Is professional practice at risk following the Volkswagen and Tesla motors revelations?

Simon Rogerson
De Montfort University

Abstract

With the world in economic crisis the headlong drive for efficiency and effectiveness together with resulting profit is the watchword. Such pressure might have resulted in real gains but has also led to unscrupulous or reckless actions. The tempering of such drive with ethical consideration is often neglected until there is a detrimental event causing public outcry which in turn places pressure on the actors to account for the reasons the event had occurred. This cause and effect map is commonplace. The Volkswagen emissions scandal and Tesla Motors public beta testing of the Autopilot software in their cars illustrate the drive for efficiency and effectiveness without proper ethical consideration. This paper focuses on the role of software engineers in such situations. An ethical analysis of the two cases is presented using the Software Engineering Code of Ethics and Professional Practice. The findings, together with previously published analyses, are used to draw general conclusions and recommendations about the efficacy of the software engineering profession.

Keywords: professional ethics, software engineering, safety critical software, defeat software, autonomous vehicles, Volkswagen, Tesla.

Introduction

Each day society becomes more and more technologically dependant. Some argue that as a consequence society becomes more and more vulnerable to catastrophe. With the world in economic crisis the headlong drive for efficiency and effectiveness (and resulting profit) is the watchword. Such pressure might have resulted in real gains but has also led to unscrupulous or reckless actions. The tempering of such drive with ethical consideration is often neglected until there is a detrimental event causing public outcry. Such an event will usually attract both media interest and social media posting which in turn places more and more pressure on the actors to account for the reasons why the event had occurred. This cause and effect map is commonplace. Consider, for example, transport which is a fundamental element of the fabric of society. In this area there have been two recent events which illustrate the drive for efficiency and effectiveness without proper ethical consideration. The first example is the Volkswagen emissions scandal which came to light in September 2015. The company installed software into millions of vehicles with diesel engines so that impressive emission readings would be recorded in laboratory conditions even though the reality is that the diesel engines do not comply with current emission regulations. The second example concerns Tesla Motors and the public beta testing of the Autopilot software in their cars. In May 2016 there was a fatal accident when a Model S Tesla under the control of the Tesla...
Autopilot software drove at full speed under a trailer resulting in the driver of the Tesla being killed.

Both examples centre on the use of software which is the focus of this paper. Both are pieces of safety critical software which is defined as software that, if it fails or malfunctions, will cause death or serious injury to people, and/or result in loss or severe damage to property or equipment, and/or cause environmental harm. The development of application software does not occur in a vacuum. Within Volkswagen and Tesla there will have been a complex network of individuals involved in decision making at different levels resulting in the production of application software which achieved a particular goal. The software engineers who wrote the software may or may not have been privy to higher level decisions and the associated reasons why such decisions were taken. But it is the software engineer who can ultimately be identified as the creator of the software and so rightly or wrongly can be held responsible for any unfavourable outcomes.

The aim of this paper is to undertake an ethical analysis of each case study using existing published accounts. Over 80 sources have been analysed but only some key sources are specifically referenced in this paper. This broad literature establishes confidence in the facts of each case described below. The ethical analysis is undertaken from a software engineering perspective through performing a professional standards analysis within the case analysis method as defined by Bynum (2004). The Software Engineering Code of Ethics and Professional Practice of the ACM (see http://www.acm.org/about/se-code) is used in this analysis as it is regarded as the most applicable set of principles for these cases. It is long established; documenting the ethical and professional obligations of software engineers and identifying the standards society expects of them (Gotterbarn, Miller & Rogerson, 1999). The focus of the paper aligns with the following statement within the preamble to the Code, “These Principles should influence software engineers to consider broadly who is affected by their work; to examine if they and their colleagues are treating other human beings with due respect; to consider how the public, if reasonably well informed, would view their decisions; to analyze how the least empowered will be affected by their decisions; and to consider whether their acts would be judged worthy of the ideal professional working as a software engineer. In all these judgments concern for the health, safety and welfare of the public is primary; that is, the ‘Public Interest’ is central to this Code.”

The two case analyses highlight a set of key issues which need to be addressed if professional integrity within software engineering is to be protected and promoted. The findings are compared with previously published analyses to ascertain common and conflicting outcomes. The paper concludes by identifying general issues which underpin guidance for future software engineering practice.

