of management throughout the organization.
- Risk assessment is performed to identify and manage risks relevant to the organization.
- Control activities such as policies, procedures, and processes are implemented to ensure that a company carries out management directives (examples include approvals, verifications, reconciliations, reviews of operating performance, security of assets, and segregation of duties).
- Relevant company data contained in the information system should be communicated in the organization and to the relevant stakeholders.
- Ongoing monitoring to assess the quality of a company's internal control systems.

In addition, the company must evaluate whether the proposed system has been implemented effectively. To provide some guidance on what is considered "effective implementation" customs refers to the COSO internal control guidelines (COSO 1992; European Commission 2007). The scores range from

0: "no control measures in place"
1: "internal control is ad hoc and unorganized"
2: "internal control has a structured approach"
3: "internal control is documented and known"
4: "internal control is subject to internal audits and evaluation"
5: "internal control measures are integrated into the business processes and continuously evaluated"

In turn, this scoring provides Dutch Customs with an indication of the maturity level of the company's self-controlling abilities.

Despite this COSO standard, it appears to be very difficult for companies to adapt their enterprise information systems so that they meet the value of being in control. From the various case studies conducted, it appears that the IT departments of these companies have great difficulties translating a high-level goal such as being in control into real IT applications, as required by the first recommendation of the COSO standard (Ayres and Braithwaite 1992; Burgemeestre et al. 2009, 2010). One reason for this difficulty is that there is no direct translation from abstract norms and values into IT systems and programming code. Hence, there seems to be a great demand for a methodology that IT experts can use when interpreting high-level norms and values for IT systems. A second reason is that in the case of participative supervision models, companies should design their IT systems for the value of being in control in such a way that government agencies like Dutch Customs trust the companies to be in control. When establishing that trust, merely designing for being

in control does not go far enough, as can be argued when looking at trust in more detail.

3 Three Stages of Trust

In their model, Lewicki and Bunker (1995, 1996) identify three sequential, cumulative stages of trust: *calculus-based trust*, *knowledge-based trust*, and *identification-based trust* (Table 1).

In the first stage, *calculus-based* trust is based on the consistency of companies' behavior and involves a continuous evaluation by government agencies leading to penalties for violations of trust and rewards for preserving it. This is the typical situation in the traditional command-and-control attitude of Dutch Customs, where they only trust companies after frequent inspections, and where they penalize companies if they detected a violation on inspection. This is a very minimalistic notion of trust.

The second stage of *knowledge-based* trust occurs when government agencies have enough knowledge about companies to understand them and to predict their likely behavior. A typical example of this type of trust is that Dutch Customs understands a company's economic reasoning and that it knows how the company makes tradeoffs between profit maximization against the cost of the penalties and the likelihood of fraudulent practices being detected. Dutch Customs can use this knowledge about the company's preferences to predict the company's behavior.

Identification-based trust, the highest level of trust, is based on the identification of the desires and intentions of companies by government agencies. This is the public–private partnership situation where a company is signaling to Dutch Customs that they are willing to balance their profit maximization with societal responsibility and good corporate governance by implementing measures and enterprise information systems for being in control of their own processes. In this case, Dutch Customs would be able to expect responsible behavior from the company and be more confident that the company will in fact pay the taxes and customs duties it owes. In this case, genuine trust exists, because there is a mutual understanding and appreciation of each others' norms and values. The mutual understanding has

Table 1 Trust basis and information needs of governments for each trust stage

Stage of trust development	Calculus-based	Knowledge-based	Identification-based
Trust basis	Consistency of behavior	Predictability of behavior	Identification and understanding
Information needs of governments to determine trust validity of companies	Actual behavior	Behavior in various contexts	Mutual understanding of each others' needs and desires
		Problem-solving strategies	Requirements for maintaining trust
		Needs, preferences, priorities	Commonly shared values

reached a level where parties are able to act on behalf of the other. This requires the company to espouse societal values and responsibility on the one hand (which relates to a company being in control of its own business processes). On the other hand, it requires Dutch Customs to understand that they should actively enable companies to maximize their profits. In particular, Dutch Customs should take the responsibility to reduce the transaction costs incurred by inspections and, hence, minimize the transaction costs of inspections of trusted companies, such as AEO certificate holders.

