Liability of the manufacturer or software developer under the Product Liability Directive for defective software that cause a self-driving car to cause damage.

LLM in Law & Technology
Tilburg Law School
Tilburg University
January 2020

Author:
Sabbath Mvula
SNR : 2032673
ANR : 313559

Master Thesis supervised by:
DR. Merel Noorman
Second reader : DR. Bo Zhao
Acknowledgements

I would like to express my very great appreciation to my supervisors, Dr. Merel Noorman and Mara Paun for their support, encouragement and guidance throughout this whole process. I would also like to thank Dr Bo Zhao for stepping in as my second reader.

I would also like to thank my father Wilson for persuading me to pursue my LLM in Law and Technology. I wish to acknowledge the support and encouragement provided by my mother Lucy. Finally, I wish to thank my friends Mitchell, Anais and Bradley for their support and encouragement.
Table of Contents

CHAPTER I INTRODUCTION ................................................................. 5
  1.1 Background and problem statement ............................................. 5
  1.2 Literature Review ..................................................................... 6
  1.3 Aim and scope of thesis .............................................................. 9
  1.4 Significance ............................................................................ 9
  1.5 Methodology .......................................................................... 10
  1.6 Structure ................................................................................ 10

CHAPTER II THE TECHNOLOGY OF THE SELF-DRIVING CAR .............. 12
  2.1 Introduction ........................................................................... 12
  2.2 History of self-driving cars ....................................................... 12
  2.2.1. Definition of self-driving cars ........................................... 13
  2.3 Levels of Autonomy ................................................................ 13
  2.4 The Self-driving Car Technology ............................................. 14
  2.4.1 Stakeholders .................................................................... 17
  2.4.2 Risks associated with self-driving cars ............................... 17
  2.5 Conclusion ............................................................................ 18

CHAPTER III PRODUCT LIABILITY LAW ........................................... 19
  3.1 Introduction ........................................................................... 19
  3.2 Background of the Product Liability Directive .......................... 19
  3.3 Product .................................................................................. 20
  3.4 Status of software under the PLD ............................................. 20
  3.5 Services .................................................................................. 23
  3.6 Producer ................................................................................ 24
  3.7 Defect ...................................................................................... 25
  3.7.1 Defences ........................................................................... 28
  3.7.2 Damage .............................................................................. 29
  3.7.3 USA Product Liability Law ................................................... 29
  3.8 Software under the Products Liability Restatement .......... 30
  3.8.1 Product Defects .................................................................. 32
  3.9 Conclusion ............................................................................ 34

CHAPTER IV DUTY OF CARE ................................................................. 36
  4 Introduction ............................................................................. 36
  4.1 General Duty of Care ................................................................ 36
  4.1.2 Duty of care of the manufacturer and software developer .... 36
  4.2 Issues affecting the duty of care .............................................. 40
  4.3 Product Safety ......................................................................... 41
  4.4 Concluding remarks ............................................................... 41

CHAPTER V CONCLUSION .................................................................. 43
  6. Bibliography ........................................................................... 46
    6.1.1 Primary Sources ................................................................. 46
    6.1.2 Legislation- Europe .............................................................. 46
    6.1.2.3 USA Legislation .............................................................. 46
    6.1.3 Case law - Europe .............................................................. 46
    6.1.4 US Case Law .................................................................... 46
    6.2 Secondary Sources ............................................................... 47
6.2.1 Books ................................................................. 47
6.2.2 Articles .................................................................. 47
6.2.3 Documents issued by official bodies ...................... 53
6.2.4 Press .................................................................. 54
1.1 Background and problem statement

We live in a world where new technologies are being invented on a regular basis. There were over ten emerging technologies in 2019 according to a report on innovation.¹ Artificial intelligence and robotics are among the emerging technologies based on the innovation report. Self-driving cars are an example of robotics and it was predicted that in 2020 most of these vehicles which operate independently without the need of a human driver to perform or supervise the driving tasks will be on the road.² The benefits that self-driving cars will bring are many including providing transportation for the elderly and disabled who might not be able to drive themselves.

Self-driving cars depend on software to be able to interpret data from sensors and determine what the car needs to do next.³ The vehicles also receive software updates that improve the functioning of the vehicle. Software can be defective during performing of these functions and this raises liability issue when a self-driving car causes damage as a result of defective software.

A software flaw in a self-driving car caused a self-driving car to cause damage in 2018 when the self-driving car could not detect Elaine Herzberg who was a pedestrian.⁴ The pedestrian suffered physical injuries which led to her death. The investigation of that accident revealed that the self-driving car had software flaws that resulted in it not being able to see the pedestrian that was on the bike.⁵ Defective software can also cause damage to property or computer data and in such a case the responsible party for the damage caused has to be found liable be it the

manufacturer or software developer. Legislative instruments determine on whom responsibility for damage caused by a self-driving car should rest on.\textsuperscript{6} The legislation instruments also determine the conditions on how the injured party can be compensated. In the European Union, Council Directive 85/274/EEC (Product Liability Directive-PLD) is the legislative instrument concerning liability for defective products.\textsuperscript{7} The PLD is applicable if software is considered to be a product under Article 2 of the Directive. Based on Article 2, it is clear that the PLD is applicable to software that is in a tangible form such as CR Rom or programme disks as it qualifies as a movable. A comparative study of the PLD and United States of America product liability laws showed that software that is on a physical carrier is regarded as a product under the PLD and RESTATEMENT (THIRD) OF TORTS: PRODUCT LIABILITY.

There is legal uncertainty as to whether software that is intangible as in the case of downloads, services or updates qualifies to be regarded as a product in both the EU and USA product liability laws. Most self-driving cars use software that is not in a physical carrier to which the current product liability laws in the EU and USA apply to. For example, Tesla self-driving cars receive regular software updates that improve safety and in the case of the software becoming defective the consumer might not be able to strictly hold the manufacturer or software developer liable since the product liability laws are unclear at the moment. Product liability laws need to be clear on whether software is a product or not since most cases of defective software causing damage will be on the rise and consumers need clarity on who the responsible parties should be. It is therefore necessary to examine if the PLD provides solutions in allocating liability to the software developer or manufacturer if defective software is the cause of damage.

\textbf{1.2 Literature Review}

Most of the literature on whether software is a product under the EU or USA product liability laws is based on the text of the Product Liability Directive and the RESTATEMENT (THIRD) OF TORTS:PRODUCT LIABILITY. These two texts define what a product is and the PLD defines it as ‘all movables ,with the exception of primary agricultural products and game even


though incorporated into another movable or into an immovable'.

Under the USA product liability laws the RESTATEMENT (THIRD) OF TORTS defines a product as tangible personal property that is distributed for commercial use or consumption. These two legislative instruments provide that these product liability laws apply to tangible products such as software in a physical carrier and not in an intangible manner as a service, download or upload.

In order to understand whether the PLD applies to software in general the author looked at various European Commission Working Documents and online journals on the subject. The first question to be asked on whether the PLD is applicable to software was answered in 1988 by Lord Cockfield who stated that the Directive was applicable to software same way it applied to handicraft and artistic products. Some scholars are of the view that the PLD is applicable to software that is on material support be it programme disks, CD rom or content information on the software. The reason for the exclusion of software from falling under the ambit of the PLD has been regarded as the fact that software was mainly sold on a tangible medium and not online when the Directive was adopted. The majority view of the scholars also agree that software that is embedded in hardware becomes a whole product and qualifies to be regarded as a product under the PLD.

There is a literature gap on the applicability of the PLD and RESTATEMENT (THIRD) OF TORTS:PRODUCT LIABILITY to intangible software. These two legislative instruments neither include nor exclude software in their definition of what a product is. There are various interpretations by scholars on this issue although most of them are in favour of software being regarded as a product under the product liability laws. The literature is unclear on the legal status of software that is connected to other internet of things because it fails to meet the

---

8 Article 2 Product Liability Directive.
movable aspect because of intangibility. The European Commission has acknowledged that the concept of what a product is under the PLD needs to be evaluated as products and services have become blurred. There is lack of doctrinal opinion on status of software based on its intangible nature.

There is limited literature on whom responsible must lie for damages caused by defective software in the EU. The European Commission states that it is difficult to identify the cause of the defect in the case of a self-driving car as it may be from internet providers, data platform holders or connectivity providers. It is also not clear how the concept of producer should be applied in light of new technologies such as self-driving cars where many actors are involved.