The Volkswagen case study

Combustion engines are the source of pollution and therefore have been subjected to emission control. The formation of NOx (nitrogen oxides) through combustion is a significant contributor to ground-level ozone and fine particle pollution. In congested urban areas motor vehicle traffic can result in dangerous levels of NOx emission. The inhalation of fine particles can damage lung tissue and cause or worsen respiratory conditions such as asthma, emphysema and bronchitis. It can aggravate existing heart disease. Children, the elderly and people with pre-existing respiratory disease are particularly at risk. The regulations in place aim to reduce pollution through NOx emission and thus reduce health risks.
The statement issued by the US Department of Justice (2017) details the facts of the Volkswagen emissions scandal. Two senior managers, Jens Hadler and Richard Dorenkamp, appear to be at the centre of the so-called defeat software’s ongoing design and implementation processes. It states, “… in 2006, Volkswagen engineers began to design a new diesel engine to meet stricter U.S. emissions standards that would take effect by model year 2007. This new engine would be the cornerstone of a new project to sell diesel vehicles in the United States that would be marketed to buyers as ‘clean diesel,’ a project that was an important strategic goal for Volkswagen’s management. When the co-conspirators realized that they could not design a diesel engine that would both meet the stricter NOx emissions standards and attract sufficient customer demand in the U.S. market, they decided they would use a software function to cheat standard U.S. emissions tests. … Volkswagen engineers working under Dorenkamp and Hadler designed and implemented a software to recognize whether a vehicle was undergoing standard U.S. emissions testing on a dynamometer or it was being driven on the road under normal driving conditions. The software accomplished this by recognizing the standard published drive cycles. Based on these inputs, if the vehicle’s software detected that it was being tested, the vehicle performed in one mode, which satisfied U.S. NOx emissions standards. If the software detected that the vehicle was not being tested, it operated in a different mode, in which the vehicle’s emissions control systems were reduced substantially, causing the vehicle to emit NOx up to 40 times higher than U.S. standards. … Disagreements over the direction of the project were articulated at a meeting over which Hadler presided, and which Dorenkamp attended. Hadler authorized Dorenkamp to proceed with the project knowing that only the use of the defeat device software would enable VW diesel vehicles to pass U.S. emissions tests.” Drawing upon the ‘Statement of Facts’, Leggett (2017) reported that whilst there had been some concerns over the propriety of the defeat software all those involved in the discussions including engineers were instructed not to get caught and furthermore to destroy related documents.

According to Mansouri (2016) Volkswagen is an autocratic company with a reputation for avoiding dissent and discussion. It has a compliant business culture where employees are aware that underperformance can result in replacement and so management demands must be met to ensure job security. The Volkswagen Group Code of Conduct (2010) seems to promote this culture of compliance. Three statements align with the ongoing conduct encouraged during the emissions debacle.

Promotion of Interests (ibid., p15) “Each of our employees makes sure that their conduct and opinions expressed in public do not harm the reputation of the Volkswagen Group.”

Secrecy (ibid., p16) “Each of our employees is obligated to maintain secrecy regarding the business or trade secrets with which they are entrusted within the scope of the performance of their duties or have otherwise become known. Silence must be maintained regarding work and matters within the Company that are significant to the Volkswagen Group or its business partners and that have not been made known publicly, such as, for example, product developments, plans, and testing.”

Responsibility for Compliance (ibid., p22) “Each of our employees who do not conduct themselves consistently with the Code must expect appropriate consequences within the scope of statutory regulations and company rules that can extend to termination of the employment relationship and claims for damages.”

ORBIT Journal DOI:
The use of defeat software was discovered by accident (Rufford & Tobin, 2016). In 2013 the EPA in the USA commissioned West Virginia University to check the emissions of three diesel cars. Two happened to be VWs. The laboratory results were compliant. However, on road tests both VWs were emitting up to 38 times the permitted levels of NOx. The results were reported to the EPA and subsequent further investigations led to the discovery of installed defeat software and the legal action which followed.