Since each stage has a different basis on which trust is built, the parties in the trust relationship use different information in each stage to determine the validity of the trust relationship. For the relationship to evolve to the next stage, sufficient information that supports the validity of the perceived trust must be gathered. The table below summarizes the trust bases and accompanying information needs of government agencies for each stage.

The model of Lewicki and Bunker makes clear that if trust in the full, genuine sense of identification-based trust is to be established between companies and governments in participative supervision models, then companies will have to do more than just making information about their businesses available. IT systems that merely make control data available would be sufficient for establishing calculus-based trust. IT systems that make business processes transparent and that make clear that companies are in control of these processes in accordance with the COSO standard, may easily establish simple knowledge-based trust by providing only for the information needs of government agencies in the second stage of trust. For full identification-based trust, the IT systems of companies should also incorporate and acknowledge the needs and values of these agencies. A complication in taking this last step is that it can be argued that incorporating governmental needs and values is something different from merely providing more or different information. These governmental needs and values should rather serve as input for designing the IT tool, which practically turns the designing of such tools into a project in which IT developers not only design a system for their companies but also for the government agencies concerned.

Designing for trust for being in-control in the eyes of Dutch Customs seems, therefore, to consist of designing a reliable enterprise information system, based on, say, principles of accountancy and EDP (e.g., principles such as segregation of duties, four-eyes-principle, accountability, (audit-)traceability, etc.) that meets the COSO standard. It also seems to involve accepting Dutch Customs as a primary client of the enterprise information system, with whom the company IT developers should argue and negotiate whether the system serves the client's purposes. Such arguments and negations moreover serve the dual purpose of exchanging norms and signaling that both parties care about these norms, in line with the classical analysis by McCauley (1963) of how the process of contract negotiation builds trust between companies. Companies and Dutch Customs exchange their norms when companies walk Dutch Customs through their business processes and present their internal control procedures; Dutch Customs indicates if they agree or not; and companies signal that they care about the values of Dutch Customs and how these values constrain the behavior of the company. This is precisely what identification-based trust focuses on in public–private partnerships.

4 Value-Sensitive Design

Designing IT systems for trust, or more generally, designing technical systems for high-level values such as trust, justice, fairness, and safety, implies that software developers and systems designers must see to it that systems inherit these values. Moreover, they have the task of demonstrating that the systems they design have these properties and that users see them as being imbued with the desired values.

In order to grasp the implications of the tasks for engineering when designing for trust, it is important to realize that in addition to separating the three cumulative stages of trust that we discussed above, full-fledged trust should be distinguished from mere epistemic confidence. Trust is a distinctively moral phenomenon, i.e., morality is constitutive of the phenomenon of trust. Trust between people is crucially concerned with assumptions or beliefs about the benevolence and moral motivation of others. In deciding to trust or in developing trust, individuals, therefore, typically look for evidence or reliable signals indicative of the moral motivation of the other party. Confidence is an epistemic category and, as such, is concerned with an estimation or prediction of the likelihood of a particular performance and behavior by a system or person. In this sense, one may trust one's plumber to fix the kitchen sink or to have a fair bit of confidence in her abilities as a plumber. She strikes one as competent if she has a documented history of doing good plumbing jobs in these cases, if she has brought all the right gear, if she seems to make sensible remarks about your problems, etc. This confidence is, however, different from trusting this plumber with the silverware in the kitchen drawer or trusting her not to overcharge you or to provide an honest statement of her competencies in the light of the tasks before her. If this construal of the core notion of full-fledged trust as implying proper moral motives and acting upon appropriate moral reasons is correct, then it is essential that factual information is supplemented with statements about moral beliefs and moral identity. A certain level of moral openness, transparency, and articulateness is required to go beyond confidence and establish trust.