In order to understand what constitutes defectiveness of a product which is a major part of this thesis the text of the PLD in particular Article 6 and the Boston Scientific case are analysed. Firstly, it is unclear on whether a software learning algorithm that causes software to be defective should be regarded as a defect or programming error that causes software to be defective. Article 6 of the PLD only mentions safety that the public is entitled to as the test for assessing defectiveness. The Boston case is the only case law in which the courts gave clarity on assessing defectiveness of a product and on how one defective product can render whole group of products defective. However, the USA courts have more product liability cases compared to the EU for example in Saloomey v. Jeppesen & Co the court ruled that defective navigational charts fell within the scope of product liability. There is no specific case law in both the EU and USA that has dealt with the liability of the manufacturer or software developer for damages caused by defective software.

---

16 ibid.
18 ibid.
Lastly, there is limited literature on the duty of care that the manufacturer or software developer has when software is interconnected with other products or services.\textsuperscript{21} However, the General Product Safety Directive still remains the key legislative instrument on duties of product safety that the manufacturer and software developer must comply with to minimise risk of defective software.\textsuperscript{22}

1.3 Aim and scope of thesis

The aim of this thesis is to examine to what extent the Product Liability Directive adequately covers the liability of the manufacturer or software developer for software defects that causes a self-driving car to cause damage. This will be the main research question:

*To what extent can it be justifiable under the Product Liability Directive to hold the manufacturer or software developer liable for damage caused by a self-driving car as a result of defective software?*

To answer the main research question, the following sub-questions must be addressed:

- *(a)* What are self-driving cars and what software issues may arise from their use?
- *(b)* Is software a product under the Product Liability Directive and if so under what conditions can the manufacturer or software developer be held liable for the defective software?
- *(c)* What is the duty of care expected of the manufacturer or software developer toward a product such as software?

1.4 Significance


The purpose of this thesis is to determine how and if the liability of the manufacturer and software developer for damages caused by defective software is addressed by the product liability laws of the European Union and United States of America. The thesis first examines if the Product Liability Directive covers intangible products such as software within its scope and then compares it to the RESTATEMENT (THIRD) OF TORTS in the USA. Lastly, this thesis will look at the duties that the manufacturer and software developer have towards a product such as software.

1.5 Methodology

This thesis aims to provide a qualitative assessment of the existing legal instruments concerning civil liability for motor vehicles and how these instruments can apply in the context of self-driving cars. A doctrinal legal research has been chosen for this purpose as it focuses on critically analysing the relevant legislation and case law that govern a specific issue. The doctrinal legal research is used when identifying and analysing the content of law. In order to answer the main research question, European Union legislative instruments such as Directives, Commission Decisions and Opinions have been critically analysed. This thesis will also provide a comparative legal research of the European and USA product liability laws to determine how the manufacturer and software developer can be held liable for defective software.

In order to fully understand the technical aspect of self-driving cars a lot of articles and readings on autonomous vehicles and artificial intelligence have been consulted.

1.6 Structure

This thesis consists of five chapters. Chapter 2 describes what self-driving cars are and the software issues that arise from their use. In chapter 3, the product liability framework in Europe and the United States of America will be analysed and compared to determine whether software

---


is regarded as a product under these two regimes. Chapter 4 discusses the duty of care that is expected of the manufacturer and software developer when they place a product such as software on the market. Chapter 5 will be the conclusion of this thesis.
CHAPTER II THE TECHNOLOGY OF THE SELF-DRIVING CAR

2.1 Introduction

This chapter explains the self-driving car technology. The chapter is structured as follows first the history of self-driving cars will be explained which will be followed by describing the levels of automation. The last sections will explain how the technology works and the software risks self-driving cars create.

2.2 History of self-driving cars

The first motor car to be invented was the Benz Motorwagen by Karl Benz in 1885.\textsuperscript{25} This car had three wheels and its engine possessed features that are still common in modern day cars such as the electric ignition and water-cooling system. engine of this car was supplied by gas and it had three wheels. Autonomous technology started operating in motor vehicles in 1925 when a car was invented which was controlled by a radio and no human being was behind the steering or shifting the gears.\textsuperscript{26}

Afterwards the Phantom Auto motor vehicle was introduced in 1926 and it could twist the steering wheel by itself.\textsuperscript{27} Computer-controlled cars were also invented and operated via a television camera input in the car which assisted the car in driving users to their destinations.\textsuperscript{28} Invention of self-driving cars began when the United States Defense Advanced Research Project Agency held Grand Challenges between 2003 and 2009.\textsuperscript{29} The challenges were about creating a fully autonomous vehicle that would complete an off-road course within the prescribed time limit. Competitors of these challenges built self-driving cars equipped with technology that could detect other vehicles on the road and most sensor systems.\textsuperscript{30} One of the teams that participated in the Grand Challenges is the team that started the Google Driverless Car Project in 2009 which is now Waymo. Google started building the self-driving cars and companies like Tesla, Volvo joined the race.

\textsuperscript{27} ibid.
\textsuperscript{28} ibid.
\textsuperscript{30} ibid, 57.
2.2.1. Definition of self-driving cars

The Oxford Legal Dictionary defines a self-driving car as ‘a vehicle that has the technology to drive itself without a person in control’.31 This definition explains how a self-driving car is autonomous as it has the ability to operate without human supervision.32 A self-driving car exercises this autonomy by being aware of its surroundings, intelligently interacting in the driving environment and ability to learn.33 The technology of the self-driving car takes over functions meant for a human driver be it steering or obeying traffic rules.34 Self-driving cars are classified based on the level of autonomy the vehicle has in executing tasks with either little or no human supervision at all.35

2.3 Levels of Autonomy

Autonomy is a crucial feature on how self-driving cars are differentiated. The Society of Automotive Engineers (SAE) developed policy guidelines on the tasks of the vehicle and human driver based on the level of autonomy the vehicle has.36 The guidelines consist of five levels of driving automation which range from 0-5. In each level the vehicle can replace the driver for all or some of the driving tasks and the higher numbers show increased automation of the vehicle in operating without human supervision.37 At Level zero the vehicle has no automation and cannot operate without human supervision. On Level one the responsibility to perform all driving tasks such as acceleration and monitoring the driving environment rests on

33 Ibid, 7.
the driver.\textsuperscript{38} Level 1 is Assisted Driving as the driver still controls the vehicle and keeps an eye on traffic.

At SAE level 2 the vehicle as partial automation and starts to conduct driving tasks such as deceleration or steering but the human driver continues monitoring and performing the driving tasks.\textsuperscript{39} The self-driving car is able to monitor the driving environment alone from levels 3-5 and the level of the car conducting driving tasks independently without human supervision increases under each level.\textsuperscript{40} Level 3 is Conditional Automation and the driver is always on standby to take back control although the vehicle is capable of steering or monitoring the environment. Level 4 is the High Automation level and the human driver does not need to be on standby as the vehicle can conduct the driving tasks and monitor the driving environment by itself.\textsuperscript{41} A self-driving car is fully autonomous at level 5 when it is capable of executing all driving tasks under all road and environment conditions that a human driver could perform.

2.4 The Self-driving Car Technology

\textit{Figure 1} below shows the basic technology of a self-driving car.\textsuperscript{42}

\begin{itemize}
\item \textsuperscript{39} ibid.
\item \textsuperscript{40} SAE International (n36).
\item \textsuperscript{41} Robert L.Rabin and Kenneth S. Abraham ‘Automated Vehicles and Manufacturer Responsibility for Accidents : A New Legal Regime (n38), 4.
\end{itemize}
(a) Lidar and Radar

Lidar and Radar detection systems perceive the physical environment of the car. Lidar is an instrument on top of a self-driving car and rotates emitting laser signals and provides better perception information to the car.\textsuperscript{43} The laser signals collide with objects that might be surrounding the car and the signals measure the distance to detect the surrounding object.\textsuperscript{44} Radar uses radio waves to detect surrounding objects and is better than lidar in measuring the speed of moving objects. During extreme weather conditions radar uses these radio waves to detect the presence of objects.