On 11 January 2017, the US Justice Department announced that, “Volkswagen had agreed to plead guilty to three criminal felony counts, and pay a $2.8 billion criminal penalty, as a result of the company’s long-running scheme to sell approximately 590,000 diesel vehicles in the U.S. by using a defeat device to cheat on emissions tests mandated by the Environmental Protection Agency (EPA) and the California Air Resources Board (CARB), and lying and obstructing justice to further the scheme.” (US Department of Justice (2017))
Professional standards analysis

According to principle 1.03 of the Software Engineering Code of Ethics and Professional Practice, software engineers should “approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy, or harm the environment. The ultimate effect of the work should be to the public good.” The defeat software is clearly unsafe given NOx pollution damages both health and the environment. The public were under the misapprehension that VW cars were emitting low levels of NOx and therefore not a health risk. Software engineers installed the defeat software in violation of this principle.

According to principle 1.04 software engineers should “disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.” There is no evidence that any software engineer disclosed. Given that there must have been a large number working on this project there appears to be a widespread violation of this principle.

According to principle 1.06 software engineers should “Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.” The emissions software was heralded publically as a success when internally there was widespread knowledge that this claim was fraudulent. Software engineers were likely to have been privy to this cover-up and so violated this principle.

According to principle 2.07 software engineers should “Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.” There is some evidence that there was concern raised about the efficacy of the defeat software but it seems those in dissent allowed themselves to be managed towards deception. Once again a principle was ultimately violated.

According to principle 3.03 software engineers should “Identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects.” The EPA regulations are explicit and are legally binding. From the evidence accessed it is unclear as to whether software engineers knew of the illegality of their actions. Nevertheless ignorance cannot be and must not be a form of defence. Hence the principle was violated.

According to principle 6.06 software engineers should “Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest.” This relates to the analysis under principle 3.03. Compliance to further the prosperity of Volkswagen was at the expense of legal compliance.

According to principle 6.07 software engineers should “Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.” Software engineers could argue internally that the software indeed performed as it was designed to. However, the design was to achieve regulatory and public deception which is a violation of this principle.

According to principle 6.13 software engineers should “Report significant violations of this Code to appropriate authorities when it is clear that consultation with people involved in
these significant violations is impossible, counter-productive or dangerous.” Given the apparent corporate culture within Volkswagen there was little point in reporting concerns further up the line. In fact the corporate code seems at odds with the professional code regarding this point. Software Engineers failed to report these breaches to appropriate authorities.

**Other analyses**

Much has been written about the Volkswagen emissions case. Many of the accounts focus on business ethics with only a few touching upon the role of the software engineers in this situation. These accounts at times are repetitive but intertwine to provide a rich view which is discussed here. There are two recurrent issues.

The first issue is whistleblowing. Software engineers are faced with a challenging landscape in which some issues are difficult to identify whilst others are easily identifiable. The Volkswagen case falls into the latter category because of the legal constraints established through EPA regulations. Plant (2015) suggests that in this situation the software engineers should have alerted external bodies since the internal lines of reporting were compromised. Merkel (2015) concurs citing the Software Engineering Code of Ethics and Professional Practice by way of justification, and adds that the lack of whistleblowers in such a large group is surprising. Both authors point to the potential personal cost of whistleblowing as the reason it did not happen. The second issue adds weight to this argument.

Rhodes (2016) argues that corporate business ethics is very much a pro-business stance which is implemented through corporate control and compliance systems, and instruments of managerial coordination. This can enable the pursuit of business self-interest through organised widespread conspiracies involving lying, cheating, fraud and lawlessness. This is what happened at Volkswagen. Queen (2015) concurs explaining that Volkswagen intentionally deceived those to whom it owed a duty of honesty. The pressure for continuous growth and the perception that failure was not an option (Ragatz, 2015) created a culture where corporate secrecy was paramount which in turn implicitly outlawed whistleblowing.

**The Tesla case study**

The American Tesla Motors is currently the world’s second largest plug-in electric car manufacturer. According to the Tesla website (https://www.tesla.com/autopilot): ‘All Tesla vehicles produced in our factory, including Model 3, have the hardware needed for full self-driving capability at a safety level substantially greater than that of a human driver’; ‘Enhanced Autopilot adds these new capabilities to the Tesla Autopilot driving experience. Your Tesla will match speed to traffic conditions, keep within a lane, automatically change lanes without requiring driver input, transition from one freeway to another, exit the freeway when your destination is near, self-park when near a parking spot and be summoned to and from your garage.’; and ‘Once on the freeway, your Tesla will determine which lane you need to be in and when. In addition to ensuring you reach your intended exit, Autopilot will watch for opportunities to move to a faster lane when you're caught behind slower traffic. When you reach your exit, your Tesla will depart the freeway, slow down and transition control back to you.’