We think that the notion of *value-sensitive design* is highly relevant to a better understanding of what is required in the context of designing for trust. Value-sensitive design (see Van den Hoven and Manders-Huits (2009) for an introduction) refers to an approach to ethics or dealing with value issues that aims firstly at expressing or incorporating values in technological and engineering design and, secondly, at analyzing technology in such a way as to show which values it expresses. This assumes that human values, norms, and moral considerations can be imparted to the things we make and use. The idea of value-sensitive design originates historically from three more or less independent lines of inquiry. The first is given by work done in the early 1990s by a group of Stanford University researchers dealing with Language, Computation, and IT, comprising Terry Winograd, Batya Friedman (who coined the term), and John Perry (Friedman 1997). This work showed that software and computer systems could easily come to contain biases, arbitrary assumptions, and the peculiar worldviews of their designers, which could then affect users in various ways. Research on biases in search technology and interfaces is a good example of their work. A second line concerns legal scholars who, at around the same time, observed that regulation in society was taking place by means of computer code and software. Code started to function as law and laws would in the

future literally be encoded, as Joel Reidenberg and Larry Lessig pointed out (see Lessing 1999, for this approach). Thirdly, advocates of so-called *privacy enhancing technology* at the Dutch and Canadian Data Protection Offices observed that technology itself, instead of traditional law, was probably the only way in which we could deal with privacy compliance and law enforcement issues given the increasing amount of dynamic privacy laws and regulations and the vast amount of data that are processed in our society (Borking and Raab 2001). They emphasized that privacy should be protected "by design." It is impossible to have lawyers check manually whether certain data practices are in breach or in compliance with the law. The software would in the long run have to take care of that on our behalf, and not only in the area of privacy.

These three developments are situated against a more general background of thinking about the value-ladenness of technology. In his 1980 political critique of technology, Winner (1980) showed that simple engineering structures (such as bridges and overpasses) could actually have political and moral consequences because they constrained the people using them. And in cognitive psychology, a new way of thinking emerged about the way technical systems immediately invite and facilitate certain behavior in those who experience them (Gibson 1979; Norman 1988).

Design for values requires that values are specified and exemplified. Reconstructed moral value concepts, e.g., democracy, justice, responsibility, privacy, may function as high-level architectural principles for the design of information systems and IT applications. These principles can be utilized as non-functional requirements, analyzed in lower level values, and eventually specified in functional specifications for the development of IT applications using, for instance, functional decomposition.

In this paper, we are looking at trust between organizational entities, whereas much research is concerned with trust relationships between individuals. In this latter research, the design perspective is emphatically taken into account. Bicchieri and Lev-On (2010) argue, for example, on the basis of extensive laboratory research and meta-analyses of trust games that communication preceding game play is highly relevant for establishing trust between players. They also find that the nature and richness of the communication channel is important, as is the content of the communication, e.g., is it dyadic or multi-actor, or is it about the strategic interaction or about completely different things. If these findings are also significant for establishing trust between organizational entities, then the design of interfaces for IT tools establishing e-trust becomes extremely important. Furthermore, designers are then faced with questions such as: which information about reputations, history, and identity should be made available; how much of this information should be made available; and when, i.e., in which phases of the usages of the tool, should this be done. Moreover, the question of how information should be presented must also be posed. "These design decisions can push subjects toward or away from trust and cooperation" (Bicchieri and Lev-On 2010).

The institutional and regulatory case that is central to our paper is not very different in this respect. We need to identify what is involved in this notion of trust and how systems in a broad sense may be designed so that they are conducive to the establishment of trust in this specific sense. Our description and analysis of the three cumulative stages of trust in the previous section and the distinction between full-fledged trust and mere epistemic confidence analyzed in this section already presents the main lower-level values that are relevant for designing IT tools for trust.