(b) Video camera

A video camera helps the self-driving car to know the movements of other motorists and pedestrians. The video camera also gathers information on upcoming traffic and surrounding traffic lights to determine if there is a need to stop or change lanes.\textsuperscript{45}

(c) Global Positioning System

The Global Positioning System (GPS) of the self-driving car contains information on details of how each road looks and the signs or lane markings in that street road.\textsuperscript{46} The GPS makes the car aware of its relevant location before it approaches and decides on what driving decisions to take.\textsuperscript{47}

(d) Software

Software that operates in a self-driving car can be programs for a computer or set of instructions on what to do. Self-driving cars use software during the sensing, planning and acting stages.\textsuperscript{48}

\textsuperscript{43} Alex Davies, ‘How Do Self-Driving Cars See? (And How Do They See Me)?’\textsuperscript{44} ibid.\textsuperscript{45} Michael Hicks and Michelle Fitzsimmons, ‘Self-driving cars :your complete guide\textsuperscript{46} Jianfeng Zhao et al, ‘The key technology toward the self-driving car’ (2018) international Journal of Intelligent Unmanned Systems 6, 4 \textsuperscript{47} Harry Surden and Mary-Anne William (n 34) 140.\textsuperscript{48} ibid, 141.
During the sensing stage, sensors such as lidar and radar gather information on location of the car and its surroundings. The software recognises the traffic signs and store them in a database using GPS. The software determines the exact trajectory, lane and speed the vehicle needs to progress.

The information gathered by the sensors, GPS and video camera goes into the computer system to analyse the data gathered and this is the planning stage. The computer system uses algorithms to analyse this data to be able to detect oncoming traffic and scan objects that might be surrounding the vehicle. The vehicle uses software to plan where it needs to go and how to avoid obstacles. During the acting stage the on-board computer moves or stops the vehicle based on the information gathered during the planning stage.

Self-driving cars also use machine learning algorithms to drive. Machine learning algorithms are programs that can self-teach and adapt to a task such as driving by analysing data. The algorithms can observe how a human driver uses brakes or accelerates on a bumpy road and after self-learning it can be able to detect what driving options to make use of on a bumpy road. Machine learning algorithms also help self-driving cars to identify surrounding objects. A self-driving car uses regression and clustering algorithms to predict movements and detect objects by comparing variables so it can predict the position and distance accurately. The vehicle makes decisions through decision making algorithms in which it self-teaches to make appropriate decisions in a certain scenario. These decision making algorithms assist the vehicle know when to use brakes or accelerate. Algorithms also run the operating system of the self-driving car.

(e) Vehicle to vehicle communication

Vehicle-to-vehicle systems (V2V) are normally on self-driving cars to enable them to communicate with each other and other motorists on the road. V2V uses short range radios to

49 ibid.
52 Harry Surden and Mary Anne (n34), 147.
53 ibid,147.
55 ibid.
communicate and send data on location, direction and braking status to other vehicles.\textsuperscript{56} This technology warns drivers and cars on the road about potential accidents and how to avoid them.

2.4.1 Stakeholders

(a) Manufacturer

The manufacturers play a part in the developing of the whole product or component parts. Currently, the manufacturers for self-driving cars and component parts are Tesla, Waymo, Volvo, Mercedes Benz, BMW.\textsuperscript{57}

(b) Software developer

The role of the software developer is to design and put in place a software system and maintain it. There can be more than one software developer in a given scenario for example the software developer who installs the embedded software inside the car might be different from the one who designs the software that is offered as a service.

(c) Internet providers and service providers

Internet providers offer internet coverage for the operation of software and other services. Service providers offer services which vary from software or maintenance of the autonomous system.

2.4.2 Risks associated with self-driving cars

Self-driving cars can have physical or mechanical defects on the operating system of the vehicle which affects public safety. A defect on hardware components such as lidar and radar

\textsuperscript{56} Chaim Gartenberg, ‘Wireless vehicle-to-vehicle communication would be required in new cars under proposed DOT rule’ <https://www.theverge.com/2016/12/13/13936342/wireless-vehicle-to-vehicle-communication-v2v-v2i-dot-nhtsa> accessed 30 June 2019.

results in the self-driving car losing sensor data and not being able to perceive the surrounding objects. This can result in the sudden crash of the vehicle into surrounding objects or human beings affecting public safety.

Software failure is another risk. Software in a self-driving car can fail because of many reasons. It can be because of poor design from the software developer, machine learning processes where the software self-teaches itself or programming choices by the programmer. This can result in the software being defective and not being able to interpret data correctly leading to accidents. This raises liability questions as to whom the responsibility must lie for the damage caused by the defective software.

Software of the self-driving car can be hacked which will lead into third parties gaining access into the operating system of the vehicle for their own agendas. Poor security design of the software can allow cyber-attackers to gain control of the self-driving car and change how it is supposed to function. Hackers may cause software to contain errors which affects the functionality and security of the self-driving car.

2.5 Conclusion

The purpose of this chapter was to explain what self-driving cars and how they operate. Self-driving cars depend on technology such as video cameras, lidar or radar and software to execute driving tasks without human supervision. Software plays a major role as it enables the vehicle to determine location, sense the driving environment and how it must proceed in given a scenario. This chapter also explained the risks associated with self-driving cars such as software failure, hacking and programming failures.

The next chapter discusses how the EU and USA product liability regime addresses the issue of defective software that causes damage.

---

CHAPTER III PRODUCT LIABILITY LAW

3.1 Introduction

The purpose of this chapter is to discuss whether software is regarded as a product under the European Union and USA product liability laws. The chapter starts off by discussing the EU product liability framework and thereafter the USA. A brief background of the PLD will be discussed and that will be followed by analysing other conditions for the PLD to be applicable such as the concept of producer and defect. Concluding remarks will be provided thereafter.

3.2 Background of the Product Liability Directive

In the 1960s and 1970s people incurred losses caused by defective products and there were no substantive laws in place that governed the liability of the producer of the product in such an instance.\(^61\) There were different national legislations on product liability law in the EU with no uniform liability regime in place.\(^62\) As a result of the diverse national laws the PLD text was adopted on 25 July 1980.\(^63\) The Directive itself is binding upon the Member States to which it is directed but it is up to national authorities to choose the methods and form of its implementation.\(^64\) The PLD is the key legal instrument on product liability in Europe with the purpose of protecting consumers against defective products produced by manufacturers of the finished product or producers of the raw material.\(^65\)

Under the Directive, liability does not rest on the premise of fault but on simple fact that a defective product that has caused damage was circulated on the market.\(^66\) As a result the manufacturer or software developer who places a defective product on the market is liable without fault based on Article 1 of the PLD. However, the PLD contains defences in Article 7 that the manufacturer or software developer can make use of to escape strict liability.

\(^{62}\) ibid.
\(^{63}\) Recital 1 Product Liability Directive.
\(^{64}\) Article 288 Treaty on the Functioning of the European Union.
\(^{65}\) Article 3 Product Liability Directive.
\(^{66}\) Article 1 Product Liability Directive.
In order to determine if the liability of the manufacturer or software developer for defective software falls under the PLD, the conditions for the Directive to be applicable will be discussed first.

### 3.3 Product

One of the conditions for the Directive to be applicable is that the damage must have been caused by a defective product.\(^{67}\) The Directive defines a product as “all movables, with the exception of primary agricultural products even though incorporated into another movable or into an immovable”.\(^{68}\) A movable is any object or thing that an individual can move from one location to the other and based on Article 2 immovable goods are excluded from falling within the scope of the Directive.

A movable that is “incorporated into another movable or into an immovable” qualifies to be a product under the Directive.\(^{69}\) An example of this would be the windows that are incorporated into a house, although the house is an immovable, the windows that are movable goods and become incorporated into the house are a product under the PLD.\(^{70}\)

There is an argument that the PLD is applicable to tangible goods even though the Directive does not expressly exclude intangible goods in Article 2.\(^{71}\) This argument arises from the inclusion of electricity in Article 2 as a product which is a non-perceptible good which would fall outside the scope of the Directive.\(^{72}\) Based on this the assumption is that the Directive is not applicable to intangible goods.\(^{73}\)

### 3.4 Status of software under the PLD

Since a product is a prerequisite for the application of the PLD it has to be answered whether software qualifies as such. Article 2 of the PLD defines products as movables, primary

\(^{67}\) Article 1 Product Liability Directive.

\(^{68}\) Article 2 Product Liability Directive.

\(^{69}\) Article 2 Product Liability Directive.


\(^{71}\) Ibid.

\(^{72}\) Ibid.

\(^{73}\) Ibid.
agricultural products and electricity and software is neither included nor excluded. When the PLD was adopted software that is not on physical form did not play a major role as it now does in self-driving cars. This might be the reason for the exclusion but nowadays software in self-driving cars can become defective and cause damages which is why it is important that it be included as a product in Article 2.