In 2016 Tesla came under scrutiny following a fatal accident involving a Model S Tesla under the control of the Tesla Autopilot. According to Lambert (2016) the attending police officer reported “On May 7 at 3:40 p.m. on U.S. 27 near the BP Station west of Williston, a 45-year-
old Ohio man was killed when he drove under the trailer of an 18-wheel semi. The top of Joshua Brown’s 2015 Tesla Model S vehicle was torn off by the force of the collision. The truck driver, Frank Baressi, 62, Tampa was not injured in the crash.” Tesla issued a statement that, “the vehicle was on a divided highway with Autopilot engaged when a tractor trailer drove across the highway perpendicular to the Model S. Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied. The high ride height of the trailer combined with its positioning across the road and the extremely rare circumstances of the impact caused the Model S to pass under the trailer, with the bottom of the trailer impacting the windshield of the Model S. Had the Model S impacted the front or rear of the trailer, even at high speed, its advanced crash safety system would likely have prevented serious injury as it has in numerous other similar incidents” (ibid.).

The Tesla statement (ibid.) addressed the issue of Autopilot software explaining, “It is important to note that Tesla disables Autopilot by default and requires explicit acknowledgement that the system is new technology and still in a public beta phase before it can be enabled. When drivers activate Autopilot, the acknowledgment box explains, among other things, that Autopilot ‘is an assist feature that requires you to keep your hands on the steering wheel at all times,’ and that ‘you need to maintain control and responsibility for your vehicle’ while using it. Additionally, every time that Autopilot is engaged, the car reminds the driver to ‘Always keep your hands on the wheel. Be prepared to take over at any time’. The system also makes frequent checks to ensure that the driver’s hands remain on the wheel and provides visual and audible alerts if hands-on is not detected. It then gradually slows down the car until hands-on is detected again.

National Highway Traffic Safety Administration (NHTSA) investigated the incident. NHTSA (2017) has recently published its findings. It confirmed that Tesla car was being operated in Autopilot mode at the time of the collision; that the Automatic Emergency Braking (AEB) system did not provide any warning or automated braking for the collision event; and that the driver took no braking, steering or other actions to avoid the collision. NHTSA found no defects in the design or performance of the AEB or Autopilot systems of the subject vehicles nor any incidents in which the systems did not perform as designed. The report states that Tesla Autopilot is an SAE Level 1 automated system and becomes Level 2 when Auto steer is activated. The SAE’s 6 levels of driving automation for on-road vehicles are shown in figure 1. As can be seen Tesla Autopilot is not classified as an automated driving system which monitors the driving environment.

| SAE level | Name                  | Narrative Definition                                                                 |
|-----------|-----------------------|--------------------------------------------------------------------------------------|
| Human     | Human driver monitors the driving environment | the full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems |
| 0         | No Automation         |                                                                                      |
| 1         | Driver Assistance     | the driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task |
| 2         | Partial Automation    | the driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration |
using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task

Automated driving system monitors the driving environment

3  Conditional Automation  the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene

4  High Automation  the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene

5  Full Automation  the full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver

Figure 1: Levels of driving automation for on-road vehicles (source: SAE International (2014))

According to Jaynes (2016) even though Autopilot is classified as Level 2 in practice, to many, it feels like Level 4. In these circumstances a driver can be lulled into a false sense of security and become distracted. The perception is further strengthened by the brand name Autopilot which implies a fully autonomous system not the semi-autonomous system which it is in reality (Roberts, 2016). Furthermore, simply having a warning that Autopilot is a beta test version does not convey the seriousness of system failure to the driver (Jaynes, 2016). Indeed, Solon (2016) explains that this is contrary to traditional car manufacturers where public beta testing software which relates directly or indirectly to safety is never used. Tesla collects in real time technical and personal data from its customers’ vehicles using this data to test the effectiveness of new software which it then, often secretly, installs into the vehicle for road testing albeit without controlling the vehicle (Simonite, 2016). On the Tesla website within a subsection of the legal terms and conditions there are statements confirming these practices (https://www.tesla.com/en_GB/about/legal#privacy-statement).