The design for *calculus-based trust* for example, expressly implies that a constant stream of relevant data must be made available by companies and that mechanisms must be in place that make tampering with data impossible. In addition, facilities for audits, monitoring, and checks are implied by this notion of trust. The design for *knowledge-based trust* goes one step further and requires, in addition to providing safeguards for the reliability of data, documentation of actions, actors, and agents, provisions for displaying the identity of companies, and the transparency of business processes themselves. Moreover, there should be a commitment in advance to repair errors in case of bad outcomes and violations of these requirements and to accept liability for these outcomes and violations. The highest rung of the ladder of trust is *identification-based trust*. This level of trust requires insight into the highest level of company strategy, establishing company values and missions and allowing key-queries identified by control-transferring governmental authorities and agencies. Companies should also be proactive in the value domain by initiating inquiries and exploring the value issue themselves. Companies are to engage in reflection on their own responsibility towards society in what could be called "spontaneous value reflexivity." Parties that have never thought about trust and responsibility and have not thought about how to realize this in their organization or are only prompted to such thinking when confronted by scandals, penalties, and negative media attention, seem less worthy of trust. A particular example of spontaneous value reflexivity that we observed in case studies was that of a large international chemical company with branches in various EU member states. The company reported to Dutch Customs that they themselves had discovered that—due to certain flaws in the cross-border VAT control procedures in the EU—there was a possibility that they were paying substantially less VAT than they should under applicable VAT legislation. Dutch Customs saw this as an indication that the company was taking care of their concerns and that this company was taking its societal responsibility seriously. It greatly enhanced the trust Dutch Customs placed in this company, and this is considered to be a prime example of system-based control.

5 Design Methods

We have argued that when companies design IT systems for establishing trust with government agencies, these systems need to incorporate the values identified in the previous section. According to the analysis of Section 3, designing for trust ultimately implies that these systems are not merely designed for the companies involved but that government agencies are also recognized as clients of the IT systems, rather than merely as their users. When turning to methods for designing and introducing the distinction between clients and users, the latter implication can be easily accommodated. According to descriptions of the design process as provided by design researchers, designing starts with a client expressing specific needs. The task of designers is then to develop a system that can realize these needs, taking into account additional design requirements such as efficiency, the technical state-of-the-art, manufacturability, and costs. Designing typically also includes communication with the clients about their needs, aimed at articulating and detailing the needs and adjusting them so that they can be realized given the additional

requirements of efficiency, the technical state-of-the-art, etc. Clients, as opposed to users, are thus those agents who order technical systems to be designed by engineers, and they are the agents with whom engineers collaborate when designing the systems. Clients come thus first in more than one meaning, where users come last, since users are agents who use the system without directly influencing its design (in special cases, agents may play both the active role of client and the passive role of user). The implication that IT systems have both the companies *and* government agencies as their clients then means that both parties are involved in the communication with the designers about what the IT systems need to do.

Our introduction of value-sensitive design as a new approach to designing suggests that existing design methods are not yet suited for designing for values. Designing may indeed seem a value-neutral activity, for instance when it is taken as a primarily instrumental process in which designers merely develop systems in response to needs of clients. The values involved in these systems are then only values held by the clients, and values for which users manipulate the systems. Yet, even in an instrumentalistic approach like this, values are involved in the design process; safety value is one example. In a more general sense, designers have ample experience with designing for ergonomic values of systems and for the values of buyers of more commercial systems. When cars or household products are redesigned, for instance, designers have methods to let the products better fit the needs and desires of buyers and customers. Hence, the values identified in the previous section as relevant to IT systems designed for trust may simply be incorporated as additional needs that the designers of these systems have to address when they are developing IT systems.

Without aiming to list all possible methods designers have to their disposal, we can characterize designing as primarily an activity in which new design problems are solved on the basis of *existing design solutions*. In the case of the redesigning of existing products, this use of existing design solutions is obvious. Redesigning starts with an existing design and aims at an improved version on the basis of the existing design. Even when designing new products or systems, however, designers draw on their knowledge of existing design solutions. In the methods by Pahl et al. (2007), designers are initially required to translate client needs into functional requirements (and further requirements) in a conceptual phase and then to analyze these functional requirements into a network of subfunctions. Existing design solutions for these subfunctions are then used to come up with the overall design solution. In studies of more creative, innovative design processes aimed at solving ill-defined design problems, designers are described as structuring these problems and making them manageable by choosing solution directions that originate in the designers' experience with past design problems (e.g., Cross 2006). A first conclusion is, therefore, that designing IT systems for trust is better not understood as entailing designing completely new IT systems; what will happen rather is that existing enterprise information systems will be redesigned for trust or that components of existing IT systems will be reassembled for trust. In addition, design research provides various means to support such design and redesign processes, means which also allow for incorporating values. We now end with sketching two design methods in some detail: *Quality Function Deployment* as a rather structured method for determining starting points for redesigning and the less structured description of more creative designing as the co-evolution of design problems and their solutions.