Another reason for the exclusion of status from the ambit of product in Article 2 is the fact that it is intangible.\textsuperscript{74} The conclusion is based on the inclusion of electricity as a product and the reasons mentioned when the definition of product was discussed. However, there is no explicit exclusion of intangible goods as products under the PLD and as a result there are various interpretations. The solution to this problem might lie in the text of the Directive being amended to either include or exclude software and this would have binding authority. However, the inclusion of software as a product will benefit consumers who suffer from damages caused by defective software in a self-driving car as they would be able to strictly hold liable the manufacturer or software developer. Most authors argue that software that is in a tangible form such as a programme disk or CD Rom falls within the definition of a product in Article 2 as it is a thing that can be physically perceived.\textsuperscript{75}

Software is often bundled with other hardware services and it is generally accepted that the bundle will be a product within the meaning of the product.\textsuperscript{76} A self-driving car can use software that is offered by a service provider to know location or perceive surrounding objects. The software that the self-driving car uses might be on a cloud and this is another case of software to which the PLD should be applicable.

Most scholars also agree that software should be regarded as a product under the PLD. Walter Van Holst is of the view that software should be regarded as a product under the PLD as a means of ensuring that manufacturers are held responsible for fixing the software bugs and errors since software can be insecure.\textsuperscript{77} Taivo Liivak also holds the same view and argues that upgraded or downloaded software should all fall under the product definition in Article 2 since

\textsuperscript{75} Ibid.
the origin of software defects is irrelevant as they all form part of the vehicle. I tend to agree with both of them bearing in mind the purpose of the PLD which is to protect consumers from defective products that are circulated on the market.

The European Commission has evaluated the applicability of the Directive to digital machines or computing devices. The Commission had to evaluate whether the PLD in its current state still protected consumers in light of developments such as the internet of things and technological developments such as autonomous vehicles and it was agreed that the Directive had to be revised to cater for new technological developments. Among the issues of discussion was whether the internet of things products fell under the definition provided in Article 2 since in these cases software is sometimes supplied as a separate product through downloads, updates or as a service.

A lot has changed in terms of technological developments from 1985 when the Directive was adopted products were mostly independent and not bundled together with other products as the case with software which has become a complex product. The kind of products that the legislators were acquainted with in 1985 is different from products of today which is why there has been majority consensus on the evaluation of the Directive in light of these technological advancements. The Directive does not need to be completely changed it only needs to include complex products such as software under Article 2 as products.

Although the main argument has been in favour of software being included as a product under the PLD so that manufacturers and software developers can be held responsible without fault on their part for defective software there are disadvantages to it that cannot be ignored. Strict liability encourages manufacturers or software developers to perform necessary tests in ensuring that most of the software placed on the market will not be defective for fear of being

78 ibid.
80 ibid, 7.
81 ibid.
held liable for defects.\textsuperscript{83} However, holding them liable for every defective software might result in them becoming reluctant to produce or develop the software for fear of compensating victims in cases of damages caused by the defective software. For example, if a software developer is strictly held liable for a certain software the chances of other developers using that software are less because of fear of it being regarded defective again.\textsuperscript{84} This limits the development of the software industry as most players will be unwilling to perfect the software that was once regarded defective and the issue of paying damages for defective software chases potential software developers on the market.

\subsection*{3.5 Services}

Software that is also in an intangible form is when it is supplied as a service. When it is supplied as a service the third-party provider hosts applications and consumers access them over the internet.\textsuperscript{85} The question of whether defective equipment that was supplied as a service could fall under the ambit of PLD was answered in the \textit{Centre hospitalier universitaire de Besancon v Thomas Dutruex}.\textsuperscript{86} In this case, Thomas Dutruex aged 13 years was admitted to a hospital and suffered burns during a surgery because of a heated mattress that he was placed on which had defective temperatures.\textsuperscript{87} It had to be decided if the hospital as the service provider of health services to Thomas Dutruex could be held liable within the scope of the PLD for the damage suffered as a result of the defective mattress without them being at fault.\textsuperscript{88} The judgement explained that ensuring consumer protection required that all producers involved in the production process be made liable for any component part that was supplied by them and was defective.\textsuperscript{89} CJEU stated that the principle of the producer being held liable for damages caused by a defect in their product and conditions under which a person who presents himself as producer must be read in light of the first and fourth recitals of the Directive.\textsuperscript{90} It was held

\begin{thebibliography}{99}
\bibitem{86} Case C-495/10 \textit{Centre hospitalier universitaire de Besancon v Thomas Dutruex and Causse Primaire d’assurance Mutuelle du Jura} [2001] ECR I-03569.
\bibitem{87} ibid, para 10.
\bibitem{88} ibid, para 18.
\bibitem{89} ibid, para 23.
\bibitem{90} ibid, para25.
\end{thebibliography}
then that during the adoption of PLD in weighing the parts played by different economic operators in the production chain a choice was made to allocate liability for damage caused by defective products to producers and in certain cases suppliers and importers.\textsuperscript{91} The ruling was that liability of the hospital could not fall under within the PLD’s scope since the mattress had been acquired elsewhere and they did not take part in the production and marketing process.\textsuperscript{92} However the liability of the service provider could fall under the PLD if the producer cannot be identified and if the supplier formed part of the distribution chain.\textsuperscript{93} Based on the above ruling, the PLD is applicable in holding the service provider of the defective software that causes a self-driving car to cause damage if they were involved in producing the software and circulating it on the market. If the software developer supplied the software as a service and is the producer who circulated it on the market their liability falls within scope of Directive.

It is to the advantage and protection of consumers that the Directive becomes applicable without exclusions based on whether the software is downloaded, updated or offered as a service. This is based on the fact that in 18 Member States consumers do not have a specific liability framework that allows them to be compensated for damages caused by defective services or software.\textsuperscript{94} It is therefore necessary that the PLD include software as a product so that consumers obtain compensation for damages caused by defective software.

**3.6 Producer**

The other grey area in the applicability of the PLD is the concept of ‘producer’. The producer is defined as the manufacturer of a finished product or a component part which makes the PLD applicable to manufacturers whose products become defective.\textsuperscript{95} Firstly, the manufacturer of a finished product is held liable in Article 3 (1), for example the manufacturer that produced the final version of the software is the one referred to in this section. Producers of raw materials are also held liable in the same vein as manufacturers of a component part of a finished product. The manufacturer who was responsible for the component part of the finished software product is also strictly held liable under the Directive. Any person who presents themselves as a

\textsuperscript{91} ibid.
\textsuperscript{92} ibid, para 26.
\textsuperscript{93} ibid.
\textsuperscript{95} Article 3 (1) Product Liability Directive.
producer by using their name on the software product or uses a trademark or other distinguishing features will be regarded as a producer.96 In the case of imported goods, the importer of the software is liable as the producer.97 Lastly, if the producer of the software cannot be identified, the supplier will be held liable.98

Software being included as a product under the PLD potentially makes the software developer be held liable as a producer. The software developer could fall under the concept of a manufacturer who is the producer of the raw material or component part since they build the software or develop it even though it may be, intangible they are still involved in the production process. In addition to that the Directive aims to protect the consumer from all producers involved in the production process if the product is defective.99

The various actors mentioned above in the case of defective software make it difficult to be able to identify who the producer is for the specific service or product that became faulty. Manufacturers are normally held liable for defects for the finished product and in light of the various liable actors there is a possibility that they may be held fully responsible for every defect. Consumers also target manufacturers after a malfunction causes an accident since they are believed to have the deepest pockets of all involved parties.100 The targeting of the manufacturer is unjustified since there are other actors who might have contributed to the software failing for example if there was no network to download or update the software.

3.7 Defect

Defectiveness of a product is another condition for the PLD to be applicable. Once the product is regarded as defective within the meaning of the PLD then the producer is held liable without fault for damages caused. Article 6 of the Directive provides that a product is defective when it does not provide the safety which a person is entitled to taking all circumstances into account.101 The Directive also mentions three circumstances that must be taken into account:

96 Article 3 (1) Product Liability Directive.
97 ibid.
98 Article 3 (2 ) Product Liability Directive.
which are presentation of the product, the use to which it could reasonably be expected that the product would be put and the time when the product was circulated.\textsuperscript{102} Moreover a product is not considered to be defective if a better product is subsequently put into circulation.\textsuperscript{103} 

Based on Article 6 a product is defective if it does not provide safety. In the case of software, it can only be regarded as defective within the meaning of the Directive if it does not provide safety and not on the fact that it has errors or bugs which do not compromise on safety. In the case of the software system being hacked it compromises on safety which makes it to be potentially a safety issue. A hacker tampering with system of the car affects safety, but it is an open-ended question whether it becomes a safety issue through access of the system or when the hacker has meddled with the operating system.\textsuperscript{104} If the software has vulnerabilities that make it easy for the hacker to gain access it can be considered to be defective as that software is not safe.