Tesla’s corporate culture is manifest within its Code of Business Conduct and Ethics (Tesla, 2010). In stark contrast to the Software Engineering Code of Ethics and Professional Practice, there is no explicit reference to health, safety and welfare of the public. Rather the code’s focus is explicitly the health, safety and welfare of Tesla. The implication appears to be that employees are expected to put Tesla first.

Professional standards analysis

According to principle 1.03 of the Software Engineering Code of Ethics and Professional Practice, software engineers should “approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy, or harm the environment. The ultimate effect of the work should be to the public good.” Autopilot software falls within the relatively new family of autonomous vehicle software. No evidence was found to suggest that stringent safety critical software engineering standards were in place. If this is true it is a poor risk mitigation strategy. Safety, quality of life and privacy could all be compromised, and software engineers are in violation of this primary principle.
According to principle 1.06 software engineers should “Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.” It appears that the use of moral algorithms and a public beta test regime is underplayed regarding potential dangers and overplayed regarding potential benefits. If this is the case it is unacceptable and software engineers are obligated to challenge this misconception.

According to principle 2.07 software engineers should “Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.” It is unclear as to whether software engineers consider and subsequently report on issues of social concern in developing autonomous vehicle software. Given the potential social impact of such software the lack of transparency is surprising, disappointing and unacceptable.

According to principle 3.10 software engineers should “Ensure adequate testing, debugging, and review of software and related documents on which they work.” Autopilot software is safety critical software and yet beta testing was, and continues to be, undertaken in the operational environment i.e. the public highway. There is no evidence to suggest that Autopilot software is defined as safety critical with Tesla. This leads to the question of adequacy of process. Software engineers need to be vigilant in ensuring adequacy bearing in mind public welfare is paramount. There is some doubt as to whether software engineers are adhering to this principle.

According to principle 3.12 software engineers should “Work to develop software and related documents that respect the privacy of those who will be affected by that software.” By default Tesla collect a wide range of personal data on the basis that this might be useful in testing existing and future software. This approach is contrary to international data privacy conventions and hence software which enables such personal data capture is questionable. Therefore software engineers may be contravening this principle.

According to principle 4.01 software engineers should “Temper all technical judgments by the need to support and maintain human values.” Using utilitarian moral algorithms in autonomous vehicle software has been proven to be technically feasible. However the moral justification is open to debate since other human values should be taken into account beyond simple calculus. Software engineers have an obligation to ensure this. Given such moral algorithms are in use, it follows that software engineers may well be contravening this principle.

According to principle 6.07 software engineers should “Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.” The branding and marketing of Autopilot suggest to some members of the public that the software is greater than SAE Level 2. Such perceptions are dangerous. Tesla has an obligation to ensure realistic public perceptions pervade. As employees of Tesla software engineers must share this obligation.

**Other analyses**

There are many accounts about autonomous vehicles which relate to the cars manufactured by Tesla. Whilst many issues are identified, two appear prevalent; public beta testing of
software and the use of moral algorithms in software. This broad body of literature has been drawn up in the discussion which follows.

Autonomous vehicles will almost certainly crash and the moral algorithm in the controlling software will affect the outcomes. Currently these algorithms use utilitarian moral decision making. However, Goodall (2014) argues that there is no obvious way to encode effectively human morality in software. If so, moral algorithms are reliant upon an implied higher machine pseudo-intelligence. Bonnefon, Shariff & Rahwan (2016) found that moral algorithms create a social dilemma. Even though people seem to agree that everyone would be better off if autonomous vehicles were utilitarian (in the sense of minimizing the number of casualties on the road), they all have a personal incentive to ride in autonomous vehicles that will protect them at all costs. McBride’s analysis (2016) concludes that the real ethical worth is in how autonomous vehicles enable people to connect, interact and strengthens communities which might have been difficult or impossible before. If true this adds a further complexity to the design of moral algorithms. Indeed Lin (2013) argues that utilitarian ethics is naïve and incomplete; rights, duties, conflicting values, and other factors should be taken into account. This is in line with the Federal Automated Vehicles Policy (2016, p26) which states that “Algorithms for resolving these conflict situations should be developed transparently using input from Federal and State regulators, drivers, passengers and vulnerable road users, and taking into account the consequences on others.”