Quality Function Deployment (QFD) is primarily a tool for adjusting existing products to let them better meet the requirements of customers (King 1989; Akao 1990). In QFD, one typically starts by identifying the relevant users of a product to be redesigned and by determining what these users value in the product. This valuation sets user requirements and these requirements are then employed by designers to identify the quantitative characteristics of the product that are relevant to meeting the user requirements, acknowledging that there is a difference between the way in which users formulate their requirements and the way in which designers describe the products they design. The relationship between these user requirements and product characteristics is determined one by one, which may be done in a qualitative or quantitative way, using symbols or values like 0, 1, 3, and 9. As such, QFD results in a matrix (see Fig. 1), in which by convention user requirements (with weight factors that indicate their relative relevance) are listed in rows, and the product characteristics are listed in columns (with a possible prioritization). This matrix is topped by a roof which contains the positive and negative correlations between the product characteristics (the QFD term for the matrix is the "house of quality"). The method allows more information about users and product to be added, like users' appreciation of competitors' products, the (quantitative) values of the characteristics of those other products, and the target (quantitative) values of the product characteristics of the product to be designed (which are typically determined in contrast to the competitors' products). Still, the matrix as shown in Fig. 1 already provides ample information to designers about how to improve the product: it provides information on what users value, what characteristics are relevant for improving the appreciation of the product by these users, and how changes in these characteristics depend on one another. Moreover, it is part and parcel of the QFD

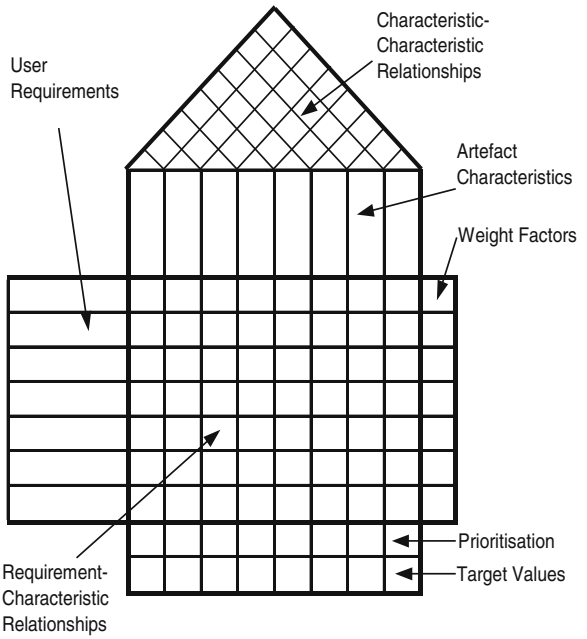

Fig. 1 The "house of quality" in QFD

method that the users themselves express what they value in the product in their own nontechnical terms, as is typically revealed by marketing research; designers are merely required to identify the relevant product characteristics and their relations to these user requirements.

For the case of designing IT systems for trust, the QFD method is readily available for redesigning existing enterprise information systems. Application of this method would mean that the values derived from trust as identified in the previous section are listed as user requirements and that IT developers analyze which of the characteristics of their existing systems are relevant to meeting these values. This analysis then can be used for redesigning enterprise information systems to better meet the values related to trust.

Redesigning enterprise information systems on the basis of QFD may, in principle, be a simple matter of adjusting the characteristics relevant for trust, and then evaluating and detailing the resulting new systems, for instance by invoking the users to assess the new systems once again. This simple linear approach to designing need not always work, which can be made clear with the second model of more creative designing as the co-evolution of design problems and their solutions. In the second model, it becomes clear that a change in a solution to a design problem may also create a change in the problem, creating new requirements that users may bring up in assessing the solution. The designing or redesigning of enterprise information systems may then lead to an ongoing cycle of adjustments in the systems and adjustments in the requirements government agencies put on these systems.