In terms of Article 6 the product must have the safety that a normal person is entitled to expect. It has been argued that this is based on the legitimate expectations of the public and not subjective expectations a normal person has in the \textit{Boston Scientific Medizintechnik} case.\textsuperscript{105} In this case the question was whether cardioverter defibrillators were defective if they belonged to a group of products whose battery depleted quickly during medical procedures causing failure.\textsuperscript{106} The court found that if products belonging to same group or forming part of same production have a potential defect it is possible to classify all of them as defective without showing the defectiveness of the product in question.\textsuperscript{107} The reasoning was based on the fact that there was an increased risk on the safety that the public at large was entitled to during health procedures and it was not necessary to establish that the safety standard was not met.\textsuperscript{108} As a result it is the legitimate expectations of the public at large that must be assessed in terms of safety not subjective expectations of an individual. However, in the case of software it might be difficult to know the overall legitimate expectations of the public as most people are not well acquainted with how software operates to know the safety that software must comply with.

\begin{itemize}
\item \textsuperscript{102} Article 6 (1) (a-c) Product Liability Directive.
\item \textsuperscript{103} Article 6 (2) Product Liability Directive.
\item \textsuperscript{104} Maurice Schellekens, ‘Car Hacking: Navigating the regulatory landscape’ (2016) Computer Law & Security Review 32,313.
\item \textsuperscript{105} Joined Cases C-503/13 and C-504/13, \textit{Boston Scientific Medizintechnik GmbH v AOK Sachsen-Anhalt and others} [2015].
\item \textsuperscript{106} ibid, paras 13.
\item \textsuperscript{107} ibid, para 41.
\item \textsuperscript{108} ibid ,para 40.
\end{itemize}
The *Boston* case is relevant in a case where software is mass-produced and some of it is found to be defective. Based on the ruling in *Boston*, if one cardioverter defibrillator is found to be defective then all products forming part of same group are automatically regarded as defective.\(^{109}\) It is argued that the court delivered a consumer-friendly judgement as plaintiff do not need to show that the product in question was defective if it belongs to a group of products that have been found to be defective.\(^{110}\) In future cases where software is mass-produced and some of it is found to be defective, a plaintiff who suffers harm from software belonging to that group will not need to show that it was defective. The same could be applicable where software became defective because of machine learning processes where it self-teaches itself and not because of a defect during mass production.\(^{111}\) Software might become defective because of a programming error and it can constitute as a defect under Article 6 if the error results in the safety of the public being affected. Lastly, it is beneficial for the consumer who seeks compensation for damages caused by defective software for the *Boston* ruling to be applied to cases of software since the PLD does not currently cover this issue.

The first circumstance to be taken into account when assessing if a product is defective is how the product is presented as stated in Article 6 of the Directive. How a product is presented on the market influences the safety expectations of the public. Presentation of the product includes how it will be offered to the public and in this sense how it is marketed, advertised or packaged is important.\(^{112}\) If the product has inaccurate or missing information it may render it as defective.\(^{113}\) However with a product such as machine learning software operating in an autonomous vehicle it might not be possible to provide all the information on the risks that might happen because the software self-teaches itself sometimes.

Article 6 of the Directive states the second circumstance as ‘the use to which it could reasonably be expected that the product would be put’.\(^{114}\) The producer has to expect the reasonable expected use of the product by the consumer and this is why the warnings and instructions on the safe use of the product are important to avoid unreasonable use. If the

---

\(^{109}\) Ibid.


\(^{112}\) D Wuyts, ‘The product liability directive -more than two decades of defective products in Europe (2014) JETL, 16.


\(^{114}\) Article 6(1) (b) Product Liability Directive.
consumer uses the product in a way that is different from the expected use of the product the producer will not be strictly held liable for it.

The last circumstance is the time when the product is put into circulation. The legitimate expectations of the public are assessed during this time and not after the product has been circulated already. Any other technical, safety and mandatory standards that occur later after product is circulated already are not taken into account.

3.7.1 Defences

The manufacturer or software developer can rely on the defences provided in the Directive if they are regarded as a producer within the meaning of the PLD. There is a reversal of burden of proof from the consumer who proves damage, defect and causation to the producer proving that the defect arose due to them complying with any of the circumstances stated in Article 7 of the PLD. Firstly, the producer is not held liable if they prove that they did not circulate the defective product on the market. A manufacturer or software developer can be exempted from liability based on this first circumstance if they prove that they did not circulate the defective software on the market. The producer is also not held liable under the PLD if they prove that the defect in the product was not present when they circulated the product on the market. Once again, the manufacturer or software developer will not be held liable upon evidence that the product was not defective before it went on the market.

The producer can also be exempted from liability if they did not sell, manufacture or distribute the defective product. Software that becomes defective after the producer did not sell it or manufacturer does not fall under the liability scope of that producer. The producer can also prove that the defect arose because they were adhering to mandatory regulations from public authorities to escape liability. Lastly, the producer of a defective product if science and technical knowledge were not advanced when the product was circulated so that the defect

115 Article 6 (1) (c) Product Liability Directive.
117 ibid.
118 Article 7 Product Liability Directive.
119 Article 7 (a) Product Liability Directive.
120 Article 7 (b) Product Liability Directive.
121 Article 7 (c) Product Liability Directive.
122 Article 7 (d) Product Liability Directive.
could be noticed. Lastly, the manufacturer of a component part is exempted from liability if the product became defective as a result of the instructions given by the manufacturer of the whole product. The manufacturer of the component part is also exempt from liability if the defect is attributable to the design of the product in which the component has been fitted according to the instructions given by the manufacturer of the product”.

The producer who provides evidence for complying with any of these defences is exempted from liability under the PLD.

### 3.7.2 Damage

It is important to discuss whether the PLD allocates for damages that arise from defective software. Article 9 of the PLD only covers for ‘damages caused by death or personal injuries or damage to, or destruction of, any item other than the defective product itself’.

Damages that can arise as a result of defective software in a self-driving car are personal injuries or death of people through accidents, damage to physical properties such as buildings and lastly damage to the self-driving car itself.

If damage stems from the hardware components of a self-driving car such as the sensors and video cameras the PLD will be applicable as these are movables. However, in the case of software that is not in material form and supplied as a service, the issue of damages becomes problematic since the defect that can occur may be it to the car itself or other cars is not covered by the Directive. Article 9 of the PLD only covers for ‘damages caused by death or personal injuries or damage to, or destruction of, any item other than the defective product itself’. If software is not regarded as a component part of the self-driving car, then a defect in the hardware components of the car itself could fall under the damages awarded based on Article 9 since the defective software will not be part of the product.

### 3.7.3 USA Product Liability Law

---

123 Article 7 (e)Product Liability Directive.
124 Ibid.
125 Article 9 Product Liability Directive.
126 Article 9 Product Liability Directive.
In USA the Restatement (Third) of Torts: Products Liability is the liability regime for damages caused by defective products.\textsuperscript{127} Most states have adopted the Products Liability Restatement into their own product liability laws but there are differences in application in various states. The liability regime is a strict one and the manufacturer or seller of the defective product is held liable without fault on their part.\textsuperscript{128} The Products Liability Restatement holds any person engaged in selling or distributing the defective product strictly liable for the defect.\textsuperscript{129} The injured party bears the burden of proving the damage caused by the defective product by the manufacturer, seller, producer or distributor.

3.8 Software under the Products Liability Restatement

A product is defined in §19 as tangible personal property that is distributed for commercial use or consumption.\textsuperscript{130} Examples of tangible property that the courts have ruled that they fall within the scope of the Product Liability Restatement are houses, water and cars. The courts have also applied product liability laws to intangible products such as gas and navigational charts.

Software is not included in the Products Liability Restatement and just like in the EU there are arguments on whether it should be regarded as a product or not. It is argued that software was in a primitive stage and not mass produced during the adoption of the Products Liability Restatement hence its exclusion.\textsuperscript{131} Nowadays most technologies are heavily reliant on software and this will result in more software failures causing a rise in damages caused by defective software. It is therefore beneficial for the consumer both in the EU and USA for software to be regarded as a product so that they can strictly hold liable manufacturers and software developers of defective software.