Public Beta testing requires the testers to report back their findings to the developers (ISTQB) which implies a dialogue between tester and developer. The Tesla approach is based on covert testing and so the testing public do not and cannot enter into a dialogue. The question which needs to be answered is to the public beta testing regime of Tesla comply with the System Safety Policy (p20) which requires companies to, “follow a robust design and validation process based on a systems-engineering approach with the goal of designing autonomous vehicle systems free of unreasonable safety risks. Thorough and measurable software testing should complement a structured and documented software development process.”

As discussed earlier there is concern about driver’s perception as to the sophistication of Autopilot. This is addressed in the Human Machine Interface Policy (p22) which in part focuses on the boundary between Level 2 systems and Level 3 systems. It states, “Manufacturers and other entities should place significant emphasis on assessing the risk of driver complacency and misuse of Level 2 systems, and develop effective countermeasures to assist drivers in properly using the system as the manufacturer expects. Manufacturers and other entities should assume that the technical distinction between the levels of automation (e.g., between Level 2 and Level 3) may not be clear to all users or to the general public. Manufacturers and other entities should develop tests, validation, and verification methods to assess their systems for effective complacency and misuse countermeasures.”

**Synthesis**

The two cases highlight the dangers of an autocratic, hierarchical business structure which focuses on compliance rather than values. There is little opportunity to raise objections and share concerns about operational directions. Staff development regarding social values is likely to be limited. The nature of software impact is likely to be misunderstood and underestimated.

Addressing safety critical software is problematic in these business structures. In the fluid environment of application software, practice and process rapidly change. For safety critical
software McDermid & Kelly (2006) recommend a range of approaches for software safety rather than applying dogmatically a standard prescriptive approach. This would promote a wider range of safety critical software than the current traditional areas. In support of this, the autocratic hierarchical business structure would need to be replaced by a more democratic, flat business structure.

Public acceptance of safety critical software is promoted through debate and transparency. In this way the ethical issues which impact on public acceptance can be analysed and addressed. For example, the Federal Automated Vehicles Policy (2016) recommends, “Manufacturers and other entities, working cooperatively with regulators and other stakeholders (e.g., drivers, passengers and vulnerable road users), should address these situations to ensure that such ethical judgments and decisions are made consciously and intentionally.”

| Idealism | Relativism | Low |
|----------|------------|-----|
| High     | Situationists | Absolutists |
|          | • Rejects moral rules | • Assumes that the best possible outcomes can always be achieved by following moral rules |
|          | • Advocates individualistic analysis of each act in each situation | |
|          | • Relativistic | |
| Low      | Subjectivists | Exceptionists |
|          | • Appraisals based on personal values and perspectives rather than universal moral principles | • Moral absolutes guide judgements but pragmatically open to exceptions to these standards |
|          | • Relativistic | • Utilitarian |

Figure 2: Taxonomy of Ethical Ideologies (source: Forsyth (1980, p 176))

The ethical behaviour of those involved in software development greatly influences the nature of the end product. Moral courage is needed to raise concerns in the face of business pressure. Those involved include business managers, technical managers and software engineers. It is useful to describe systematically the underpinning individual ethical ideology because this helps in the understanding of judgement and action. Forsyth (1980) suggests a taxonomy of ethical ideologies which is summarised in figure 2. There are two dimensions; the first being the degree an individual accepts universal moral rules and the second being the degree of idealism. It appears that there is a tendency for those, including software engineers, involved in the two case studies to lie within the subjectivist quadrant focusing on what is right for the company at the expense of everything else. This is problematic and a challenge to establishing professional integrity within software engineering.