In this second model, designers start by considering a design problem that does not suggest an immediate solution. Designers have to develop the solution, and in doing so, drawing from their past experiences with designing, they not only create a solution or solution direction but also change the original problem. Designing becomes in this way a process in which a "space" of design solutions co-evolves with a "space" of design problems (Cross and Dorst 1998; Cross 2006). The choice for a specific general solution to a problem may put that problem in a new light, yielding a more clear understanding of that problem, with the subsequent consequence that another general solution should be sought. Or a solution to a part of the original problem may yield to that better understanding of the problem. Designing then becomes a succession of syntheses of solutions and analyses of those solutions and the effects they have on the problem. The designer looks for a problem–solution pair that "matches," in that the solution is an acceptable response to the way in which this solution sets down the original problem. Finding that specific pair is the creative leap in designing, when it is recognized that a tenable map is found that bridge the co-evolving problem and solution spaces.

This model does not provide cookbook recipes for guiding the design of IT systems that establish trust with government agencies, but it may provide the proper way of understanding this design, for instance by explaining the iterations needed. Merely adjusting existing enterprise information systems such that they better meet the requirements set by government agencies may not yet be the creative leap that companies need to take to satisfy these agencies. In this model, an adjustment like this may change the problem agencies have, inciting new requirements from the agencies. Actually, our cases studies showed that the fundamental problem is that the IT developers are typically not experts on interpreting high-level values as to what it

means for a company to be in-control. On the other hand, the EDP auditors from Dutch Customs may very well be experts at analyzing in-control issues, but they typically do not have the detailed knowledge that IT experts from the company do to know all the relevant business process. To some extent, redesigning an IT system to be in-control has almost the same software complexity level as solving the infamous millennium-bug problem that happened at the turn of the century. What we observed in the case studies was a complex negotiation process that evolved between the experts of the company and of Dutch Customs, which is probably best analyzed as the abovementioned process in which a "space" of design solutions co-evolves with a "space" of design problems. Typically, in the case studies, there is a co-evolution process taking place in which the company proposes a space of redesign options for their IT systems to become in-control, and Dutch Customs incrementally reformulates its interpretation of how they view in-control in the specific context of this company. The process we observed also had elements of co-creation of a space of shared values related to the issue of being in-control, where the contributions of the company and Dutch Customs are equally important. Another interesting observation is that Dutch Customs explicitly stated that they wanted to be acknowledged by the company in this negotiation process not as an opponent, but as one of their key stakeholders, reflecting the societal responsibility of the company. Hence, the outcome of this process is not only an improved IT system for being in-control in the company but more importantly the process of co-creation of these shared values. These shared values constitute the foundation of the enhanced trust-based relationship in system-based control. Hence, what is needed is that companies design their IT systems for trust, acknowledging the government agencies as clients *and* as users of the systems, with requirements that can be related to these systems with tools like QFD, and with requirements that can change as a result of the designing of the IT systems.

6 Conclusions

In this paper, we considered trust in the economic and public domain, analyzing the task of designing systems for trust in information technology, focusing on our specific case of companies and the trust relationship with Dutch Customs through information technology systems. We introduced the model by Lewicki and Bunker to separate the three cumulative stages of calculus-based trust, knowledge-based trust, and identification-based trust, showing that each stage requires the provision of additional types of information for establishing trust. We distinguished trust from mere epistemic confidence, showing that for establishing trust over mere confidence a certain level of moral openness, transparency, and articulateness is required. We introduced the paradigm of value-sensitive design and two engineering design methods to argue that information technology developers and engineers can design for high-level values such as trust: these high-level values can be analyzed in terms of lower level values that can be more easily incorporated as requirements in designing, and existing design methods such as quality function deployment already provide means to design for values. Moreover, in design methods, it is standard that engineers argue and negotiate with their clients, suggesting that both parties in the

trust relationship should be seen as clients for whom information technology systems are designed.