Since product liability laws differ in the USA, courts have ruled on whether the Products Liability Restatement is applicable to intangibles such as software. A judgement that is relevant

\textsuperscript{127} RESTATEMENT (THIRD) OF TORTS: PRODUCT LIABILITY (American Law Institute 1998) hereafter Products Liability Restatement
\textsuperscript{128} RESTATEMENT (THIRD) OF TORTS: PRODUCT LIABILITY § 1 (American Law Institute 1998)
\textsuperscript{129} Ibid.
\textsuperscript{130} RESTATEMENT (THIRD) OF TORTS : PRODUCT LIABILITY § 19 (a) (American Law Institute 1998).
on this topic is the case of *Aetna Casualty and Surety Co. v. Jeppesen & Co.* In this case Jeppesen designed navigational charts which had inaccurate information without altering them which resulted the plane relying on the defective chart. The court ruled that Jeppesen had a responsibility to ensure that the information on the navigational charts did not contain errors and regarded the information on the charts as a product for strict liability purposes. Based on this ruling software can be regarded as a product under strict liability laws as it can contain errors and provide wrong information just like the navigational charts which were found to be defective.

In *Retail Systems, Inc. v. CNA Insurance Cos.*, a computer consultant instituted a claim against an insurer for loss of their computer tape and data. The court ruled that the computer data lost constituted tangible personal property since it was of permanent value since it was integrated with physical computer data and therefore the case fell the *Product Liability Restatement*. These two cases show that the courts are willing to include information and computer data as product. Software has applications that contain data which can provide wrong information due to errors and if computer data could be regarded as a product the same could be applicable for software.

The *Product Liability Restatement* in § 19 mentions the conditions which must guide the courts when determining whether any product falls under the strict liability framework in the USA. These are consideration of public interest in life and health, whether product is the stream of commerce and checking if there is justice when imposing loss on the manufacturer. Software plays a major role in business and it is in the public interest of consumers for software to be considered as a product so that they get compensated under the product liability laws. Michael Scott also argues that the conditions mentioned in § 19 highlight that software that provides security for computer systems should be a product under the product liability regime.

---

133 *ibid*, 672-673.
134 *ibid*, 676-677.
136 *ibid*, 736-737.
137 *ibid*.
3.8.1 Product Defects

In USA product defects are classified as manufacturing, design or failure to give warning or instruction defects.\textsuperscript{139} These categories are normally used in practice in the European Union. The first defect is the manufacturing defects and it is when the product departs from the intended design by the manufacturer. \textsuperscript{140}Manufacturing defects are assessed according to the consumer expectations test since the product would have been damaged or flawed contrary to what the consumer expects from the use of that product. The manufacturer has a duty to exercise reasonable care when making products so that they are not unsafe but comply with what a consumer expects a safe product to be. The consumer has to prove that the product did not operate as the manufacturer warranted regarding its use and how the malfunction proves that the defect occurred during the manufacturing process.\textsuperscript{141} In \textit{Roland Todd White v Mazda Motor of America INC} a plaintiff instituted a product liability claim after his Mazda caught fire on the highway because of a defect in the vehicle’s fuel system.\textsuperscript{142} The plaintiff did not succeed against the defendant because the Connecticut Supreme Court held that the plaintiff did not properly prove the defect, how his car could not have caught fire without the defect and how the defect was attributable to the manufacturer. The plaintiff did not also have enough evidence in \textit{Johnson v Black &Decker Inc} after a router failed to turn off injuring them in the process.\textsuperscript{143} The court reasoned that evidence needed to support this claim that the route did not conform as specified by the manufacturer. Based on these two rulings, the courts require the consumer to provide evidence in case of manufacturing defects which might be problematic in the case of defective software since the consumer might not be able to prove the malfunction as they do not know how it operates. In addition to that the issue of providing evidence is costly as expert witnesses might be needed to prove how the malfunction was a manufacturing defect.

The second defect is the failure to warn or providing inadequate instructions. This applies when “the foreseeable risks of harm posed by the product could have been reduced or avoided by the provision of reasonable instructions or warnings”.\textsuperscript{144} Instructions must be understandable

\textsuperscript{139} RESTATEMENT (THIRD) OF TORTS : PRODUCT LIABILITY§2 ( American Law Institute 1998).
\textsuperscript{140} RESTATEMENT (THIRD) OF TORTS : PRODUCT LIABILITY §2 (a) (American Law Institute 1998).
\textsuperscript{142} Roland Todd White v Mazda Motor of America INC 99 A.3d 1079, 1090 (Conn .2014).
\textsuperscript{144} RESTATEMENT (THIRD) OF TORTS : PRODUCT LIABILITY §2 (c) (American Law Institute 1998).
to an average person in order to be regarded as adequate. In Nowak v. Faberge, the consumer purchased a hairspray that was flammable. Although the bottle had a warning label on how it was unsafe to use it near open fire, the plaintiff on the day failed to use the spray because the valve was not working. The plaintiff used a can opener to open the hairspray but since they were near a gas stove fire broke out burning them in the process. The court ruled that the producers were strictly liable for the damage suffered by the plaintiff because there were no sufficient warnings in place to avoid this damage. In the context of software, an instruction defect would be the failure of the producer or software developer to give adequate warning to the user or provide warnings on what would happen if incorrect software is downloaded or if software does not update. The manufacturer can be strictly held liable if they fail to provide sufficient warnings on the risks associated with incorrect download or failed download of software be it security breaches or failure of operating system.

The third defects are design defects. The manufacturer is strictly liable for design defects if “the foreseeable risks of harm posed by the product could have been reduced or avoided by adopting a reasonable alternative design”. Contrary to the consumer expectations test that is applied for a manufacturing defect, the risk utility approach is used to prove if there was a design defect but different states employ one of the two tests. In terms of the risk utility approach the plaintiff has to prove that an alternative design could have reduced the risk caused by the product through preventative measures. The risk utility approach was applied in Potter v. Chicago Pneumatic Tool Co when the plaintiff suffered injuries and the court’s reasoning was that the plaintiff had presented evidence of alternative designs that could have minimized the injury. Contrary to the Potter ruling, the consumer expectations test was applied in Brown v. The Raymond Corp. In this case a forklift crushed the plaintiff’s foot and the court rejected the claim on the basis that evidence provided did not establish that the product was dangerous beyond what an ordinary consumer expected. Defective software will fall under design defects if it does something not anticipated outside of what it was programmed to do. In such a case it has to be proved if there was an alternative design the manufacturer or software developer could have used to prevent the software from doing the unexpected or having errors.

145 Nowak v. Faberge USA 32 F.3d 755 (3d Cir.1994).
146 RESTATEMENT (THIRD) OF TORTS : PRODUCT LIABILITY §2 (b) ( American Law Institute 1998).
149 Brown v. The Raymond Corp 432 F.3d 640 (6th Cir.2005).
150 ibid, 644.
3.9 Conclusion

The Product Liability Directive is the key legislation in the European Union that establishes the strict liability of producers for defective products placed on the market. The Directive was adopted to protect consumers that seek compensation from producers of defective products. Only movable products fall under the applicability of the Directive and this is a problematic issue in the case of software that is not in a physical carrier. Software that is intangible does not meet the movable criteria of a product and the PLD does not include or exclude software as a product under Article 2.

It would be beneficial to the consumer for software to be included as a product so that they can get compensation from producers and software developers of defective software. Secondly, the inclusion of software as a product makes the liability of the software developer and manufacturer fall within the scope of the PLD for the defective software. Software operates in most technologies nowadays and it has the potential to fail which will result in an increase of defective software cases and therefore it is vital that the Directive includes software in its definition of a product. Holding manufacturers and software developers strictly liable for defective software reduces the risks of defective software as producers of the software will fear paying for damages in case the software gets defective.

The Court of Justice has not provided clarity on the status of software under the PLD but has given the criteria on assessing defectiveness of a product in *Boston* which is the legitimate expectations of the public in terms of safety. The defectiveness criteria in *Boston* is applicable in the case where defective software caused damage.

In USA, the *Products Liability Restatement* which is the legal framework for product liability does not also include software in the definition of a product. USA courts have included defective navigational charts and computer data as falling under product liability laws. If the courts can include these intangible products as falling under the *Products Liability Restatement* so should software. The USA courts adopt the consumer expectations test when assessing manufacturing defects which was the approach taken by the Court of Justice in *Boston*. The consumer expectations test is in line with purposes of product liability laws which is protecting
the consumer. However, the utility risk approach is a better approach for badly designed software as a new version of software can be proof that the older version had defects.
CHAPTER IV DUTY OF CARE

4 Introduction

Self-driving cars have the potential to cause damage because of defects in the operating system. In ensuring safety, manufacturers and software developers have a duty of care in promoting safety, providing warnings and instructions and protecting the interests of the consumer. It is therefore necessary to analyse the kind of duties of care expected from these actors. This chapter will first discuss what the duty of care is and then analyse the duty of care that is expected of the manufacturer or software. The chapter will also look at the issues that affect the duty of care such as foreseeability and lastly examine the legal obligations of product safety for producers set out in the General Product Safety Directive.