Conclusions

Of the two cases discussed in this paper Volkswagen concerns illegal actions and Tesla concerns legal ones but both concern unethical actions. There are serious issues related to professional practice which need to be addressed. It is hoped such issues are exceptional but sadly it is likely they are commonplace. Unethical actions related to software engineering can be addressed from two sides of application software development. One side focuses on resisting the temptation to perform unethical practice whilst the other side focuses on
reducing the opportunity of performing unethical practice. Recommendations are either reactive where the measure is in response to a particular event or circumstance or proactive where the measure is an attempt to promote desired future behaviour. This approach is illustrated in figure 3 as a matrix containing some examples of typical measures.

| SIDE ONE                                      | SIDE TWO                                 |
|-----------------------------------------------|------------------------------------------|
| Resist the temptation to perform unethical practice | Reduce the opportunity of performing unethical practice |
| **REACTIVE**                                  | **PROACTIVE**                            |
| • Issue fines and other penalties for unethical practice | • Education and Training programmes |
| • Rewards and recognition for good practice  | • Mandatory ethics committees associated with operational actions |
| • Replacing senior decision makers who have line responsibility for those involved in unethical practice | • Regulation and policies frameworks |
|                                              | • Public awareness programmes which generate public pressure |

Figure 3: **Two sided ethical measures matrix**

In the proactive side-two quadrant, Plant (2015) suggests the US government should create a framework, an ombudsman role and safe harbor legislation in order to promote the ethically challenged software industry. In the proactive side-one quadrant new software engineering graduates should have the ethical tools, skills and confidence to challenge decisions made by and instructions from their seniors where such actions from these senior staff are ethically questionable. Such challenge should be capable without detrimental impact on these junior members of staff.

Bowen (2000) suggests a strategy across the complete matrix which is pertinent to base case studies. He states, “it is unethical to develop software for safety-related systems without following the best practice available. All software engineers and managers wishing to produce safety-critical systems in a professional manner should ensure they have the right training and skills for the task. They should be able to speak out without fear of undue repercussions if they feel a system is impossible or dangerous to develop. It is important that companies, universities, standards bodies, professional institutions, governments, and all those with an interest in the well-being of society at large ensure that appropriate mechanisms are in place to help achieve this aim.”

So is professional practice at risk following the Volkswagen and Tesla Motors revelations? The short answer is definitely yes unless software engineers are capable and willing to identify the ethical challenges in software being developed, have the confidence to articulate identified ethical risks, and have the opportunity to influence key decision makers about these ethical risks.

**References**

Bonnefon, J., Shariff, A. & Rahwan, I. (2016) The social dilemma of autonomous vehicles. Science, 24 June Vol. 35, pp 1573-1576.
Bowen, J. (2000) The ethics of safety-critical systems. Communications of the ACM, April Vol.43, No.4, pp91-97.

Bynum, T.W. (2004) Ethical Decision-Making and Case Analysis in Computer Ethics. in Bynum, T.W. & Rogerson S. (editors) Computer Ethics and Professional Responsibility, Chapter3, pp 60-86.

Forsyth, D.R. (1980) A taxonomy of ethical ideologies, Journal of Personality and Social Psychology, Vol 39 No 1 pp175-184.

Goodall, N.J. (2014) Ethical Decision Making During Automated Vehicle Crashes. Transportation Research Record, No. 2424, pp. 58–65.

Gotterbarn, D., Miller, K. & Rogerson, S., Software Engineering Code of Ethics is Approved, Communications of the ACM, October Vol 42 No 10, 1999, pp102-107 and Computer, Oct 1999, pp 84-89.

ISTQB. What is beta testing? Available at http://istqbexamcertification.com/what-is-beta-testing/ Accessed 14 January 2017

Jaynes, N. (2016) Tesla is the only carmaker beta testing 'autopilot' tech, and that's a problem. MashableUK, 9 July. Available at http://mashable.com/2016/07/09/tesla-beta-testing-autopilot-on-public/#dbKporN3uqT Accessed 20 July 2016.

Lambert, F. (2016) A fatal Tesla Autopilot accident prompts an evaluation by NHTSA. electrek, 30 June. Available at https://electrek.co/2016/06/30/tesla-autopilot-fatal-crash-nhtsa-investigation/ Accessed 20 July 2016.

Leggett, T. (2017) VW papers shed light on emissions scandal. BBC News, 12 January. Available at http://www.bbc.co.uk/news/business-38603723 Accessed 17 January 2017

Lin P. (2013) The Ethics of Autonomous Cars. The Atlantic. 8 October. Available at https://www.theatlantic.com/technology/archive/2013/10/the-ethics-of-autonomous-cars/280360/ Accessed 20 July 2016.

Mansouri, N.(2016) A Case Study of Volkswagen Unethical Practice in Diesel Emission Test. International Journal of Science and Engineering Applications Vol 5 Iss 4. pp 211-216.