Our analysis of the task of designing for trust may be taken as a case of value-sensitive design. In this approach, the high-level value of trust is examined and broken down into lower-level values and requirements that can be used by engineers in their decisions in designing technical systems, thus guiding engineers when designing such systems and helping to avoid a situation in which the designed systems, once completed, are judged afterward to be deficient with regard to values. Philosophical accounts and analyses of these values have shown to be instrumental to designing for trust, as may be expected; what was perhaps less expected is that we showed that engineering design methods already incorporate means for designing for trust.

Acknowledgement We would like to thank Marc de Vries and acknowledge his authorship of Fig. 1. The research by Pieter Vermaas was supported by the Netherlands Organization for Scientific Research (NWO).

References

Akao, Y. (Ed.). (1990). *Quality function deployment: Integrating customer requirements into product design*. Cambridge: Productivity.

Ayres, I., & Braithwaite, J. (1992). *Responsive regulation: Transcending the deregulation debate*. Oxford: Oxford University Press.

Bicchieri, C., & Lev-On, A. (2010). Studying the ethical implications of e-trust in the lab. *Ethics and Information Technology*, in press

Borking, J. J., & Raab, C. D. (2001). Laws, PETs and other technologies for privacy protection. *The Journal of Information, Law and Technology*, 2001 (1) http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2001_1/borking

Burgemeestre, B., Hulstijn, J., & Tan, Y.-H. (2009). Rule-based versus principle-based regulatory compliance. In G. Governatori (Ed.), *Proceedings of JURIX 2009* (pp. 37–46). Amsterdam: IOS.

Burgemeestre, B., Hulstijn, J., & Tan, Y.-H. (2010). Towards an architecture for self-regulating agents: A case study in international trade. *Lecture Notes in Computer Science, 6069*, 320–333.

Committee of Sponsoring Organizations of the Treadway Commission, COSO (1992). COSO. Internal control integrated framework. Technical report

Cross, N. (2006). *Designerly ways of knowing*. London: Springer.

Cross, N., & Dorst, K. (1998). Co-evolution of problem and solution spaces in creative design: Observations from an empirical study. In J. Gero & M. L. Maher (Eds.), *Computational models of creative design IV* (pp. 243–262). Sydney: University of Sydney.

Gribnau, H. (2008). Soft law and taxation: the case of the Netherlands. *Legisprudence, 1*, 291–326.

European Commission. (2007). *AEO guidelines, technical report TAXUD/2006/1450*. Brussels: General Taxation and Customs Union.

Friedman, B. (Ed.). (1997). *Human values and the design of computer technology, CSLI lecture notes 72*. Cambridge: Cambridge University Press.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston: Houghton-Mifflin.

Kamerling, R. N. J., & van der Putten, J. A. M. (2007). *Tax auditing in the Netherlands*. Usselo: Dutch Tax and Customs Administration.

King, B. (1989). *Better design in half the time: implementing QFD in America*. Methuen: GOAL/QPC.

Lessing, L. (1999). *Code and other laws of cyberspace*. New York: Basic.

Lewicki, R. J., & Bunker, B. B. (1995). Trust in relationships: a model of trust development and decline. In B. B. Bunker & J. Z. Rubin (Eds.), *Conflict, cooperation and justice* (pp. 133–174). San Francisco: Jossey-Bass.

Lewicki, R. J., & Bunker, B. B. (1996). Developing and maintaining trust in work relationships. In R. M. Kramer & T. R. Tyler (Eds.), *Trust in organizations: frontiers of theory and research* (pp. 114–139). Thousand Oaks: Sage.

McCauley, S. (1963). Non-contractual relations in business: a preliminary study. *American Sociological Review, 28*(1), 55–67.

Norman, D. A. (1988). *The psychology of everyday things*. New York: Basic.

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. H. (2007). *Engineering design: a systematic approach* (3rd ed.). London: Springer.

Van den Hoven, J., & Manders-Huits, N. (2009). Value sensitive design. In J. K. B. Olsen, S. A. Pedersen, & V. F. Hendricks (Eds.), *A companion to the philosophy of technology* (pp. 477–481). Chichester: Wiley Blackwell.

Winner, L. (1980). Do artifacts have politics? *Daedalus, 109*, 121–136.