4.1 General Duty of Care

A duty of care is the legal responsibility of an individual to avoid any behaviour or omissions that could reasonably be foreseen to cause harm to others. It is a moral and recognised duty to care for other people.151 This is a concept from Dutch law where every individual is required to act with carefulness or diligence in protecting others against harm even though the harm might emanate from elsewhere.152 The duty of care depends on the particular circumstances and includes warning potential victims of harm and taking measures to prevent the occurrence of harm.153 Three elements must be met in establishing the duty of care that one person can owe to others and these are: the defendant must owe the plaintiff a duty of care, there must be a breach of that duty and the plaintiff must suffer damages as a result of the breach.154

4.1.2 Duty of care of the manufacturer and software developer

Self-driving cars operating on level 3-5 have increased automation as the car performs all safety functions under certain traffic or environmental conditions.155 It does so through being connected to the internet for services such as software updates or downloads maybe necessary

---

152 Ibid, 17 see also art.1402, 6:162 of the Dutch Civil Code of 1838.
154 Van Dam Cees, ‘European Tort Law’ (Oxford University Press 2009), 90.
155 James M. Anderson (n 29), 3.
for parking which may be complex for ordinary people to understand. The user of the self-driving therefore needs to be equipped with necessary knowledge of how the car functions for better understanding and use of the vehicle. Based on this, manufacturers and software developers have a duty to provide adequate warnings, instructions and proper information on the proper use of the self-driving car and software features. Software developers are in the best position to foresee insecure software causing bugs and have expert knowledge on fixing them. It makes sense for this duty of care to rest on them.

In the context of insecure software, a software developer has the duty to design and develop secure software and provide adequate instructions to the licensee on the safe use and dangers of the software.\footnote{Michael D Scott, ‘Tort Liability for Vendors of Insecure Software :Has the Time Finally Come’ (2008) Md.L.Rev , 443.} This duty is necessary as insecure software has the potential to be defective and not functioning as intended for the specific purpose. Instructions and safe designs mitigate the risks of defective software causing harm. The standard of care expected of the software developer is only reasonableness because software vendors argue that software is subject to defects and bugs.\footnote{ibid,443.} Michael D Scott is of the opinion that the important element in establishing the standard of care for secure software is to determine the custom or usage in the software industry and the security standards that relate to software.\footnote{ibid, 446.} As a result, most software developers can be found to be negligent if they could prevent the harm if it was foreseeable in the software.\footnote{Michael L. Rustad & Thomas H.Koenig, 'The Tort Law of Negligent Enablement of Cybercrime (2005) Berkeley Technology Law Journal 20,1575.} Industry standards of what constitutes good software development play a huge role in the assessment of whether a reasonable duty of care was met by the software developer in minimising risk of insecure software.

Software upgrades or downloads will play a huge role in autonomous vehicles and this will create software reliability challenges.\footnote{ibid, 70.} Self-driving cars will be connected to the internet for these upgrades or downloads which makes the vehicle more vulnerable to cyberattacks.\footnote{ibid.} A virus can attack the system and cause the software to be defective. Software developers have a duty to ensure that the source of the software upgrades is legitimate and uncorrupted.\footnote{ibid, 70.} They
also need to ensure that users or criminals cannot hack into the vehicle’s hardware and software systems by performing inspections regularly.\textsuperscript{163}

If the software was purchased on contract the seller has contractual duties that emanate from the sale of that software. It is the duty of the seller of the software to ensure that the software has characteristics that the buyer expects on the basis of the contract.\textsuperscript{164} A buyer of the software would not obviously expect vulnerabilities in the software that hinder use of the product. In the event of software vulnerabilities, the software supplier or developer is under an obligation to repair them without delay, but this depends on the terms and conditions of the contract of sale.\textsuperscript{165} The duty to fix bugs of the software is normally under the license agreement that the user accepts.\textsuperscript{166}

The risks associated with the operation of software such as hacking, internet failures which may result in failure of software updating or downloading may cause harm to consumers. In this regard manufacturers and software developers have a duty to implement monitoring technologies that notify the user or stakeholders when something is about to or has gone wrong.\textsuperscript{167} This includes implementation of disabling unexpected conduct of the robot and informing victims on actions to take to avoid harm.\textsuperscript{168} However, not every action by an autonomous car can be predicted because of machine learning software that are self-teaching making the duty to monitor extremely difficult. In addition to that a monitoring duty raises privacy concerns as a lot of data is stored inside the self-driving car.\textsuperscript{169}

The duty of care that the manufacturers owe to passengers and consumers depend on the level of automation the vehicle operates on.\textsuperscript{170} The duty expected on autonomous vehicles operating

\textsuperscript{163} ibid.
\textsuperscript{164} Tjong Tjin Tai et al Duties of care and negligence against cybercrime (n152), 55.
\textsuperscript{165} ibid, 57.
\textsuperscript{168} ibid,33.
\textsuperscript{169} ibid, 34.
on level 0-2 is reasonable care since the human driver can take over some aspects of the driving tasks. A heightened standard of reasonable care is expected on autonomous vehicles operating on levels 3-5 as the car completes most of the driving tasks without human supervision and this requires designing a car that is safe as users have to trust its actions.

Manufacturers owe consumers a duty to take reasonable care when they place products on the market. In Donoghue v Stenvenson, Ms Donoghue the plaintiff was offered a drink bought by a friend and did not check the contents of the bottle since it was invisible. A snail had decomposed in that bottle and she fell sick and got diagnosed with gastroenteritis and took legal action against Mr David Stenvenson the manufacturer of the ginger beer. In delivering the judgement, Lord Atkins stated that reasonable care ought to be taken to avoid acts or omissions which a person can reasonable foresee that they will harm others affirmed that negligence is a tort and a plaintiff can institute civil action if the defendant’s negligence results in them suffering injuries or loss of property.\textsuperscript{171} The court held that manufacturers owe consumers a duty to take reasonable care when they place products on the market. Secondly, the ‘neighbour principle’ was introduced and it simply means that reasonable care must be taken to avoid commissions or omissions that can reasonably be foreseen to likely injure other people that are related to the act in question.\textsuperscript{172} Based on this ruling it was concluded that Ms Donoghue was entitled to damages. The manufacturer exercises reasonable care when they check that the product placed on the market does not pose risks to the health and safety of the public and a failure to check constitutes negligence on their part.

There is also the duty to choose the right system and tasks for the technology.\textsuperscript{173} This duty involves manufacturers and software developers ensuring that users or operators of self-driving cars have the necessary skills and abilities to operate them. Included in this duty is monitoring the technology and providing safety checks regularly. Autonomous systems complicate this duty as they can behave in an unpredicted manner which makes it difficult for manufacturers or operators to know the right skills and duties to discharge in a certain case.

\textsuperscript{171} ibid.
\textsuperscript{172} ibid.
It is argued that there must also be a duty on producers to equip technology with recording information on how the technology operates.\textsuperscript{174} In the case of self-driving cars, the recording technology will reconstruct events that caused the accidents making it easier in allocating responsibility.\textsuperscript{175} Since there is recording of lots of personal data this duty must be exercised in accordance with data protection rules.

### 4.2 Issues affecting the duty of care

The first element that affects the duty of care is the fact that some actions and type of harm that an autonomous system can be unforeseeable.\textsuperscript{176} Autonomous systems make independent decisions that may not be anticipated despite the manufacturer taking reasonable care in maintaining or updating the autonomous system. The duty of care is difficult to assess in the case of unforeseeable actions by autonomous systems as they are self-learning and can behave in a different manner from what they were programmed to do. In the context of software, some software errors are not generally foreseeable despite efforts by software developers to fix them bugs somehow occur.\textsuperscript{177} Despite this software developers are urged to reasonably foresee software flaws that may arise since they are in a better position to rectify the flaws.\textsuperscript{178}

Different actors in the operation of the vehicle such as manufacturers, software developers and infrastructure providers make it difficult to determine who has the duty to always make sure that the product is safe. If software is combined with other services, responsibility for guaranteeing the safety of the combined product becomes unclear on whom it might fall.\textsuperscript{179} Does it remain the duty of the manufacturer or software developer to ensure that the product is safe on a regular basis when there was no network on the day when the product became defective if it relies on network? These are questions that complicate the duty of care.