McBride, N.K. (2016) The ethics of driverless cars ACM SIGCAS Computers and Society (ACM Digital Library); January 2016, Vol. 45 Issue: Number 3 p179-184

McDermid, J. & Kelly, T. (2006) Software in Safety Critical Systems: Achievement and Prediction, Nuclear Future, Volume 02, No.03, pp. 140-146.

Merkel, R. (2015) Where were the whistleblowers in the Volkswagen emissions scandal? The Conversation, 30 September. Available at https://theconversation.com/where-were-the-whistleblowers-in-the-volkswagen-emissions-scandal-48249 Accessed 14 September 2016.

NHTSA (2017) The Automatic Emergency Braking (AEB) or Autopilot systems may not function as designed, increasing the risk of a crash. Final Report, Investigation: PE 16-007

Plant, R. (2015) A Software Engineer Reflects on the VW Scandal. The Wall Street Journal, 15 October. Available at http://blogs.wsj.com/experts/2015/10/16/a-software-engineer-reflects-on-the-vw-scandal/ Accessed 15 January 2017
Queen, E.L. (2015) How could VW be so dumb? Blame the unethical culture endemic in business. The Conversation. 26 September. Available at https://theconversation.com/how-could-vw-be-so-dumb-blame-the-unethical-culture-endemic-in-business-48137 Accessed 15 Sept 2016.

Ragatz, J.A., (2015) What Can We Learn from the Volkswagen Scandal? Faculty Publications. Paper 297. Available at http://digitalcommons.theamericancollege.edu/faculty/297 Accessed 6 September 2016.

Rhodes, C. (2016) Democratic Business Ethics: Volkswagen’s emissions scandal and the disruption of corporate sovereignty. Organization Studies, Vol. 37(10) pp 1501–1518.

Roberts, J. (2016) What is Tesla Autopilot? Tesla's driving assist feature explained. Trusted Reviews 14 July. Available at http://www.trustedreviews.com/opinions/what-is-tesla-autopilot Accessed 20 July 2016.

Rufford N. & Tobin, D. (2016) Who is to blame for dieselgate? Driving tracks down the engineers and asks, could the scandal really destroy VW? Sunday Times Driving 07. Available at https://www.driving.co.uk/news/who-is-to-blame-for-dieselgate-driving-tracks-down-the-engineers-and-asks-could-the-scandal-really-destroy-vw/ Accessed 15 January 2017

SAE International (2014) Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. Standard J3106, issued 16 January.

Simonite, T. (2016) Tesla Tests Self-Driving Functions with Secret Updates to Its Customers’ Cars. MIT Technology Review, 24 May. Available at https://www.technologyreview.com/s/601567/tesla-tests-self-driving-functions-with-secret-updates-to-its-customers-cars/?utm_campaign=add_this&utm_source=email&utm_medium=post Accessed 21 January 2017.

Solon, O. (2016) Should Tesla be ‘beta testing' autopilot if there is a chance someone might die? The Guardian, 6 July. Available at https://www.theguardian.com/technology/2016/jul/06/tesla-autopilot-fatal-crash-public-beta-testing Accessed 14 January 2017

Tesla, (2010), Code of Business Conduct and Ethics. Adopted by the Board of Directors on May 20, 2010.

US Department of Justice (2017) Volkswagen AG Agrees to Plead Guilty and Pay $4.3 Billion in Criminal and Civil Penalties and Six Volkswagen Executives and Employees Are Indicted in Connection with Conspiracy to Cheat U.S. Emissions Tests. Justice News, 11 January. Available at https://www.justice.gov/opa/pr/volkswagen-ag-agrees-plead-guilty-and-pay-43-billion-criminal-and-civil-penalties-six Accessed 15 January 2017

US Department of Transport (2016) Federal Automated Vehicles Policy: Accelerating the Next Revolution In Roadway Safety. September, 12507-102616-v10a, DOT HS 812 329.

Volkswagen (2010), The Volkswagen Group Code of Conduct. Available at http://en.volkswagen.com/content/medialib/vwd4/de/Volkswagen/Nachhaltigkeit/service/download/corporate_governance/Code_of_Conduct/_jcr_content/renditions/rendition.file/the-volkswagen-group-code-of-conduct.pdf Accessed 28 January 2017