\textsuperscript{174} ibid, 47.
\textsuperscript{175} ibid.
\textsuperscript{177} ibid,443.
\textsuperscript{178} ibid.
\textsuperscript{179} European Commission, ‘Advancing the Internet of Things’ (n21), 22.
4.3 Product Safety

In the European Union, the General Product Safety Directive (GPSD) sets out the standards that a product must comply with in order to be regarded as safe for consumers. The Directive applies to any product ‘in the context of providing services, which is intended for consumers or supplied for them’.\textsuperscript{180} Software is a product under the GPSD and therefore the legal duties of product safety under the Directive are applicable to the manufacturer or software developer of software. The manufacturer has the duty to place a product on the market that does not present any risks to the safety and health of the public.\textsuperscript{181} The Directive states that manufacturers who are deemed as producers have to make sure that the product is packaged with adequate instructions, warnings, installation and maintenance.\textsuperscript{182}

The Directive also imposes a legal duty of care for manufacturers not to supply products which they know or presume not to comply with safety standards and in the case of knowledge the product has to be recalled.\textsuperscript{183} In the context of manufacturers and software developers for software it means that they have a legal duty to ensure that before software is placed on the market it is packaged correctly with adequate instructions on warnings, use, installation and maintenance. These legal obligations are consumer friendly as they minimise the risks of unsafe software on the market.

4.4 Concluding remarks

Manufacturers and software developers have a legal duty to exercise reasonable care when they place products such as software on the market. These duties of care minimise the risk of defective software which has the potential to cause harm. The duty of care is greater if the self-driving car operates on level 3-5 as the car makes more autonomous decisions. It is therefore necessary that consumers be equipped with better knowledge on the use and risks associated with self-driving cars. The duty of care in this regard would be providing adequate instructions, warnings and proper information on the use of the self-driving car and how the software operates. Software developers have a duty to design and develop secure software that is safe.

\textsuperscript{180} Article 2 (a) General Product Safety Directive.
\textsuperscript{181} Article 2 (b) General Product Safety Directive.
\textsuperscript{182} Article 2 (b) General Product Safety Directive.
\textsuperscript{183} Article 5 (2) General Product Safety Directive.
Insecure secure is vulnerable and can be hacked by third parties. These parties can change different functions of the car and this affects security and safety. It is therefore necessary that software developers regularly check that the software is still up to standard to avoid these risks.

The General Product Safety Directive provides for legal obligations that producers of products must comply with when they place products on the market to promote public safety and health. These are applicable to manufacturers and software developers and the Directive emphasizes that they have to provide adequate warnings and instructions on the installation and maintenance of the product.

Some of the actions of autonomous systems are unforeseeable which makes the duty of care complicated as the manufacturer or software developer could not have anticipated it in order to put in place measures to minimise the risk.
CHAPTER V CONCLUSION

The purpose of this thesis has been to examine to what extent the manufacturer or software developer can be held liable for damages in a self-driving car that are caused by defective software. The main research question was the following

To what extent can the manufacturer or software developer be held liable for damages that are caused by defective software in a self-driving car under the Product Liability Directive?

The underlying aim of this thesis was to find out if the problem of allocating responsibility on the manufacturer or software developer for defective software that causes a self-driving car to cause damage is addressed under the PLD. In order for the software developer and manufacturer to be strictly held liable under the Directive, software should be considered as a product under Article 2.

The Product Liability Directive in its current state is only applicable to movable products only and not intangible products. Software in a tangible form is regarded as a product under the Directive but in its intangible form it falls outside the scope of Article 2. The Directive does mention anything about software as being a product or not. As a result, the legal status of software under European Product Liability Laws is unclear because the PLD does not include software in Article 2.

It is also unclear if the PLD is applicable when software is offered as a service. In Dutruex it was held that the PLD does not apply to services when the court had to answer if the hospital was strictly held liable for injuries suffered by a patient as a result of a defective mattress. Some scholars consider software as a service not a product. Regarding software as a service and not a product is problematic as consumers will have difficulties in finding the responsible party for the defective software that was offered as a service.

There are diverging views on the status of software by scholars but most of them are in favour of regarding software as a product under the PLD. Software being included as a product under the PLD makes it easier for the consumer to strictly hold liable the manufacturer or software
developer for defective software that causes damage. The applicability of the PLD in assigning responsibility to manufacturers and software developers for defective software is dependent on software being included as a product. The manufacturer or software developer cannot currently be strictly held liable for intangible defective software as it has not been clarified yet by the Directive or Court of Justice if it is a product.

The thesis also examined the PLD’s concept of defect in the context of software. Article 6 only mentions that a product is defective if it does not provide safety that the person is not entitled to. The Court of Justice in *Boston* clarified that it is the safety of the legitimate expectations that should be looked at and not subjective expectations. Defective software can contain errors which might make third access gain control of the software system and this has the potential to affect safety of the public. The court in *Boston* ruled that if one product is regarded as defect, the other products forming part of the same group have the potential to become defective. Software might not be defective during production but may be due to machine learning algorithms since they self-teach or due to a programming error. The defect would not have originated during mass production as in the *Boston* case so it is unclear if the courts will still apply this ruling to defective software or will decide on a case by case basis.

In the USA, the *Products Liability Restatement* does not include software under the definition of a product. Tangible products are included but just like in the EU software that is intangible is excluded. Both the product liability legislations of the EU and USA do not include software in their definitions of a product. US state courts have ruled that other intangible products such as defective computer data and navigational charts fell under the scope of product liability laws. It would benefit the consumer if the courts also hold manufacturers and software developers strictly liable for defective software under the *Products Liability Restatement*.

This thesis also looked at the concept of duty of care and assessed the duty that is expected of the manufacturer or software developer. Manufacturers and software developers must exercise reasonable care when they place products such as software by providing clear instructions on the use and maintenance. The duty of care also includes designing and developing safe software that is not unsafe. The duty of care is affected by the fact that some actions of autonomous systems are unforeseeable that it would be difficult to know the level of care that was expected to minimise the risk. The most important duty that the manufacturer has is to ensure public safety and the General Product Safety Directive states the legal obligations that producers must
comply with to prevent unsafe products on the market. These duties are also applicable in minimising the risk of defective software on the market.

In conclusion, the manufacturer or software developer can be held liable for damages caused by defective software in a self-driving car under the PLD if software is included as a product first. As it stands software is not included which makes manufacturers and software developers not strictly liable for defective software both in the EU and USA since the PLD and *Products Liability Restatement* do not include software as a product under the product liability laws. Therefore, software should be regarded as a product under product liability laws so that consumers can strictly hold the manufacturers and software developers for defective products.
6. Bibliography

6.1.1 Primary Sources

6.1.2 Legislation- Europe


6.1.2.3 USA Legislation

RESTATMENT (THIRD ) OF TORTS: PRODUCT LIABILITY (American Law Institute 1998)

6.1.3 Case law -Europe

1. Case C-495/10 Centre hospitalier universitaire de Besancon v Thomas Dutruex and Causse Primaire d’assurance Mladie du Jura [2001] ECR I -03569
2. Donoghue v Stevenson 1932 SC (HL), 580
3. Joined Cases C-503/13 and C-504/13, Boston Scientific Medizintechnik GmbH v AOK Sachsen-Anhalt and others

6.1.4 US Case Law

5. Nowak v. Faberge USA 32 F.3d 755 (3d Cir.1994)
7. Saloomey v. Jeppesen& Co. 707 F.2d 671 (2d Cir.1983)
6.2 Secondary Sources

6.2.1 Books


7. Terry Hutchinson and Nigel Duncan, ‘Defining and Describing What We Do : Doctrinal Legal Research Methods in Law ( Taylor &Francis Group 2013)

8. Van Dam Cees, ‘European Tort Law’ (Oxford University Press 2009),

6.2.2 Articles

1. Alan Butler, ‘Products Liability and the Internet of Insecure Things: Should Manufacturers Be Liable for Damages Caused by Hacked Devices (2017)University of


### 6.2.3 Documents issued by official bodies


6. European Commission, Communication from the Commission to the European Parliament ,the Council, the European Economic and Social Committee and the


6.2.4 Press


   accessed 25 June 2019


8. Luc Olinga, ‘Tesla bulks up on IT talent for ‘care of the future ‘ fight’<


10. Luke Dormehl and Stephen Edelstein, ‘Sit back, relax and enjoy a ride through the


13. Michael Hicks and Michelle Fitzsimmons, ‘Self-driving cars :your complete guide<

